COST SENSITIVE META LEARNING

SAMAR ALI SHILBAYEH

SCHOOL OF COMPUTING, SCIENCE AND ENGINEERING
UNIVERSITY OF SALFORD
MANCHESTER, UK

Submitted in Partial Fulfilment of the Requirements of the Degree of Doctor Of Philosophy, 2015

Contents

List of figures

List of Tables

ACKNOWLEDGMENTS

I would like to thank Allah Almighty, for giving me the strength to carry on this work and for blessing me with many generous people who have been my greatest support in both my personal and professional life. I would like to take this opportunity to express my deepest regards and gratitude to my supervisor Prof. Sunil Vadera for his dedication and support throughout this project.

Special thanks to my dad Ali and my mum Fatimah, and my sisters, Linda, Samah, Nour, Layal and Farah for all their support and assistance, without their love, help, and encouragement this journey would not have been possible.

Parts of the current PhD research have resulted in the following conferences and presentations.

Shilbayeh, S.,& Vadera,S.(2014). Feature selection in meta-learning framework. *Science and Information Conference (SAI)*, 269-275.

Shilbayeh, S., & Vadera, S.(2013). Meta-Learning framework based on landmarking *IMSIO2013 international Conference*. Proceedings of the 5th European Conference in Intelligent Management Systems in Operations, 97-105.

Shilbayeh,S.&Vadera,S.(2012). Cost sensitive meta-learning framework. *SPARC Conference*. University of Salford, UK.

Shilbayeh, S., Vadera .S.(2013). Feature selection in Meta-learning work. *Dean's Annual Research Showcase*. University of Salford, UK.

ABSTRACT

Classification is one of the primary tasks of data mining and aims to assign a class label to unseen examples by using a model learned from a training dataset. Most of the accepted classifiers are designed to minimize the error rate but in practice data mining involves costs such as the cost of getting the data, and cost of making an error. Hence the following question arises:

Among all the available classification algorithms, and in considering a specific type of data and cost, which is the best algorithm for my problem?

It is well known to the machine learning community that there is no single algorithm that performs best for all domains. This observation motivates the need to develop an "algorithm selector" which is the work of automating the process of choosing between different algorithms given a specific domain of application.

Thus, this research develops a new meta-learning system for recommending cost-sensitive classification methods. The system is based on the idea of applying machine learning to discover knowledge about the performance of different data mining algorithms. It includes components that repeatedly apply different classification methods on data sets and measuring their performance. The characteristics of the data sets, combined with the algorithm and the performance provide the training examples. A decision tree algorithm is applied on the training examples to induce the knowledge which can then be applied to recommend algorithms for new data sets, and then active learning is used to automate the ability to choose the most informative data set that should enter the learning process.

This thesis makes contributions to both the fields of meta-learning, and cost sensitive learning in that it develops a new meta-learning approach for recommending cost-sensitive methods.

Although, meta-learning is not new, the task of accelerating the learning process remains an open problem, and the thesis develops a novel active learning strategy based on clustering that gives the learner the ability to choose which data to learn from and accordingly, speed up the meta-learning process.

Both the meta-learning system and use of active learning are implemented in the WEKA system and evaluated by applying them on different datasets and comparing the results with existing studies available in the literature. The results show that the meta-learning system developed produces better results than METAL, a well-known meta-learning system and that the use of clustering and active learning has a positive effect on accelerating the meta-learning process, where all tested datasets show a decrement of error rate prediction by 75 %.

CHAPTER ONE: INTRODUCTION

1.1 Research Aim, Objectives and Motivation

There is no doubt in the machine-learning community that increasing the amount of data used in the learning process will positively impact the learning process performance because it increases the classifier version space size (Baxter, 2000; Dietterich, 1995); however, the question arises in regard to the type of data that should be used in the process. Moreover, there is consideration as to whether the next unlabelled data should be chosen randomly or whether a specific methodology should be adapted to select the most proper data. In the developed active learning selective methodology, the previous questions will be answered.

1.2 Research Motivation

This research is motivated by the fact that it is now widely acknowledged that there is no single machine learning algorithm that is always the best. For years, researchers have tried to develop better and better machine learning algorithms. Better decision tree learners, better neural networks, better association rule mining methods, etc. etc. In recent years, there has also been a recognition that costs as well as accuracy need to be taken into account, leading to further algorithms and issues.

Hence, there is a real need to develop a system for recommending an algorithm. However, we don't have the knowledge of which algorithm works best under a given situation.

Thus, the aim of this research is to build a meta-learning system for cost-sensitive learning that is able to understand the relationship between the learning task or domain and the learning strategies, and also develop a smart active learning method that is able to seek the data that the learner needs. The learning process is based on the accumulative experience gained from previous experience. In a meta-level learning process, learning does not need to start from scratch; instead each new task will take advantage of accumulating experience on the performance of multiple applications of a learning system.

1.3 Research Objectives

Given the above motivation, the research objectives are:

1. To carry out an in-depth, comprehensive literature review centred on the present data mining approaches, their use in meta-learning, cost-sensitive learning, and active-learning.

2. To devise a meta-learning system with the capacity to make suggestions concerning learning approaches that take cost into consideration

3. To devise an active learning approach that provides learners with the ability to choose the most informative data for the learning process, and accordingly quicken the learning process

4. To conduct an empirical evaluation of the meta-learning approach, and accordingly contrast the findings with another well-known meta-learning system.

5. To assess the active learning approach devised in this research by drawing a contrast between the findings obtained and those from a passive learning approach that randomly selects data.

1.4 Research Methodology

A number of research approaches have been examined, with the most suitable one adopted in this study. Different research methodologies include the following:

- **Descriptive research vs. Analytical research**

  This approach includes the study that explains the present state without any degree of control over input variables, commonly adopted in business and social science, where scholars can only explain the theory or the facts, the factors affecting theory, or otherwise explaining what occurs in regard to a particular phenomenon or what has happened. Thus, the descriptive research cannot consider the study results' validity owing to the fact it does not describe the result causes (Kane, 1983), whereas in the case

of the analytical research, the question is posed as to why it is that way or why we have this result. This is achieved through critical evaluation for the state through incorporating different input variables in an effort to complete a critical assessment for the results (Kothari, 2011; Silverman, 2013).

- **Conceptual vs. Empirical and Scientific methods**

Conceptual studies are carried out to devise a new theory or concept, or otherwise as an effort to describing a presence one, such as the cause behind a particular disease, for example. This is referred to as a 'pen and paper' approach owing to its reliance on the use of concepts, which are then either proven or disproven. Empirical studies, on the other hand, involve a number of experiments and observations carried out in an effort to validate (or otherwise) an existing theory, or to develop a new one. In such a research, the researcher has complete control over the input variables, as well as over the design of the experiment, adhering to researcher needs (Kothari, 2011). In contrast, scientific approaches make use of **both conceptual and empirical assessment approaches methods**, beginning with the formulation of a hypothesis, with experiments then designed with the aim of testing the suggested hypothesis. Complete control is maintained by the researcher in proving or disproving the hypothesis. In such approaches, it is common for researchers to prove theory through the completion of experiments and observations, ensuring research bias on the experiments result outcomes is decreased (Lakatos, 1980). The current study falls under this approach.

- **Applied vs. Fundamental**

Applied researches are those that have made use of different approaches in an effort to overcome problems facing businesses or other entities, or society. In contrast, fundamental studies are centred on establishing hypotheses or defining theory, including a new mathematical framework or devising a new scientific algorithm (Bond & Fox, 2013; Kothari, 2011).

- **Quantitative vs. Qualitative**

Quantitative studies are centred on quantity, measurements, its application in the field that uses different measurements, including computational, mathematical and statistical approaches, and methods to justify theory or facilitate understanding of the links between different patterns and entities. Importantly, quantitative concerns quality and types that cannot be measured, and has a strong link with sociological and behavioural considerations, mainly adopted in regard to marketing and social science, where the researches are concerned with establishing why people think in certain ways (Kothari, 2004, 2011; Patton, 1980; Silverman, 2013). The argument has been posited that both qualitative and quantitative methods need to be carried out alongside one another, with Kuhn (1996) stating that 'large amounts of qualitative work have usually been prerequisite to fruitful quantification in the physical sciences'.

The study methodology applied in this study is **scientific**, utilising both empirical assessment and conceptual approaches, and including the following stages

1. Outlining the study questions by emphasising the key issue driving the study, and including the fact that there is wide acceptance that there is no individual cost sensitive data mining algorithm that is most suitable, which means the data miner is required to assess all methods in an effort to determine the best in line with the issue.
2. Carrying out an in-depth literature review on the present methods and techniques to overcome the problem.
3. Design and implement a solution in mind of the problems, which involves devising a new cost-sensitive meta-learning system that has the ability to estimate the costs and accuracy for a particular cost-sensitive method, and thus guide the most suitable algorithm for a particular problem.
4. Empirical evaluation: This involves carrying out a number of experiments to cover all learning processes. Subsequently, the different feature selection methods are assessed considering the fact that there is no best feature selection method for all cases. Following,

the features that are seen to be uncorrelated or irrelevant are removed, with different cost-sensitive methods. The entire process is observed and assessed, with active learning proposed and developed in mind of eradicating some of the issues inherent in the meta-learning process. A mixture of 10 folds cross validation and leave one out validation methods are used in validation process.

5. Results analysis: Involves analysing and contrasting the results with similar works in the same domain. A conclusion is established from the findings. The key objective of the developed system is achieved, with the results seen to outperform the existing work in the same field.

1.5 Outline of Thesis

The following summarises the organisation of this thesis.

Chapter 1: Introduction

This chapter has presented an introduction to the thesis and the research hypothesis, motivation, and objectives.

Chapter 2: Background

This chapter presents the background and a literature review covering the fields of cost-sensitive learning, meta-learning, feature selection and active learning.

Chapter 3: Cost sensitive meta-learning system development

This chapter presents the development of a cost sensitive meta-learning system that will be able to recommend a cost-sensitive data mining method given a specific data set, and able to improve its recommendations as it learns from the experience gained from data characterization.

Chapter 4: Empirical evaluation of new cost sensitive meta-learning system.

This chapter presents an empirical evaluation of the developed system. This includes comparing the results obtained from applying a cost sensitive meta-learning system with the results published from METAL project. The evaluation is based on comparing the accuracy and the cost. It also includes an evaluation of the developed active learning system relative to passive learning with random selection

Chapter 5: Conclusion and future work

This concludes the thesis by reviewing the extent to which the objectives of this study have been met, summarising the contributions and presenting directions on the future work.

CHAPTER TWO: BACKGROUND AND LITERATURE REVIEW

This chapter provides the background, focusing on the four main areas contributing to this research: Section 2.1 covers cost-sensitive learning ; Section 2.2 covers meta-learning and its application in data mining; Section 2.3 describes feature selection methods and their application in data mining in general and in meta-learning work specifically; and Section 2.4 describes active learning, and covers the general trends of active learning in data mining, as well as the application of active learning in meta-learning.

2.1 Cost-Sensitive Learning

This section provides an overview of the cost-sensitive learning background and a literature review.

2.1.1 Cost-Sensitive Background

In classification problems, most learning algorithms aim at reducing the number of misclassification instances and increasing the number of correctly classified instances. In learning processes, costs are incurred, such as costs of misclassification, costs of testing, costs of obtaining data … etc. Cost-sensitive learning is a type of learning that takes costs into consideration; aiming at minimising the costs associated with the learning process. In cost-sensitive learning, an unknown example should be predicted to have the class that leads to the lowest expected cost (Elkan, 2001; Turney, 1995). Moreover, the model built from cost-sensitive learning should be developed in such a way as to achieve high accuracy but with low costs. An obvious example often presented in the literature for cost-sensitive analysis is medical diagnosis, where the doctor decides whether or not to carry out medical tests before making a diagnosis (Qin, Zhang, Wang, & Zhang, 2011; Turney, 1995; Zadrozny, Langford, & Abe, 2003). Using a test can incur costs, so therefore the question is raised as to whether or not it is worth carrying out a particular test. In credit card fraud detection, for example, if the cost of predicting credit card fraud is more than the amount of losses, is it then worth completing the test?

With the aim of developing a cost-sensitive learning concept, many researchers have worked in this field, with Elkan (2001) defining the term '***optimal solution***' as meaning the learning process that minimises the costs of misclassification whilst maximising accuracy. Turney (2000) lists the various different types of costs:

- Misclassification costs: is the cost of making error such as classifying a non-cancer patient as cancer patient.
- Test costs : is the cost of performing a specific test, such as the cost of performing a patient blood test
- Human computer interaction costs: this is the cost that needs human work, such as the cost of tuning the model parameters, the cost of applying domain knowledge into learned model, and the cost of transforming the data into specific format to be used in a specific machine learning system.

In the literature, misclassification costs are highlighted as being the most important costs in machine-learning (McCarthy, Zabar, & Weiss, 2005b; Turney, 1995, 2000), it depends on whether the predicted instance is a false negative (positive but classified as negative) or a false positive (negative but classified as positive). It is widely agreed that the costs of misclassification for the rare class (positive class) are often higher than the costs of misclassification for the negative class (dominant class).

The next section highlights one of the main problems in developing a cost-sensitive classifier, which is known as the imbalance data problem (Sun, Wong, & Kamel, 2009).

2.1.2 Cost-Sensitive Learning in the Imbalanced Data Problem

In machine-learning, the class imbalance problem is one of the key challenges associated with cost-sensitive learning. In some applications, one class is rare, and the costs of not recognising it correctly are very high. If the data is imbalanced there is a danger that a learner will be overwhelmed by non-rare classes and accordingly will tend to classify each instance as frequent instance (Kotsiantis, Kanellopoulos, & Pintelas, 2006; Sun et al., 2009), making the model useless. An obvious example of this problem is cancer diagnosis, where the cost of misclassifying an example is high but the proportion of cases is low. In such cases, any non-cost-sensitive classifier will tend to classify most instances as non-cancer, but the costs of

misclassifying cancer patients are more expensive (more serious) than misclassifying a non-cancer patient; therefore, for datasets that contain a 95% dominant class and 5% rare class, classifying all instances as the dominant class will produce a high accuracy of 95%. For the medical applications, such a classifier is rather useless as it fails to identify the disease, which is the main interest of the user. As a result of this, there is a serious need to adapt a certain approach to tackle this problem; thus, building a classifier that does not consider the cost of misclassification would not perform well owing to the fact it is biased in classifying most of the instances under the category of a frequent class, resulting in a useless classifier. Taking into consideration that the misclassification costs of rare items are usually higher than the misclassification costs of frequent items, the needs for a cost-sensitive learner with the capacity to deal with imbalanced data is fundamental in developing a good classifier. Importantly, devising a good classifier over skewed data needs to consider three different factors (Sun, Kamel, Wong, & Wang, 2007):

1. Imbalanced ratio: The ratio between rare classes and major classes.
2. Small sample size: Where a smaller sample size has more effect in terms of recognising infrequent behaviour.
3. Separability: Meaning the difficulties in separating the small ratio classes from large ratio classes.

More details on the imbalanced data issue are covered by Ganganwar (2012), the reader is also referred to a comprehensive literature review of cost sensitive learning by Lomax and Vadera (2013). The following subsections provide a brief description of cost-sensitive learning.

This section presents an overview of the different studies covered by the literature in terms of cost-sensitive classification. The word ‘cost’ in the literature is used to describe the term in a very abstract sense, where cost has different measurement units, such as monetary units (British pounds), temporal units (seconds) or abstract units of utility (Turney, 2000), for example. Cost is not only a physical entity that can be measured, but also includes time wasted and the loss in patient life, such as misclassifying a patient with cancer as having no cancer.

An extensive search of the literature has been conducted in an effort to identify all existing cost-sensitive learning algorithms. The available algorithms are established and classified by the method through which an algorithm deals with cost, and according to the cost type covered. Some of the algorithms aim at minimising a certain type of error, such as misclassification error, or the cost of obtaining data. On the other hand, some aim at minimising more than one error type, such as the costs of misclassification error and the costs of obtaining data at the same time.

Cost-sensitive algorithms vary according to the way they incorporate costs in the learning process. Two approaches are identified in the literature, the first of which is designing a classifier that is cost-sensitive, known as a direct method, whilst the second, which uses an indirect approach, involves designing a wrapper as a separate phase with the objective to convert cost insensitive learning algorithms to cost-sensitive. The following two subsections describe these two approaches

2.1.3.1 Direct Methods

The algorithmic approach changes the steps of an accuracy based classifier to take account of costs by directly utilising the misclassification costs (or other cost types) in the learning algorithm itself. For example, in regard to decision tree learning, the information theoretic measure is adjusted, along with the threshold, based on the costs of the various classes in an effort to include the cost of misclassification (Lomax & Vadera, 2013), with several works revealed under this category, such as *EG2* introduced by Nunez (1991), *CS-ID3* by Tan & Schlimmer (1989), and *IDX* by Norton (1989). All of these systems developed a cost-sensitive tree by introducing a new cost information ratio, which produces a cost factor in the information gain used in deciding which attribute the decision tree will select during the decision tree construction process.

In traditional cost in-sensitive decision tree induction, the entropy is calculated for the training data to measure how homogenous the data is (Quinlan, 1986). The ID3 decision tree algorithm developed by Quinlan (1986) uses the information gain measured for each attribute to decide which attribute should be chosen next when constructing a decision tree. This measure of information gain is based on the entropy for an attribute S, calculated using ***(2-1):***

$$\text{Entropy (S)} = \sum_{i=1}^{n} -p_i \, log_2(p_i) \qquad \textbf{(2-1)}$$

Where S is an attribute, n is number of example for which the proportion of examples in class i is p_i.

Given this definition of entropy, Information gain for a specific attribute X with respect to a set of example S is calculated using ***(2-2):***

$$\text{ID3:} \quad \text{Gain(X)} = \text{Entropy (S)} - \sum_{v \epsilon Value(X)} \frac{|S_v|}{|S|} \text{Entropy } (S_v) \qquad \textbf{(2-2)}$$

Value(X) is the set of X attributes values,

$|S_v|$: $total\ number\ of\ examples\ belongs\ to\ a\ specific\ attribute\ values\ , and$

S_V: *example belong to specific attribute value*

$|s|\ ia\ the\ total\ number\ of\ examples$

As mentioned above, ID3 uses the information gain value to decide which attribute to select. In direct methods that aim to take account of cost, this measure is changed to include costs. Different authors have experimented with different measures, leading to different algorithms. The measures, called the Information Cost Function (ICF) and the algorithms are:

EG2: $$ICF_X = 2^{Gain(x)} = \frac{1}{(C_X+1)^{\omega}} \qquad \textbf{(2-3)}$$

IDX: $$ICF_X = \frac{Gain(x)}{C_X} \qquad \textbf{(2-4)}$$

CS-ID3: $$ICF_X = \frac{(Gain(x))^2}{C_X} \qquad \textbf{(2-5)}$$

Where C_X is the cost of attribute x, ω is a predefined parameter that bias one attribute over others. For further details on cost-sensitive classifiers using a direct method, see the Literature review carried out by Lomex & Vadera (2013) and (Wang, 2013). The next subsection covers wrapper cost sensitive method.

2.1.3.2 Wrapper Methods

In this approach, the cost-sensitive learning process uses a wrapper in order to convert a cost-insensitive algorithm to a cost-sensitive without changing the internal behaviour of the learning algorithm. This is known as a black box as it deals with the algorithm as a closed box, without changing any of the classifier behaviours or parameters. In contrast to the transparent box, which deals with the algorithm itself (direct method), this approach does not require any knowledge of a particular algorithm behaviour. There are three methods that utilise this approach, all of which are detailed below.

A. Sampling

This class of algorithm changes the frequency of the instances in the training set according to its cost. As mentioned previously, the sampling approach was originally proposed to solve the problem of imbalanced data that affects the induced learner accuracy. The idea of this technique is to convert a cost in-sensitive learner to cost-sensitive learner by increasing the number of costly class examples and reducing the number of non-costly class. As a result, increasing the frequency of costly class will increase its weight, which ultimately reflects its importance. Elkan (2001) suggests changing the classes' distribution in the training set till the costly class has a higher number of examples that reflects its cost. In the literature, two sampling approaches are proposed, namely ***random sampling***, which implies changing the data distribution randomly, and ***determinate sampling***, which implies changing the data distribution in a predefined determinate way (McCarthy, Zabar, & Weiss, 2005a) . Both approaches can be applied to any algorithm in the pre-processing stage, meaning the algorithm will receive data relatively adjusted according to reflect its cost. Random and determinate sampling can be carried out in the following ways:

1. Over-sampling: Including increasing the number of costly class examples (Chawla, Bowyer, Hall, & Kegelmeyer, 2002).

2. Under-sampling: Including reducing the number of less costly class examples (Chawla et al., 2002; Drummond & Holte, 2005; Kotsiantis et al., 2006; Weiss, 2004).

It has been noticed that over-sampling can increase the occurrence of over-fitting because producing the exact copies of existing data produces a model that cannot perform well in the testing phase. In addition to this, it can produce an additional computational task if the training data is large (Kotsiantis et al., 2006; Mease, Wyner, & Buja, 2007). On the other hand, under-sampling could cause losses in data and may reduce the learning accuracy as it may discard some potential majority data. Generally speaking, re-sampling methods are attractive because they do not include any changes in the algorithm itself; instead, it adjusts the data distribution to make it more biased toward the costly class, meaning it is a simple way that can be adapted without being concerned about the internal classifier behaviour.

B. Ensemble Learning Method

The ensemble learning method is a supervised learning approach combining multiple models so as to produce a 'better' classifier. This depends on learning from multiple model prediction, which is combined in a specific manner (either voting or averaging) so as to induce a new learning model. The predictions of each learning process can be combined in different ways, such as through voting, averaging and weighting. The following summarises some of the ensemble methods from the literature.

- Boosting

Boosting is the process of inducing a set of classifiers on the same data set in order to achieve empowerment (Schapire & Singer, 1999). In this case, the process will be carried out in a sequential manner and in different turns. At the end of each turn, the weights are adjusted so as to reflect the instance importance for the next learning turn. The boosting technique was initiated by Kearns & Valiant (1988), who asked: '*Can a set of weak learners create a single strong learner*?' In boosting, the final result is the accumulation of individual learners applied to the dataset either by averaging or voting.

Cost-sensitive boosting includes changing the class distribution so as to reflect the cost of the induced class by assigning higher costs for the class that is more important. In

some papers, boosting is recognised as one type of sampling as a matter of changing the data set distribution, which is referred to in (Guo, Yin, Dong, Yang, & Zhou, 2008) as 'advanced sampling'. The following summarises boosting in cost-sensitive methods.

- AdaBoost: AdaBoost is a machine-learning algorithm developed by Freund, Seung, Shamir, & Tishby (1993), which learns a highly accurate learner by accumulating different weak hypothesis. Each instance is given a specific weight, which reflects its importance in the learning process. Importantly, each hypothesis is trained on the same training example, with different distributions on different turns. On each turn, the algorithm increases the weight of the wrongly classified instance and decreases the weight of the correctly classified instance.
- AdaCost: Although AdaBoost takes into consideration the misclassified class by increasing its weight and decreasing the weight of correctly classified classes, nonetheless, it deals with all misclassified classes in the same way; increasing the weight of misclassified classes and decreasing the correctly classified weight in the same ratio. For cost sensitive learning using AdaCost technique, costly classes are assigned more weight because it includes higher misclassification costs. This strategy is adapted by AdaCost, as proposed by Fan, Stolfo, Zhang, & Chan (1999), and uses the same strategy as AdaBoost but increases the weight of costly instances that are wrongly classified by a misclassification adjustment factor.

- Weighting: This approach is inspired by the boosting idea by weighting each instance in such a way so as to reflect its importance in the learning task. However, the boosting strategy, on the other hand, assigns the initial equal weight for all instances in the first step, where this weight either increases if the instance is misclassified or otherwise decreases. Ting (1998) proposes an instance weighting approach that provides a different weight for each example, according to its misclassification cost from the first turn.

➢ Bagging

In bagging, booststrap samples (random sample with replacement) are generated from the training set, applying a specific learning algorithm to different data samples with the

objective to generate different models that are aggregated so as to produce single outcome. Examples of cost-sensitive algorithms that use bagging include:

- MetaCost (Relabeling) (Domingos, 1999) is the name afforded to the algorithm that utilises the bagging approach. The main idea is to change the label of each training example to be the label of optimal class according to the conditional risk equation (minimising the cost) and then learning a new classifier in order to predict this new label. This is known as sampling with labelling.
- Costing, as proposed by Zadrozny et al. (2003), is the algorithm that utilises the bagging approach by applying a base learner to samples of the data for the aim of generating different models. The use of sampling in costing is based on a folk theorem, which implies the transference of 'a cost in-sensitive learner to cost-sensitive learner could be done by changing the training set instances distribution by multiplying it by a factor that is proportional to the relative cost of each example' which means changing the data distribution in the generated samples to minimize the cost of the original data. In contrast with MetaCost, costing changes the distribution of the training sample each time in such a way so as to minimise the misclassification cost and then uses the base learner in the new sampled data.

This section has described the use of wrapper methods for cost-sensitive learning. The next section will cover meta-learning and its application in data mining.

2.2 Meta-Learning Background and Literature Review

In traditional data mining, learning algorithms are applied in order to build a pattern that predicts the value of unknown attributes given the values of other attributes. Different data-mining algorithms are applied, such as clustering, association rule and classification, which include nearest-neighbour methods, decision tree induction, lazy learning and rule-based learning. The choice of specific data-mining techniques depends on the kind of data, the task that will be achieved and the domain area. Data mining techniques have been developed in recent years to fit all market needs—whether commercial, educational or medical. Most organisations are now seeking to add value to their data, which impacts their decision; however, most of these organisations are overwhelmed by data and look forward to

transferring such huge amounts of data into knowledge. Data mining provides a method of discovering this knowledge; unfortunately, however, both fields are growing constantly, which makes dealing with large amounts of data—and those developed techniques—restricted to specialised experts. Another way of doing this is to apply the repetitive processing of trial and error so as to garner satisfactory results. Meta-learning concepts provide many techniques that help in tackling this problem, such as by automatically learning from any previous learning experience and applying this knowledge when facing a new problem. Moreover, such techniques can be learnt from every new task, where such knowledge then can be applied to any new problem, thus being more experienced and informed over time. The following presents a meta-learning perspective, goal, application and general idea overview, in addition to various recent studies in this field.

2.2.1 Meta-Learning Perspective and Overview

Meta-learning has attracted considerable interest in the machine-learning community during recent years. This section presents some of the meta-learning definitions and concepts that help in defining the research area, and will accordingly highlight our problem in a suitable and thorough way. In this part, we will cover meta-learning goals and benefits, as well as meta-learning application.

A number of meta-learning definitions are given in the literature. The following provides a summary of those different meta-learning aspects and definitions:

- ‘Meta-learning is defined as the process of learning how to learn, i.e., the learner learns its learning process from its knowledge about the task under the analysis’ (Giraud-Carrier, 2008).
- ‘Meta-learning is the understanding of the interaction between the mechanism of learning and the concrete contexts in which that mechanism is applicable’ (Giraud-Carrier, 2008).
- ‘Meta-learning is the process of creating optimal predictive model and reuse previous experience from analysis of other problems’ (Vilalta & Drissi, 2002).
- ‘It is the process of automatically or dynamically learning an appropriate learner bias’ (Anderson & Oates, 2007).

All these definitions have a similar meaning, emphasizing the idea of understanding the behaviour of the existing domain and make the link between the current problem and the learning task, in order to learn from the learning process itself. This research takes advantages of the benefits of meta-learning to develop a cost sensitive meta-learning system.

2.2.2 Meta-Learning Goals and Benefits

The benefits of meta-learning can be summarised as follows:

1. Learning from the previous experience

   In traditional data mining, an algorithm is used on some data and there is no accumulated experience as a result of using the algorithm. For example, the algorithm may not have worked well on a particular type of data and the lessons learned would only be learned by the individual. In contrast, the idea with meta-learning is to learn from the accumulative experiences (Brazdil, Soares, & Da Costa, 2003; Vanschoren, 2010) .

2. Algorithm selection

   As mentioned above, meta-learning is used in order to choose an appropriate algorithm for a specific task (Vilalta & Drissi, 2002). The goal of a recommender is not only concerned with choosing the best algorithm but also on ranking the algorithms according to their predictive accuracy (Brazdil, Christophe, Carlos, & Ricardo, 2008; Brazdil et al., 2003) or any other performance measurement, such as accuracy and cost in this work. Generally speaking, the algorithm recommender is the process of choosing the best algorithm (set of algorithms) that produces good results after applying a set of algorithms on a specific data set with specific characteristics.

3. Model generation

   Model generation is different to the algorithm selector in that it finds model parameters to make it suitable for a specific task. Vanschoren (2010) points out that it is not only the algorithms selection that can be adapted using meta-learning, but also different parameter

settings, which will naturally allow the performance of the same algorithm to vary on different datasets, thus leading to a new concept in the meta-learning (Gomes, Prudancio, Soares, Rossi, & Carvalho, 2012). The choice of parameter value that results in best performance is carried out using the following steps: (1) a set of meta-features that characterises the problem under the domain is developed; (2) different parameter settings are chosen, along with their performance on different classifiers at base-level learning; and (3) a meta learner is used to build a model that predicts the performance of different algorithm settings on different problems or otherwise to predict the best algorithm parameters values (amongst a set of candidates), producing the best performance based on each data problem meta-features (Hutter & Hamadi, 2005)

2.2.3 Meta-Learning Literature Review

The above describes the main aim of meta-learning. This section presents several key studies in the meta-learning field, which are summarised and discussed below.

The idea of meta-learning is not new; one of the earliest studies in this field was carried out by Rice (1976), who proposed an initial model for the algorithm selection problem by defining four essential factors known to impact the algorithm selector:

1. The collection of problem instances.
2. A set of algorithms to tackle such problem instances.
3. A number of performance criteria to evaluate the algorithm.
4. A number of features characterising the instance properties.

The idea of meta-learning then is developed and proposed in the machine-learning community as a result of their needs to select an algorithm considered a best choice for a specific problem task. Wolpert & Macready (1997) suggest that there is no best solution to a specific problem, '*...for any algorithm, any elevated performance over one class of problems is exactly paid for offset performance over another class*,' which is referred to as the *No Free Lunch Theory* (Wolpert & Macready, 1997).

Several researchers (Aha, 1992; Rendell & Cho, 1990; Rendell & Ragavan, 1993) implemented Rice's initial abstract model by completing studies that characterise the problem instance, such as through consideration to data size (percentage of positive example) in an

effort to characterise the problem (Rendell & Cho, 1990), and using a rule-based learning algorithm to develop rules that control the algorithm selector problem: for example, *if the data has the following characteristics, C1, C2, then use algorithm A1 and A2* (Aha, 1992). Later on, a major European project known as StatLog (1991–1994) adopted the Rice approach in the algorithm selection problem by relating the characteristics of a task under analysis with algorithm performance. In this project, problem instances were characterised using different training set measurements, known as meta-features, where such characterisations are evaluated using simple methods such as dataset size, number of attributes and number of classes; statistical measurements such as mean value, skewness and standard deviation and theoretical measurements such as less entropy and noise. Moreover, they used different classification algorithms as meta-learners to predict the best algorithm for specific problems. The StatLog approach assigns an ***applicable*** and ***inapplicable*** label to each classifier after applying it on a specific data problem in comparison with the "best classifier" which is the classifier that has a lower classification error in the same problem, and depending on a predefined ranges for the ***applicability,*** for example if naïve Bayes performance is 90 % on a specific dataset (the best classifier), and the ranges of *applicability* is defined as $\mp$5%, then if neural network performs 80 % in the same dataset, it will consider as ***inapplicable***. The problem with this methodology is it's sensitive to the applicability boundaries which may be dependent on the data set.

Another significant work in this field is carried out in the NOEMON project (Kalousis & Theoharis, 1999). In this project, the performance of each data set is compared with another existing one (two algorithms are evaluated at the same time). NOEMON uses instance base reasoning to map between data characterisation space and algorithm performance space.

In a more recent study, a European project called METAL (Brazdil et al., 2003) uses K-NN as meta-learner to find the closest training data to the data under analysis. This project also introduced the concept of *landmarking* which adopts the idea of using a simple learning algorithm to understand the place of a specific problem space. The idea of *landmarking* is extended to *relative landmarking* (the relative order between landmarking algorithms) and sub-sampling, which is the use of a landmarker on a sample of the data (Furnkranz & Petrak, 2001). Other studies that also use the K-NN method to establish the datasets that are near to one another are (Brazdil et al., 2003; Nakhaeizadeh & Schnabl, 1997).

In contrast with the use of meta-learning for recommending an algorithm, several authors suggest its use for recommending parameters such as Vanschoren (2010), Gomes et al. (2012), Miranda, Prudancio, Carvalho, & Soares (2012), and Sun (2014). This approach is concerned with changing the model parameters to make a specific algorithm suit a specific problem, accepting the idea that different parameter tuning gives different results for a specific classifier on a specific dataset, then a meta-mining approach is defined in (Behja, Marzak, & Trousse, 2012; Hilario, Nguyen, Do, Woznica, & Kalousis, 2009, 2011; Keet, awrynowicz, daeamato, & Hilario, 2013) which is in contrast with meta-learning, the meta-mining approach is applied in all knowledge discovery processes rather than in the learning task only, and it opens the algorithm that was considered as closed box in meta-learning, as a result of this work *e-Lico* (*E-Laboratory for Interdisciplinary Collaborative)* is started which includes the following:

1. Planning the data mining process using hierarchical task networks.
2. Meta-Mining process: learning process that learns the whole knowledge discovery process.
3. Model Generation which includes dealing with the algorithm itself to optimize the algorithm mining task.
4. Data mining ontology (DMO) is the knowledge base that contains data mining from learning gained from previous experience.

Recently, a wealth of Literature reviews have been carried out to cover the meta-learning approaches and trends during recent years, such as the Literature review carried out by Balte, Pise, & Kulkarni (2014), which covers landmarking technology as an important data characterisation method, with its different challenges and future directions. In Lemke, Budka, & Gabrys (2013), the authors present an overview of meta-learning technology trends and challenges providing a general view for meta-learning concepts, different solutions in designing meta-learning frameworks, and covering general meta-learning research challenges.

2.3 Feature Selection

A significant problem that most data mining agents face is where to focus attention through the learning process. In order to achieve good results, the learning process should establish which part is most relevant, and should remove the part that is irrelevant. In data mining, it is fairly accepted that, if the data under processing is very large, the learning process will be consumed with dealing with a large amount of data, some of them are irrelevant, redundant and misleading, where this process is referred to as feature subset selection (Almuallim & Dietterich, 1991; Kira & Rendell, 1992; Pajkossy, 2013).

One direction of research is to continue to seek the ultimate feature selection method that always works well. Another approach, adopted by this research, is accepting that one method does not fit all requirements, but instead aims to identify which method works best for a given data set. However, this is not easy, since details of which algorithm works best under different circumstances is not known. Thus, we have a meta-learning problem, namely:

Can we automatically learn which feature selection algorithm works best for different circumstances?

Part of this work aims at answering this question by developing a new meta-learning system that aims to learn from the experience of applying different feature selection methods on data sets with different characteristics. This section is divided into two parts: feature selection background, which points out the main feature selection problem and different feature selection methods. Part two covers the recent literature of applying feature selection in the data mining process and in meta-learning.

2.3.1 Feature Selection Background

The feature selection problem has attracted a wealth of researchers in machine-learning (Almuallim & Dietterich, 1991; Kira & Rendell, 1992) . Many feature selection methods are proposed with a number of searching strategies. The following subsections describe the feature selection problem, and the various feature selection methods.

2.3.1.1 Feature Selection Problem

In this section, we highlight the feature selection problem, investigate the feature characterisation elements, and accordingly describe, in detail, the algorithms and techniques used for feature selection.

A general definition for feature selection is a process of selecting a subset of features that maximise predictive power (Kira & Rendell, 1992; Koller & Sahami, 1996a, 1996b), or to find a subset of features that are not correlated because correlated features adversely affect the performance of inductive learning algorithms (Yu & Liu, 2003).

A feature is characterised by the following (Molina, Belanche, & Nebot, 2002) :

- Relevancy

  A feature is relevant when it has a critical impact on a classifier's predictive power, and its role cannot be assumed by other features. A relevant feature plays an important role in the learning process, whereas an irrelevant feature is a feature that can be replaced by others without any influence on predictive power. Moreover, irrelevant input may lead to over-fitting, such as in the domain of medical diagnosis, for example, including the patient ID, which might induce a model that predicts an illness from a patient's ID (Ladha & Deepa, 2011).

- Redundancy

  A feature is redundant if there is another feature in the data that has the same effect, where both features are usually highly correlated, thus removing one will not impact learning power.

2.3.1.2 Feature Selection Process

A traditional feature selection problem comprises three main steps: feature subset creation, subset evaluation, and stopping criteria.

1. Subset creation

Subset creation is the process of selecting an optimal subset that maximises the predictive power and shows an improvement in the inductive process in terms of classifier performance, learning speed, generalisation capacity or the simplicity of learner (Molina et al., 2002). In the feature selection process, a subset candidate is created, with each subset then checked by using some evaluation criteria. If this subset shows better performance over a previous candidate, then the new subset will replace the previous one. This process is repeated until good enough solution is generated, with this process depending on a search strategy adapted by the feature selection algorithm. An important point that needs to be covered in the subset creation stage is the starting point: selecting the point in the feature space to start with all features, no features or randomly chosen features. In terms of search strategy, there are three approaches, as discussed below.

- Forward

  Starts with no feature and adds them one by one. If the new one performs better than the old one, the new one will replace it. This process continues until no more features add any improvement to the learning process. The process terminates here, with the best features or best subset of features.

- Backward

  This begins with all features and removes those that are irrelevant, removing subset by subset. Each time, the subset result is smaller than the previous one, until the last and most relevant subset is created.

- Bi-directional

  Starts anywhere in the middle and searches through the search space backward or forward. Practically speaking, forward selection is less computationally expensive because the search engine begins with a lesser number of features, which makes the process much faster. However, backward selection considers feature interaction

more in its searching strategy as a result of starting with a large number of features (Kohavi & John, 1997).

2. Subset Evaluation

Subset evaluation is the process of checking whether each subset candidate is good enough. According to some evaluation criteria, the process of subset evaluation falls into two groups: wrapper and filter methods. In the wrapper method, the classifier itself is used to evaluate which subset of the features is more predictive, eliminating the one which is less predictive (Kohavi & John, 1997). In this technique, the classifier as the black box is considered part of the searching process, taking the feature subset and evaluating its accuracy using cross-validation evaluation; in turn, another subset enters the evaluation process until the most suitable subset in induced (Blum & Langley, 1997; John, Kohavi, & Pfleger, 1994). The idea of the wrapper approach is shown in Figure 2.1(Hall, 1999b). This process is claimed to be computationally expensive as the learning algorithm itself undertakes the responsibility of finding the best subset; on the other hand, however, the positive aspect is that it reduces the algorithm bias because the same classifier, which is used in the searching process, will be used later on in the learning.

In the filter method, data will be characterised by itself, independently from the learning algorithm. The feature selection problem, in this case, totally depends on the data characteristics, its correlation, its dependency, and many other data evaluation criteria. The main disadvantage of this approach is that it ignores the effect of the inducer in the feature selection process; however, it is more adaptable when different classifiers are learnt for the same dataset as a result of not using the learning algorithm in the feature searching process. Nonetheless, its advantages outweigh its disadvantages (John et al., 1994):

- Reduced computational costs as less data is involved in the learning process.
- Adaptable to any changes in the learning algorithm.
- Much faster scaling to a large data set than the wrapper algorithm, as it can be re-run many times.

Some of the drawbacks known for the filter method include the fact that its results do not show an explicit choice of preferable feature subset; instead, ranked feature selection is

given, and the user should specify how many features are suitable for a specific learning process (Hall, 1999b). The filter approach concept is shown in Figure 2-1

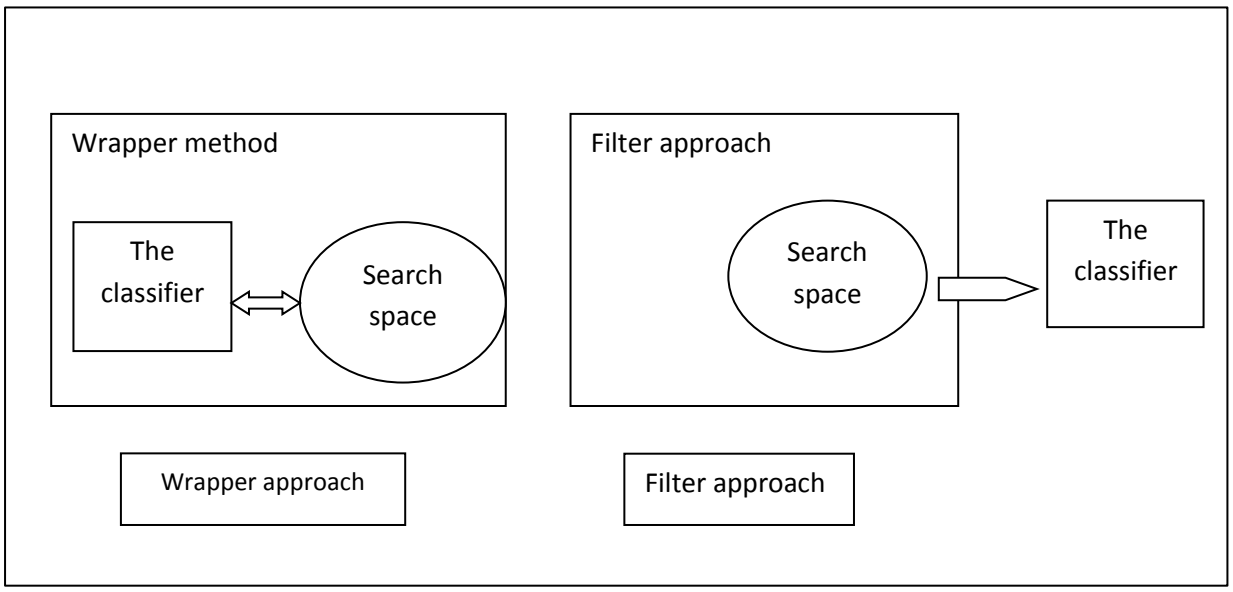

Figure 2-1: Wrapper and Filter methods (Hall, 1999b)

3. Stopping criteria

Feature selection has to decide when to stop searching through the feature space, which depends on the evaluation criteria. Some of the used stopping criteria are as follows:

- Adding or removing features that do not add any value to the learning process (they do not change the learning performance)
- Reaching a specific stop pre-defined point, which includes:
    - Whether a predefined subset is selected or a predefined number of iterations is reached (Dash & Liu, 1997)
    - The result is 'good enough'
    - Whether a specific number of features are selected
    - No changes in the evaluation performance if feature is added or removed.

2.3.2 Feature Selection Literature Review

The first pioneer works in feature selection were carried out by Almuallim & Dietterich (1991) and Kira & Rendell (1992), both of which used the filter approach in two different ways. The former defines the FOCUS algorithm, which starts with an empty set and uses the breadth-first search strategy to establish the good enough solution by choosing the minimal subset that can be used to determine the label value for all instances in the training set, whereas, Kira & Rendell (1992) defined the RELIF algorithm which assigns a weight to each feature and accordingly finds the good enough feature that exceeds a user threshold. These two researchers assumed Boolean attributes, whereas Kononenko (1994) reports extensions to their methods that handle non-Boolean attributes and multiple classes. Other recent works rely on the wrapper approach rather than the filter approach. As mentioned before, the main argument of using the wrapper method is that the inducer itself is used in the feature selection process, which will reduce induction bias, rather than using two different isolated strategies: one for feature selection and one for induction (as is the case in the filter approach). The first pioneer work that uses the wrapper approach in the feature selection problem was that by John et al. (1994), where a decision tree is used to establish the "good enough" feature subset. The simplest way to do this is to run the learning algorithm in a chosen feature subset and accordingly calculate the accuracy, then return back to introduce another subset that is either removed if it is worse or added if it is better. Various other attempts are carried out to replace the decision tree, used in the wrapper methods by other induction algorithms, such as (Aha & Bankert, 1994; Blum & Langley, 1997), both of which used a nearest neighbour algorithm within the wrapper algorithm. Likewise, Skalak (1994) works with the same concept but replaces greedy search with hill-climbing search.

On the other hand, Koller & Sahami (1996b) developed an information theory-based feature selection method that uses cross-entropy to minimise the amount of losses during the feature selection process. Yang & Honavar (1998) used a genetic algorithm for feature subset selection to search for a subset of features that give good enough accuracy. Moreover books written by Liu & Motoda (1998) and Liu (2005) provide a general overview of the feature selection methods and techniques, giving insight into important trends in feature selection work.

There is a wealth of research comparing different feature selection approaches in different applications. Hall & Homles (2003) performed a benchmark comparison of several attribute selection methods for two supervised classification learning schemes, C4.5 and naive Bayes. Both studies show that different feature selection methods perform differently in different cases which conclude no best methods for all cases and emphasizes on the need of an approach to know in advance which is best for a specific task.

In this work, the importance of feature selection exists in exploring the type of feature selection method most suited to a specific type of data, and which are far from its learning scope. Thus, this is the aim of the process, where a direct connection will be carried out between feature selection methods and dataset characteristics with the aim of establishing which algorithm performs best if a specific feature selection method is carried out over a specific type of data.

2.3.3 Feature Selection in Meta-learning Work

The literature search did not reveal any use of meta-learning for feature selection, other than its use for reducing the number of meta-features by Kalousis & Hilario (2001) and Todorovski, Brazdil, & Soares (2000). In the first work, a feature selection wrapper method is used to make a ***pairwise comparison***. In this comparison, the relative performance of each pair of base learners is contrasted using different meta-learners after removing irrelevant meta-features using feature selection methods. The result shows the importance of using feature selection in the meta-level process. In the second work, feature selection methods are used to remove irrelevant features in meta-level learning. In the first step, when a new dataset emerges, a similar dataset from the meta-knowledge is found using k nearest point by calculating the nearest distance between two datasets using their attributes values, then the predicted new label for the new data is calculated to be the average of performance of each k nearest points found in the meta-knowledge. The performance for both before and after using feature selection in the meta-level process shows that choosing the relevant meta-feature has a potential effect on meta-learner predictive power.

It is worth pointing out that, with those considerable available works in feature selection in general, a data miner has to choose which feature selection approach to use, which requires

time, effort and considerable experimentation. Hence, this thesis will also explore the use of meta-learning for feature selection.

The next section covers active learning and its application in data mining process.

2.4 Active Learning

Active learning is a subfield of machine-learning, where the learning algorithm is given the ability to choose the data from which it learns with the aim of making the learner perform better with less training examples. The learning algorithm in active learning has control over the input data in its learning process (Cohn, Atlas, & Ladner, 1994; Lindenbaum, Markovitch, & Rusakov, 2004; Tong & Koller, 2002). This is a very practical way of reducing the costs needed to label all training examples whilst eliminating some degree of redundancy in the information content of the labelled examples. Active learning is well-motivated in many modern machine-learning problems, where unlabelled data may be abundant or easy to obtain, such as in the case of unlabelled webpages available in thousands of pages; however, labels are difficult, time-consuming or expensive to obtain, such as classifying each web page as 'relevant' or 'not relevant' to web users. It has been shown that less than 0.0001 % of all web pages have topic labels (Fu, Zhu, & Li, 2013); therefore, having to label those thousands of web pages would be very time-consuming. Another example is detection of fraud in banks, where an expert in a bank needs to invest considerable effort and time to label a bank customer's behaviour as fraudulent or not fraudulent.

From previous motivation, there are general machine learning trends that use the unlabelled examples, as well as the labelled ones, in the learning process. In active learning, the learner has the capability of asking for the labelling of any unlabelled examples by directing the query process into more informative unlabelled examples. It is fairly well known in the machine learning community that using active learning that queries the data instances to be labelled for training can achieve higher accuracy with data in comparison with passive learning, which ultimately depends on blindly training with all available data—and economically, it is more effective (Roy & McCallum, 2001; Settles, 2011) . Many studies have developed different strategies that are able to achieve this aim, as will be explained later in this section.

2.4.1 Active Learning Background

This section sheds light on the main directions of using active learning in machine learning and provides a background on this topic and in using active learning for meta-learning.

Active learning is best defined in contrast to passive learning: in passive learning, a learner has access to the initial randomly generated labelled examples to build a hypothesis that covers most of the training data. In the traditional learning process (passive learning), the more labelled training data that is available, the more accurate the classifier that will be developed (Prudancio & Ludermir, 2007). In most learning problems, having a labelled (classified) example is not an easy process; this needs time and effort. Therefore, developing a strategy that conducts a classification process with the least possible labelled data is the main aim of active learning. The active learning model is a slightly different framework than other learning frameworks in which a small number of available initial data has labels, whilst a large amount does not come with any labels; that is, most of the training data set is simply an input without any associated label. The goal of active learning is to seek more unlabelled examples for labelling but those unlabelled example should have specific condition to say that they are 'informative enough' to be allowed to enter the labelling process. The term 'informative' is defined in active learning using different methods and different strategies. The following presents an overview of active learning scenarios and methods.

2.4.1.1 Active Learning Scenarios

There are many scenarios in which active learning can be used. The main three scenarios found in the literature are described here: (1) membership query synthesis, (2) stream-based selective sampling, and (3) pool-based sampling.

1. Membership Query Synthesis

   One of the earliest scenarios introduced in the active learning problem is membership query (Angluin, 1988). In this scenario, the learner will use any unlabelled examples from input space; sometimes, the learner generates their own data. This approach has encountered many problems as the data generated sometimes does not represent the original data distribution.

2. Stream-based Selective Sampling

In the stream selective sampling, instances are drawn from actual data distribution, with the learner then deciding whether to include or discard it from the learning process. This approach, in some studies, is known as stream-sampling or selective sampling (Settles, 2011) because instances are selected one by one to enter the learning process.

3. Pool-based Sampling

This scenario is the most widely used scenario in active learning problems. In this scenario, the learner considers that there is a small pool of labelled examples and a large pool of unlabelled examples. Each time a query is applied on an unlabelled example, the most informative (or best) unlabelled example is chosen to be labelled, this includes starting with a large pool of unlabelled examples, picking a few points for labelling, then querying for more instances to be labelled in many different ways. Below is the active learning process for the pool-based sampling scenario (Melo , Hannois , Rodrigues , & Natal 2011) :

- Start with a large pool of unlabelled data
- Pick a few points at random and get their label
- Repeat the following:
    1. Fit a classifier to the labels seen so far
    2. Pick the BEST unlabelled point to get a label for:
        i. Closest to the boundary.
        ii. Most likely to decrease classifier uncertainty.
        iii. Most likely to reduce the overall classifier error rate.
        iv. High disagreement between learners (in the case of more than one classifier) on how to label it.
        v. Most likely to reduce variance and bias.

The pool-based scenario is different to the stream-based scenario in the sense that the pool-based scenario evaluates the given data prior to selecting the best query. This evaluation depends on the strategy used for this process. On the other hand, however, data

in the stream-based sampling scenario comes as stream, with the learner then evaluating each query one by one, and accordingly deciding whether to accept or discard the data. In our work, we will use the pool-based sampling.

In this scenario, deciding which data is more informative is known as ***query strategy***. Below is a description of the different query strategies found in the literature.

2.4.1.2 Active Learning Query Strategy

In the literature, several methods have been used to query the most informative data for labelling. Different query strategies are described below.

1. Uncertainty Sampling

Uncertainty sampling is the most common search strategy used in active learning. In some studies, it is known as ***query by uncertainty*** or ***uncertainty reduction***. In this strategy, the pre-built classifier (the classifier that was built from the small pool of labelled data) is less confident in terms of how to classify the labelled data; those are the data that are nearer to the classifier boundaries. As a conclusion, if data uncertainty is high, this implies that the current model does not have a sufficient knowledge in how to classify it, thus meaning it is most likely that including such data in the learning process will improve model performance. This strategy is used in different machine-learning fields, such as natural language processing (Lewis & Gale, 1994), and in different classifiers: decision tree (Lewis & Gale, 1994), support vector machine (Tong & Koller, 2002) and in nearest-neighbour classifier (Lindenbaum et al., 2004). The key point in this approach is how to evaluate uncertainty measurements.

In uncertainty sampling, uncertain instances are the most attractive instances, with different uncertainty schemas used to measure those instances, such as a well-known entropy measurements used to measure data uncertainty (Holub, Perona, & Burl, 2008). Density measure is another example of uncertainty by calculating the number of examples that are nearer to the examples that are uncertain (more examples around specified data means that this data is more unlikely outlier) (Zhu, Wang, Yao, & Tsou, 2008a).

2. Error Reduction

This method depends on the estimation of how much classification error is likely to be reduced using the new dataset (Cohn, Ghahramani, & Jordan, 1996; Zhu, Lafferty, & Ghahramani, 2003).

3. Variance and Bias Reduction

This method depends on selecting samples that reduce learner bias and variance, which can be done by applying two learners: the single learner, such as naive Bayes, and bagging using naive Bayes as its base learner (or any other ensemble techniques), then asking the active learner to choose the informative data that reduces the difference between prediction using the first learner and the second learner (Aminian, 2005).

4. Outlier Detection

Most of work carried out on active learning considers the data around the decision boundaries, as explained above (uncertainty sampling), although it is worth pointing out here that the uncertainty sampling technique fails to detect outliers. Outlier data is highly uncertain but does not provide any help in the learning process because outlier data is defined as an observation that deviates too much from other observations, and also does not provide any extra knowledge if they enter the learning process.

In Figure 2-2, both A and B points are closer to the decision boundary, meaning they will be taken as uncertain samples; however, Point A is more informative than point B because there are two label examples close to it, whilst point B is considered an outlier point. The density measure can be evaluated based on how many examples there are similar or near to it, where greater density samples means more data near to a specific point, which means that it is less likely this point is an outlier (Xu, Akella, & Zhang, 2007; Zhu et al., 2008a).

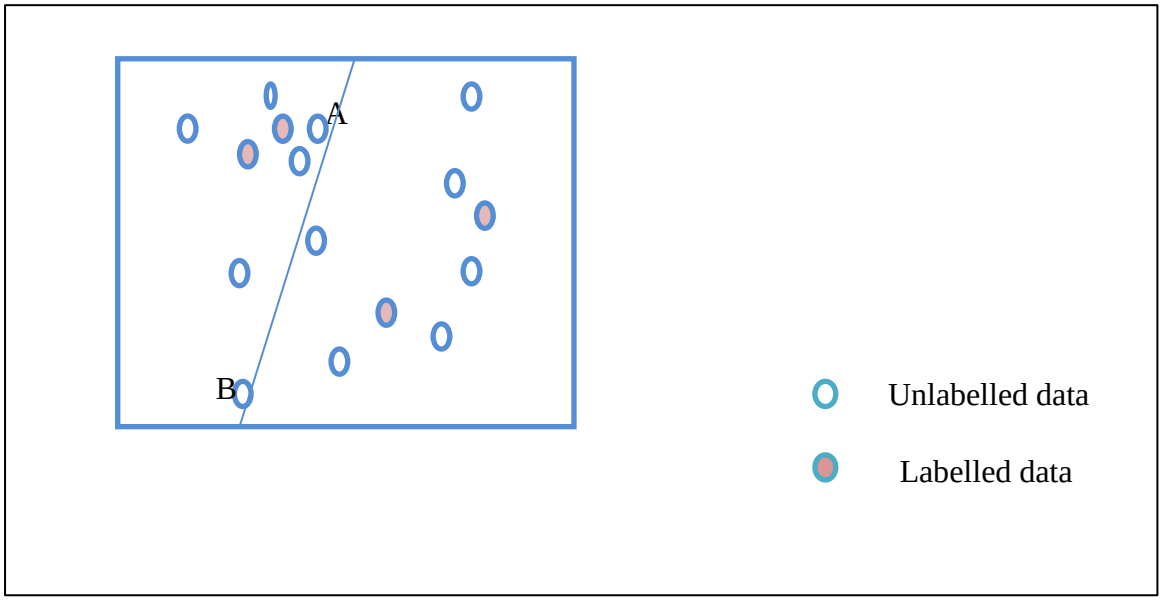

Figure 2-2: Uncertainty samples that are outliers (Zhu et al., 2008)

5. Query-By-Committee

Query by committee is a very effective active learning approach used in many classification problems (Cohn et al., 1994; Melville & Mooney, 2004; Seung, Opper, & Sompolinsky, 1992). It is the same as the ***query by uncertainty*** method, although the main difference between query uncertainty sampling and query by committee is that query by committee is a multi-classifiers approach where more than one classifier is imposed upon the original labelled data set. The more there are disagreements on how to

classify a piece of data, the more informative the data. The idea behind this strategy is that enables classification strategy, where different classifiers should agree, to a high extent, on classifying new instances (by voting or averaging). In active learning, the disagreement gives new instances the favour of being chosen to enter the new classification process (Melville & Mooney, 2004) because this disagreement means that the classifier (set of classifiers) is not sure (not certain) on how to classify it, which emphasises the uncertainty sampling concept.

6. Sampling by Clustering

In previous methods, the labelled data used to build the first learner is chosen randomly with the assumption that they have some prior data distributions; in most cases, however, this assumption is not correct due to the small size of initial data chosen, meaning it is most likely they are not a good representative for the whole dataset. For the aim of solving this problem, some researchers suggest clustering the whole dataset and then choosing the data that is more representative. Such data are the data that are nearer to the initial clusters' centres (Nguyen & Smeulders, 2004; Xu, Yu, Tresp, Xu, & Wang, 2003; Zhu et al., 2008a). In the work of (Zhu, Wang, Yao, & Tsou, 2008b), all unlabelled examples are clustered first, and data that are around the centre are taken to form the first set of labelled examples. Subsequently, more informative data are queried from unlabelled data using uncertainty sampling and density measures, as described above.

Another work adopted active learning by clustering (Nguyen & Smeulders, 2004), in which active learning using clustering is developed so that data from the same cluster is given the same label, which reduces the efforts needed to label all data in the same cluster. In the study of (Xu et al., 2003), the active learner measures uncertainty sample for the data around boundaries, then clusters those (uncertain) samples to query the most representative points around the cluster centre for text classification problems.

2.4.2 Active Learning in Meta-learning Scenarios

In different machine-learning problems, it is recognised that the existence of several algorithms compete to be applied to specific classes of problems. In a traditional way, a

costly empirical evaluation of each classifier on the given problem is the most widely adopted strategy; alternatively, reliance can be placed upon the expert knowledge to choose the most effective classifier. Both ways are ineffective because, in the first case, it is impractical to try every single possible algorithm on a data set—especially given the existence of large number of algorithms and large sizes of datasets. On the other hand, however, if the expert consistently selects an in-effective classifier, meaning the most effective classifier will not be learned through the learning process, so it will be very practical if the most effective data mining procedure is known in advance. Through using the concept of meta-learning, we can build an algorithm selector that is able to establish, in advance, the most effective classifier to be used in a specific learning problem. In meta-learning, meta-knowledge is used to predict which algorithm is more effective for a specific data set, or to predict the performance of a given algorithm in a given dataset. The desired meta-knowledge is developed by using a set of training examples, known as meta-examples. Each meta-example stores a number of features that characterise a given example, known as meta-features, and the performance of the set of algorithms applied to those examples when empirically evaluated over it.

The main aim of using active learning in meta-learning is providing the learner with the ability to choose the most suitable, informative data for its learning process; where the most suitable data is the data that creates new knowledge in the process of meta-knowledge development, because training data that doesn't add any new knowledge is time-wasting and resources-consuming. Meta-knowledge development is the process of applying learners on meta-examples to create the basic knowledge that is able to predict which classifier (or set of classifiers) is more appropriate for specific data characteristics. The process of developing meta-knowledge is the process of generating different meta-examples, as required by performing an empirical evaluation of candidate classifiers on different sets of problems. Essentially, the more diverse the problems used, the more general and richer the meta-knowledge that will be developed. This process is highly expensive as it depends on the methodology of empirical evaluation, the number of candidate algorithms and the number of meta-examples used in the meta-knowledge development process. There is no doubt in the machine learning community that increasing the amount of data used in the learning process will positively impact the learning process performance because it increases the classifier version space size; however, the question arises in regard to the type of data that should be used in the process. Moreover, there is consideration as to whether the next unlabelled data

should be chosen randomly? Or whether a specific methodology should be adapted to select the most proper data? In the selective methodology, the previous questions will be answered and the work will be evaluated by making comparisons between active learning using random sampling and active learning using a selective cluster-based algorithm ALBC (Active Learning Based on Clustering).

2.4.3 Active Learning Literature Review

The studies of active learning initially focused on a model of learning with membership queries (Angluin, 1988). In this model, the learner is given the ability to query any unlabelled data for labelling without any constraints. The main drawback for this model is that a query may result in data that are very far from its data original distribution. Hence, active learning studies turned to seek unlabelled examples that are more informative for the learning process. This in-formativeness varies in different studies and has different points of view with very different justifications. The following summarises the different active learning approaches found in the literature review.

In the literature, the most common technique used in a labelling query is uncertainty sampling (Cohn et al., 1994; Lindenbaum et al., 2004), which chooses a sample where the classifier is mostly uncertain on how to classify. This means choosing the unlabelled data that lies on the decision boundaries, and using them to clarify this boundary (Lewis & Gale, 1994; Lindenbaum et al., 2004; Tong & Koller, 2002). In a related approach, Freund et al. (1993) introduced the committee-based approach deploys the same idea of uncertainty sampling, but the prediction will be done by a set (committee) of learners as opposed to a single learner. This allows competing hypotheses, where each member votes on the labelling of query candidates. More disagreement between committees indicates that the chosen examples they have voted on are more informative.

In the same field, committee by bagging and committee by sampling is proposed by Mamitsuka (1998), which uses ensemble methods to help reduce errors (Lindenbaum et al., 2004; Roy & McCallum, 2001).

Several papers have surveyed active learning, focusing on different active learning perspectives and directions: the Literature review done by Settles (2011) focuses on general active learning scenarios and query methods, whilst that by Dasgupta (2011) concentrates on

two directions of active learning, namely search through a version space to find the data that reduces it, whilst another is exploiting the cluster structure in data, which is our approach in this research. On the other hand, in the work of (Fu et al., 2013), the authors concentrate on the active learning methods that take the instance correlation between each other when selecting the most informative data for labelling.

In meta-learning, the literature shows that active learning is used to reduce the amount of meta-examples needed to create a meta-knowledge (Prudancio & Ludermir, 2007, 2008; Prudancio, Soares, & Ludermir, 2011). In a study by Prudancio & Ludermir (2007), active learning is used to help a meta-learner to choose the most informative data for its learning process. Initially, K-NN is used as a meta-learner to label unlabelled examples by using the previously labelled examples. The active learner then chooses the most uncertain unlabelled example to enter the learning process. The K-NN uncertainty is calculated using the following ratio:

$$uncertainity\_ratio = \frac{D_u}{\sum_{i=0}^{U} D_{ui}} \qquad (2\text{-}6)$$

D_u: is the distance between unlabelled data and the nearest label neighbour

$\sum_{i=0}^{U} D_{ui}$: The summation of distances between unlabelled data and their nearest neighbour labelled data. The highest value of this ratio indicates that the data has similar points with a conflicting label, which means that the current label is uncertain in terms of how to classify this instance. Later, in a study by Prudancio & Ludermir (2008), three different active learning methods were combined to produce a ranked assessment for active learning problems. Those methods are:

1. Uncertainty method: using the uncertainty ratio in ***(2-6)***.
2. Entropy method: this method adapts the entropy measure, which defines the probability of class prediction using K-NN classifier, entropy can be calculated using the following equation :

$$Ent(\tilde{e}|\mathrm{E}) = -\sum_{l=1}^{L} p(C(\tilde{e}) = c_l|\mathrm{E}) * \times og_2\ p(C(\tilde{e}) = c_l|\mathrm{E}) \qquad (2\text{-}7)$$

E is the set of labelled examples, $\tilde{E}$ is a set of problems used to generate unlabelled example, E_l is the subset of labelled problems associated to the class label c_l

3. Combined method: This is the combination of the two previous measures (uncertainty and entropy), using the following equation:

$$Combined\ Ratio = uncertainity\_ratio \times W_u + Ent(\tilde{e}|\mathrm{E}) \times W_e \qquad \textbf{(2-8)}$$

W_u and W_r: weights for uncertainty ratio and entropy ratio such as $W_u + W_r = 1$.

uncertainity_ratio and *Ent* is the uncertainty ratio and entropy ratio computed in ***(2.6)*** and ***(2.7)*** respectively.

Another work (Prudancio, Soares et al., 2011) uses uncertainty sampling combined with datasetoids to select the most informative data, where a datasetoids is an easy way of generating meta-examples by switching between data label and independent attributes, Accordingly, the independent attribute in meta-example becomes a class label in datasetoids and a class label of meta-example becomes an independent attribute in datasetoids. Datasetoids is a way to obtain a new datasets from existing ones to increase the number of meta-examples (see Figure 2-3).

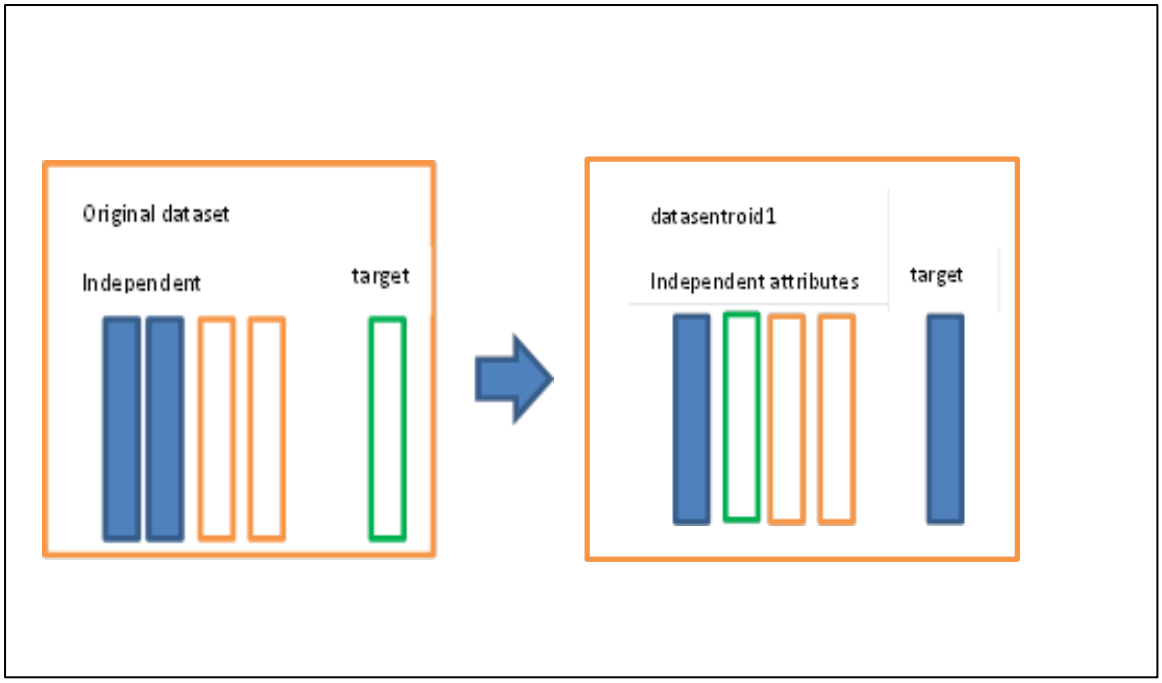

Figure 2-3: Datasetoid generation (Prudancio, Soares et al., 2011)

2.5 Summary of the Literature Review

This chapter has presented an in-depth Literature Review, which shows that a number of different efforts have been directed towards developing data mining algorithms that adhere to different business needs, highlighting recognition of the value of costs in the decision-making process. As such, several authors have developed cost-sensitive algorithms that are centred on various cost types in the data mining process, with additional feature selection approaches suggested in order to eradicate or decrease the impacts of irrelevant characteristics. A wide-ranging Literature review of the field has been conducted to appreciate the various methods and to answer the question driving this study.

The Literature review carried out revealed that there has been little work on applying meta-learning for recommending cost-sensitive learning algorithms. The Literature review summarised two categories of cost-sensitive learning methods as follows:

1. *Direct methods* that deal with the internal structure in an effort to ensure cost sensitivity' and

2. *Wrapper methods* that add individual phases that complete the conversion of any error-based classifier in such a way so as to make cost-sensitive decisions.

Using these two approaches, any cost insensitive classifier can be converted into cost sensitive. The major benefit of using a wrapper approach is that it works with the data itself without amending an algorithm, so it requires no knowledge about the algorithm which helps in applying this approach to any new algorithm, so it is straightforward and easy to implement. Some researchers have found that applying a resampling on imbalanced datasets considering the cost of each class performs very well. However, it includes some drawbacks when such as over-fitting over-sampling, information losses in under-sampling and it needs time to be applied especially in the case of large datasets. For example, MetaCost which includes bagging with re-labelling needs considerable time, especially in the case of large datasets, because a learner is applied on different data samples and then each data sample is relabelled with the label that gives minimum expected cost. This process could be very slow in the case of a large dataset. Other examples are AdaBoost and AdaCost, both of which includes applying different learners on the same dataset considering the weight of

misclassified instance on each turn by giving instances that are wrongly classified more weight than instances that are correctly classified. Both of AdaCost and AdaBoost reduce over-fitting because more than one learner is applied during the learning process which reduces learner bias, but applying them could be a slow process as well. On the other hand, making the classifier cost sensitive by directly providing and utilizing the cost into the learning algorithm without any changes in the data under processing is an effective way because it is faster than a wrapper method.

Generally speaking, it is noticed that some cost sensitive algorithms work well on a specific dataset and don't perform well on another dataset. Given the difficulty of knowing which cost-sensitive method works well for given situation, the thesis aims to adopt meta-learning. Hence this chapter also surveyed the field of meta-learning; Table 2-1 summarises different meta-learning approaches.

Table 2-1: Meta-learning approaches

Year/Reference	***Meta-learning approach***
1975 (Rice, 1975)	Meta-learning main elements are proposed which includes: problem, problem characterizes, algorithm, and evaluation performance.
1997 (Wolpert & Macready, 1997)	No free lunch theory which stated that no one machine learning algorithm outperforms others in all cases
1990-2000 (King, Feng, & Sutherland, 1995)	Algorithm selectors automation project with different dataset characterization such as METAL, statLog, and NOMEAN
2000-2009	Ranked algorithms using different meta-learner such as

(Brazdil et al., 2003)	K-NN which is used to rank algorithm according to specific criteria rather than giving single algorithm
2000-2005 (Pfahringer, Bensusan, & Giraud-Carrier, 2000)	Landmarking which includes using a simple algorithm to understand the place of a specific problem space, suitable if applied on simple, non-complicated landmarker algorithm.
2007-2011 (Prudancio, Soares et al., 2011)	Active learning in meta-learning concept, used in area where unlabeled examples are numerous, and labeled example are expensive or hard to obtain.
2010-2014 (Sun, 2014)	Model generation : used to characterize the algorithm itself by applying a meta-learning concept on algorithm parameters.
2012-2014 (Keet et al., 2013)	Meta-mining which includes meta-learning in all data mining processes.

The chapter also surveyed active learning methods which have the potential of improving the meta-learning process. Active learning methods are suggested as a way of overcoming supervised learning issues when the training data is costly or otherwise difficult to secure. Active learning is applied in an effort to provide the learning process with control over the data utilised throughout its learning process through the use of a small set of labelled examples and larger set of unlabelled examples. As can be seen when reviewing the literature, active learning is applied in the context of the meta-learning framework in order to decrease the number of meta-examples in an effort to speed-up the meta-learning process. As shown in prior work, active learning is summarised into the following categories, (see Table 2-2 and Table-2-3):

Table 2-2: Active learning approaches

Active learning approaches	***Description***	***Drawback***
Membership query (Angluin, 1988)	Active learner uses any unlabeled data includes artificial data	Data generated is very far from the original.
Stream based selective sampling (Cohn et al., 1994)	Data is chosen randomly then the learner decides whether or not to include it in the learning process	Data is drawn from original data distribution but the learner receives data one by one and then selects the data that are more informative, which is a slow process and similar datasets could be chosen more than one time.
Pool based sampling (Melo et al., 2011)	A large pool of unlabeled data and a small pool of labeled data	The learner scan all data and choses the data that are more informative, disadvantages of this method depends on the way of the informative data will be selected (the query strategy).

Table 2-3: Active learning query strategy approaches

Active learning query strategy	***Description***	***Drawback***
Uncertainty sampling (Holub et al., 2008)	Chooses the data that the classifier is uncertain about	Includes outlier data which are uncertain but outside the scope of the learning process.
Error reduction (Roy & McCallum, 2001)	Chooses the data that reduces the classifier error	Data that is chosen from small pool doesn't represent the whole dataset
Bias and variance reduction (Aminian, 2005).	Chooses the data that reduces the learner bias and variance	Data that is chosen from small pool doesn't represent the whole dataset.
Outlier detection (Zhu et al., 2008a)	Chooses the data that are uncertain after removing the outlier data.	Data that is chosen from small pool doesn't represent the whole dataset.
Query-By-Committee (Cohn et al., 1994)	Chooses the data that set of classifiers are mostly not agreed on how to classify it	Data that is chosen from small pool doesn't represent the whole dataset.
Active learning based in clustering (Urner, Wulff, & Ben-David, 2013)	Cluster the whole labeled data then choses the data that are around the cluster center	No clear studies about the quality of generated cluster.

In summary, the literature review shows that there is a real need for understanding when cost-sensitive algorithms work well and for an approach to recommend a particular learner given a new data set.

Hence, the following chapter develops a new meta-learning system for meta-learning that includes feature selection methods, cost sensitive methods and that also explores the use of active learning.

CHAPTER THREE: COST SENSITIVE META LEARNING

As motivated in chapter 1, there is no single best algorithm that establishes good enough cost-sensitive classifiers. Moreover, the main aim of this thesis was highlighted, which is exploring the development of methods that can continuously improve so as to make recommendations based on new datasets. This research explores two ideas, both of which are described in this chapter: Section 3.1 describes an approach that learns about learners as new data sets are presented, whilst Section 3.2 describes an active learning approach that aims at identifying datasets geared towards assisting the system in improving what it learnt quicker. Chapter 4 presents an empirical evaluation and comparison between the work presented in this thesis and existing work in the field.

3.1 Cost-Sensitive Meta-learning

Given there is no unique, widely acknowledged best cost-sensitive learning algorithm, the ideal is to provide a system that will take a dataset and recommend a learner that will produce an good enough cost-sensitive classifier. Although the literature contains various comparisons and some ideas about when certain systems work well, there is no systematic knowledge that can be encoded into a knowledge base so as to produce an expert system for making such recommendations. Hence, the idea explored in this section is concerned with garnering knowledge pertaining to cost-sensitive learning algorithms. In applying learning, various examples are needed, where each one consists of some features and a class. We can generate examples by applying a learning algorithm to some datasets, and recording how well it works. Given the features of the datasets, such as skew and size, etc.—which are notably termed meta-features—we can apply various learning algorithms in an effort to induce knowledge relating to the performance of a particular cost-sensitive learning algorithm. In order to explore this idea further, the key questions needing to be addressed include:

1. What features would be best in characterising the data?
2. Feature-selection can play an important role, prior to applying a learning algorithm, so how can this be incorporated within the meta-learning process?
3. How can we learn which cost-sensitive approach is suitable for a particular dataset?
4. Given a wide range of learning algorithms, which should be used for learning meta-knowledge?
5. How can we use meta-knowledge to make a recommendation?

Figure 3-1 presents the proposed meta-learning process aimed at addressing these questions.

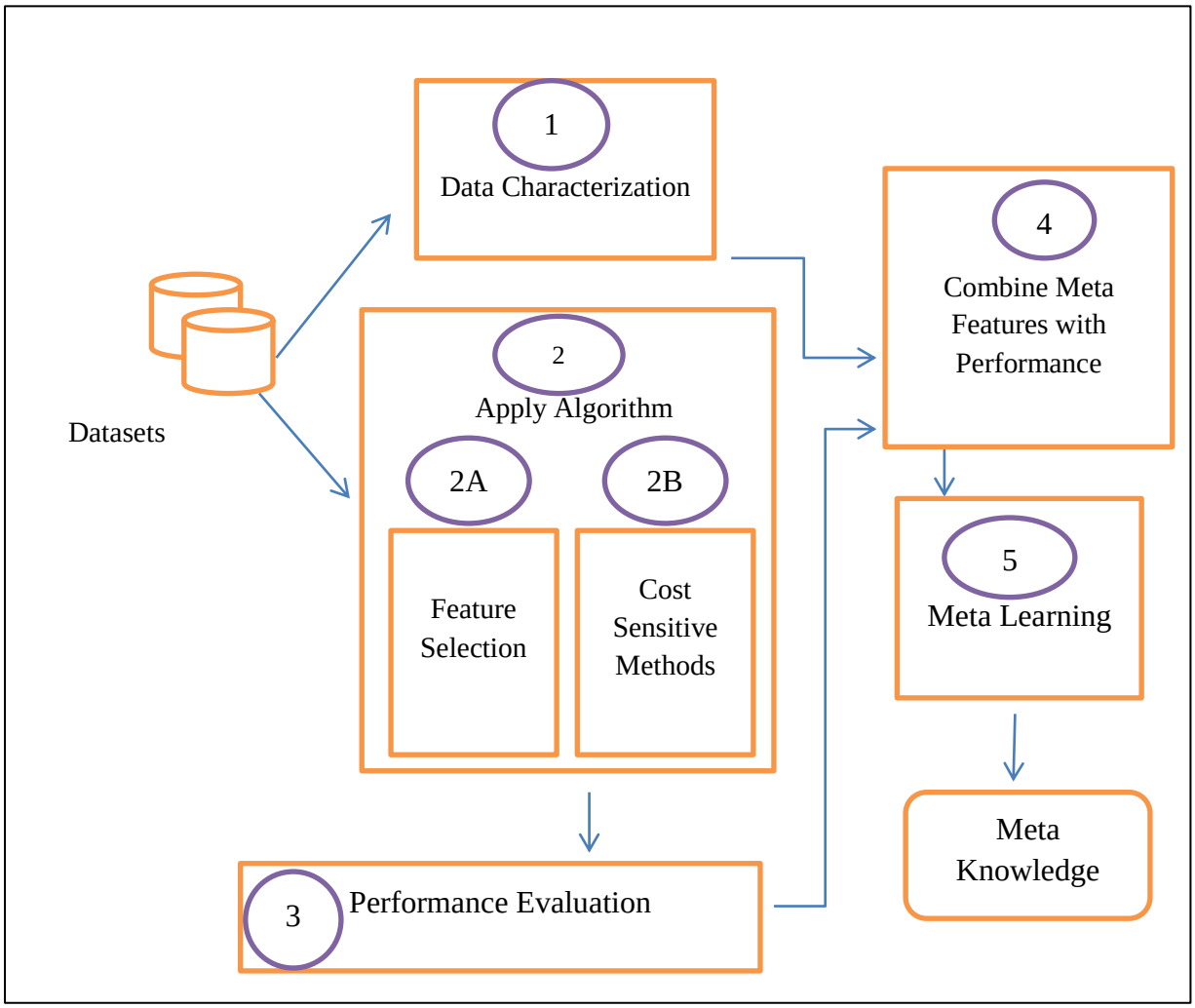

Figure 3-1:Meta-Learning for cost sensitive learning

The following describes the main components presented in Figure 3-1.

1. **Data Set Characterisation**: This involves applying different data characterisation techniques to a given example of datasets in an effort to understand the nature of the

data, including simple, statistical and mathematical measures. This phase results in the meta-features that will be fed to the next stage of the process (Box 1 of Figure 3-1)

2. **Apply Algorithm**: This involves applying different learning and feature-selection algorithms:

   A. **Feature Selection**: This applies different feature-selection methods and search strategies in the process of building knowledge on which feature-selection methods will suit specific datasets. The name of the features selection algorithm used and the results of applying different base learners on the data selected, with the dataset characterisation, forms an example. The table of examples produced provide input for a learner, which provides meta-knowledge pertaining to feature-selection methods, their performance and cost.

   B. **Cost-Sensitive Learning:** This involves applying different cost-sensitive and -insensitive algorithms. The name of the algorithm and results, together with the dataset characteristics, form an example. The table of examples provides input to a learner, which results in the meta-knowledge predicting the classifier performance and costs for a given dataset.

3. **Performance Evaluation**: This evaluates the performance of the classifiers using various measures, such as accuracy and misclassification costs.

4. **Combine Meta-features with Performance**: The performance results, along with classifier name and feature-selection methods, as well as cost-sensitive methods, all are combined with meta-features for all datasets.

5. **Meta-Learning Process**: The main goal of this phase is to learn about the performance of learning algorithms, which includes the result of all previous phases: datasets of examples with their characteristics (meta-features), different feature-selection strategies, and all sets of algorithms, along with their performance after applying cost-sensitive and -insensitive algorithms, are all fed to the learner with the aim of developing meta-knowledge about the performance of the algorithms.

In Base learning process the following classifiers are used:

- J48: decision tree classifiers is used to learn a classification rule which concludes the value of dependent attribute given the value of independent attributes, decision tree is a very popular classification techniques which is used in most of the classification problems, the benefits of using decision tree are as follows (Bhargava, Sharma, Bhargava, & Mathuria, 2013) :
    - Easy to be used
    - Easy to be interpreted by the end user
    - It can handle the missing values
    - High performance with quick results
- Neural Network: neural network is a classifier that convert some input to output, neural network structure is made up of numerous interconnected units each unit consists of input/output that implements a certain function with a given weight to produce the classifier output (Hecht-Nielsen, 1989).
- Naïve Bayes: probabilities classifier based on Bayes theorem with strong independent variables, it is used to predict a specific class membership probabilities based on others, such as the probability that a given sample belongs to a particular class(Lewis, 1998). Despite its simplicity, impressive results can be achieved using it (Hall et al., 2009; Jordan, 2002; Patil & Sherekar, 2013).
- OneR: it is a simple classification algorithm that generate one rule for each predictor then select the rule with highest accuracy, it is easy to be used, produces rules that are simply interrupted (Devasena, Sumathi, Gomathi, & Hemalatha, 2011; Kabakchieva, 2013).
- Part: it is a simple, yet accurate classifier for generating a PART decision tree by building a partial J48 decision tree and makes the "best" leaf into a rule (Devasena et al., 2011).
- ZeroR: it is classifier that predicts the majority class (if nominal) or the average value (if numeric) (Hall et al., 2009), although there is no predictability power in ZeroR, it is useful for determining a baseline performance for other classification methods (Nasa, 2012).

Figure 3-2 and Figure 3-3 present more detailed diagrams of the meta-learning process for feature-selection and cost-sensitive learning, respectively, showing how the performance of different methods, combined with the features of the data, are used to produce meta-knowledge. Thus, as illustrated in Figure 3-2, various datasets, combined with their characteristics, are taken as input. Feature-selection methods are selected, and a base learning method is chosen and applied, with its performance then evaluated. The features of the dataset, methods used and performance are all combined to form the examples and learning algorithm applied to learn the meta-knowledge.

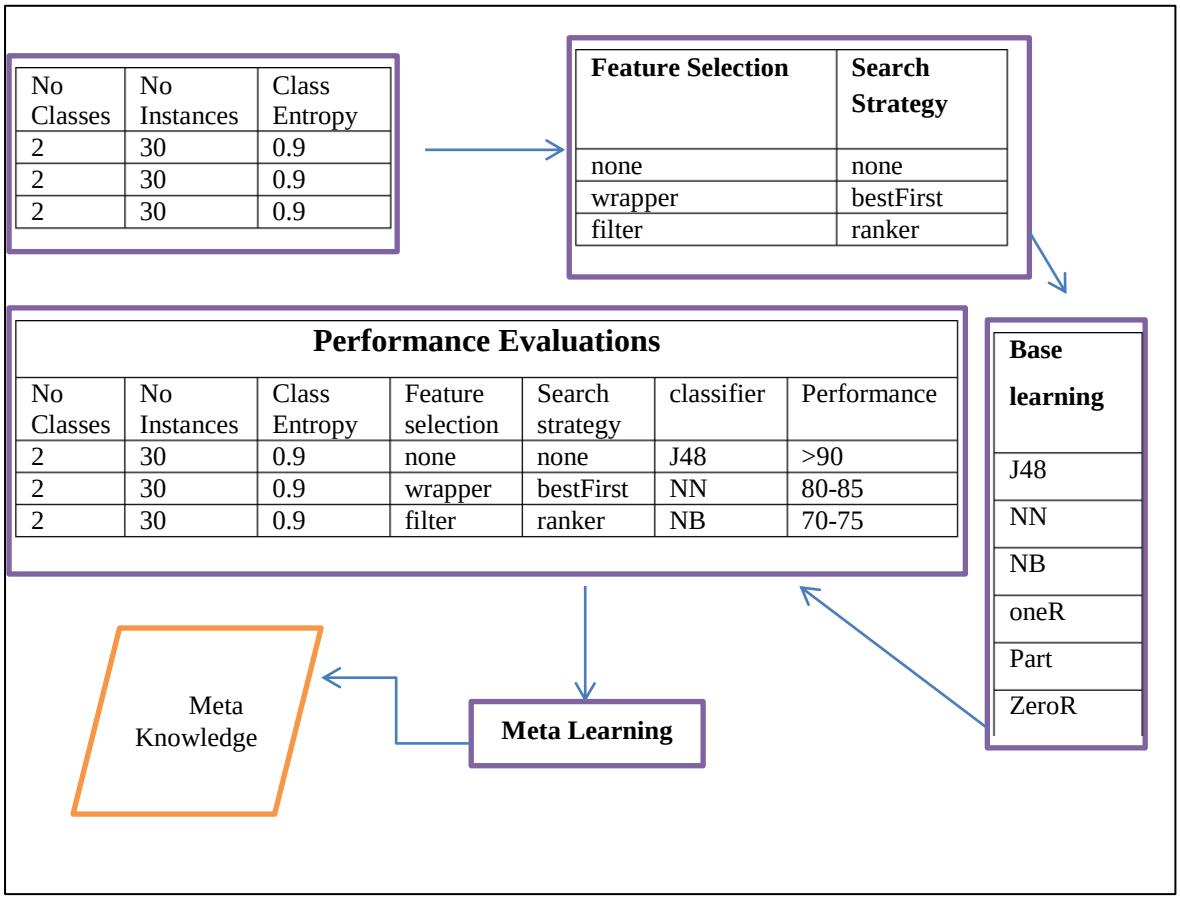

Figure 3-2: Feature selection meta-knowledge development

In Figure 3-3, some datasets and their characteristics are taken as input. Cost-sensitive wrapper methods are selected (none, sampling, minimum expected cost and bagging), and a

base learning method is then selected and applied, and its performance evaluated. The wrapper methods, base learning methods used and performance are all combined to form the examples, and a learning algorithm is applied to learn the meta-knowledge.

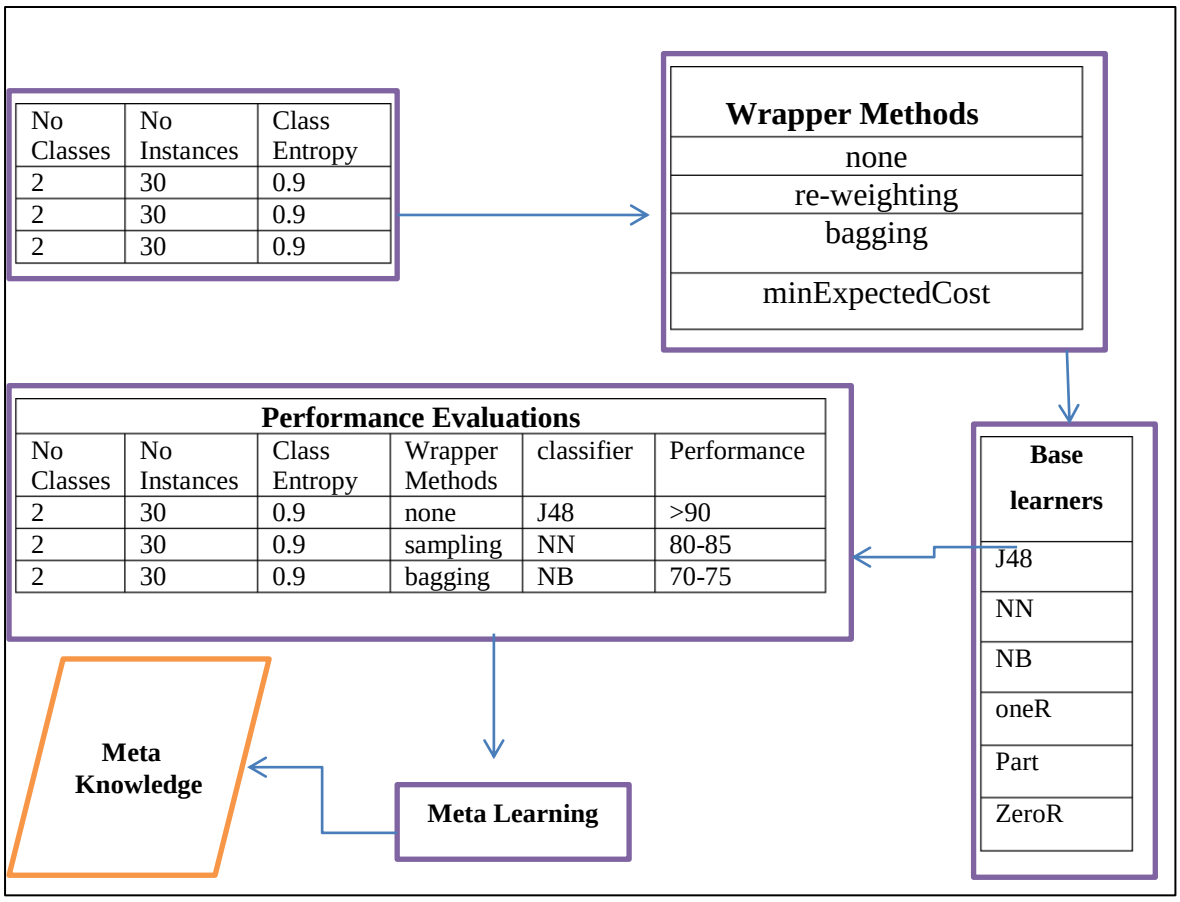

Figure 3-3: Cost sensitive meta-knowledge development

The following Subsections provide a more detailed explanation of these phases, and accordingly address the questions raised above.

3.1.1 Data Set Characterisation

This subsection addresses the question: What features would be best in characterising the data? (Q1 P46). The aim is extracting knowledge relating to when a particular learning algorithm performs better on a specific task. This is not trivial, as pointed out by Rice (1975):

‘*The determination of the best (or even good) features is one of the most important, yet nebulous, aspects of the algorithm selection problem’*.

Towards this aim, the first stage is identifying the features that are the process of discovering the dataset characteristics through the use of simple, theoretical and information-based measurements. This section introduces these methods and accordingly explains each characteristic with different examples.

Simple Measurements

These descriptors of the dataset provide a very general and simple measure of complexity or the size of the data problem, and include the following:

- ***Number of Instances:*** This is the total number of observations in the entire dataset. This is an important characteristic in supervised learning owing to the number of observations used in building a model effects its inductive bias (Baxter, 2000), and can result in model overfitting (Dietterich, 1995). It is known that we will have a better, generalised learning model when we use a larger number of instances in the learning process (Vilalta & Drissi, 2002); however, this factor is highly affected by other factors, such as any bias in the dataset and number of features used. Some studies point out that it is not only the degree to which data is skewed that impacts the learner performance towards rare classes, but also the scale of the sample, which affects leaner reliability (Sun et al., 2007). Some experimental observations reported in Japkowicz & Stephen (2002) indicate that, as the size of training set increases, the error rate caused by skewed data decreases.

- ***Number of Attributes:*** The number of attributes is an important factor in building a specific model. A minimum number of attributes should be used when applying inductive learners, predominantly owing to the fact that irrelevant attributes have an adverse effect on the learning process and predictive power (Yu & Liu, 2003).

- ***Number of Classes:*** The total number of classes is an important factor in describing the data. A large number of classes cause concern in regard to how many instances

may be available for training on each class. If the number of instances mostly belongs to a certain class, the data then becomes imbalanced, causing biases.

Figure 3-4 provides some of the simple measurement variation over different datasets.

Figure 3-4 shows, there are many datasets with two classes, some of which have more than two classes—as in the case of chess and glass—whereas the number of attributes varies from 61 attributes (sonar datasets) to 5 attributes (iris, diabetes). The number of instances varies from 24 instances (contact lenses) to 958 (tic-tac).

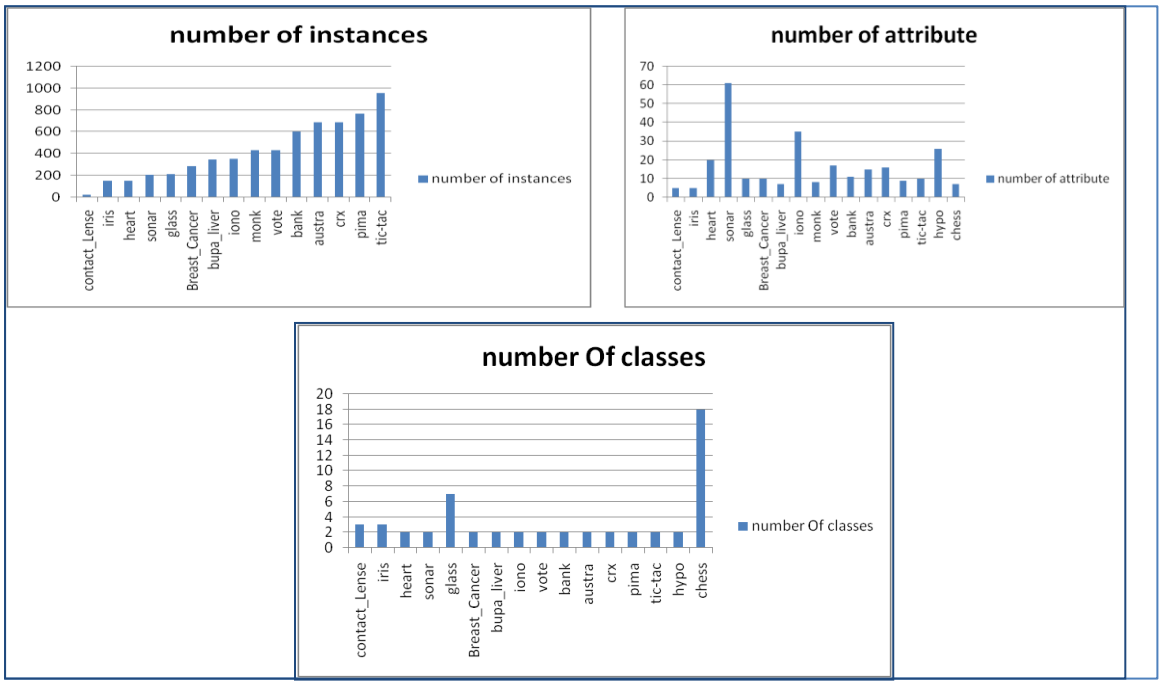

Figure 3-4: Simple measurements for Different datasets

Statistical Measurements

Statistical measures provide a deeper characterisation of the data, and include the following:

- ***The Skewness:*** Skewness describes the deviation from a normal distribution. This measure provides an indication of symmetry in the data; in other words, if the data looks the same from the right side and left side, it is symmetric, below presents the measure.

$$\boldsymbol{Skewness} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \frac{(\boldsymbol{Y_i} - \boldsymbol{\bar{Y}})^{\boldsymbol{3}}}{(\boldsymbol{n-1}) * \boldsymbol{S^3}} \qquad \textbf{(3-1)}$$

Where $\bar{Y}$ is the *mean value*, *S* is the *standard deviation* and *n* is the *number of data points in the dataset.*

Zero skewness means that the data is symmetric; negative value means it is biased to the left; and positive means it is biased to the right. This measurement is used in several meta-learning studies, such as the StatLog project (King et al., 1995) and Metal (Brazdil et al., 2003). Data skewness is a very important factor in cost-sensitive learning owing to the fact that data that are skewed tend to classify each instance to the more dominant class

- ***Class Ratio:*** This measures the degree to which data is balanced by calculating the data distribution of the number of instances for the dominant class over the total number of instances for all classes. This is an important characteristic owing to the fact that imbalanced data can cause the generation of unreliable classifiers. This is because learning a model from imbalanced data tends to generate a model that classifies every instance as dominant class without affecting its performance. For example, consider the case of cancer diagnosis: if the percentage of cancer patient is 7%, then a classifier that classifies all patients as negative (not sick) has 93% accuracy. As mentioned previously, in some applications, the cost of misclassification for a rare class is much higher than the cost of misclassification in the dominant class. For this aim, many researchers have studied the effect of imbalanced data. This ratio is calculated using ***(3-2)***(Elkan, 2001):

$$\textbf{Class Ratio} = \frac{\boldsymbol{number\ of\ instances\ for\ the\ dominate\ class}}{\boldsymbol{total\ number\ of\ instances}} \qquad \textbf{(3-2)}$$

When the data is not biased, for a 2-class problem, the expected example ratio then is 0.5. The more it departs from this, the less balanced the data.

- ***Cost Ratio:*** In cost-sensitive classifiers, the cost ratio plays a critical role owing to the fact it reveals the importance of one class over another. The cost ratio reflects the importance of a specific misclassification error from one class over a misclassification error from another class: for example if the cost of wrongly classifying a sick patient as healthy in a medical diagnosis problem is twice that of misclassifying a healthy patient as sick, then the cost ratio in this case would be 2. In two class problems, the cost ratio is defined as the cost of false negatives over the cost of false positives. In multi-class problems, the cost ratio is more complicated: the cost matrix involved is not the usual 2 x 2 cost matrix presented when solving two class problems (more details on how to calculate the cost ratio in the case of multi-class problem is given in Section 3.1.3).

Information Theoretic Measures

The measures used here are motivated by information theory and applicable for discrete values in addition to continuous values. The following two measures are used:

- ***Class Entropy:*** This is a measure of homogeneity of the set of examples in specific datasets. Given a set $\boldsymbol{S}$ of positive and negative examples of some target concept, the entropy of set $\boldsymbol{S}$ relative to this binary classification is given by ***(3-3)***(Fayyad & Irani, 1992):

$$\boldsymbol{E(S) = -P(E)Log_2P(E) - P(N)Log_2P(N)} \qquad \textbf{(3-3)}$$

Where $P(E)$ is the probability of a positive example and $P(N)$ is the probability of negative example. In machine-learning, it has been shown that entropy is related to information in the sense that, the higher the entropy or uncertainty of some data, the more information is required in order to completely describe that data (Quinlan, 1986; Rendell & Ragavan, 1993).

- ***Conditional Entropy:*** Measures the amount of information needed to describe the outcome of a random variable *Y* given the value of *X*, where *Y* is the attribute and *X* is the instance(Wang, Yu, & Yang, 2002). Conditional entropy is calculated using the following:

  1. The first step to calculate *H (Y|X)*, which is the entropy of *Y* given the *X*, is by calculating the sum of each column of data.
  2. Calculate the weight of each column as the column sum divided by the total number of observations *N*.
  3. Calculate the entropy for each column using the entropy function in ***(3-3).***

  Conditional entropy by columns is calculated using ***(3-4)***:

$$\boldsymbol{H(Y|X)=\sum_{x\in X} P(x)H(Y|X=x)} \qquad \textbf{(3-4)}$$

  Conditional entropy is a critical data characteristic that emphasises the relationship between the attributes and the class owing to the fact it gives the entropy of the class variable given the value of the attribute X. It has been used in many meta-learning projects, such as statLog and METAL.

Figure 3-4 presents class skewness, entropy, and conditional entropy for the datasets used in this research. As shown, the datasets are selected with different class skewness, ranging from positive or negative, different class entropy, and different conditional entropy.

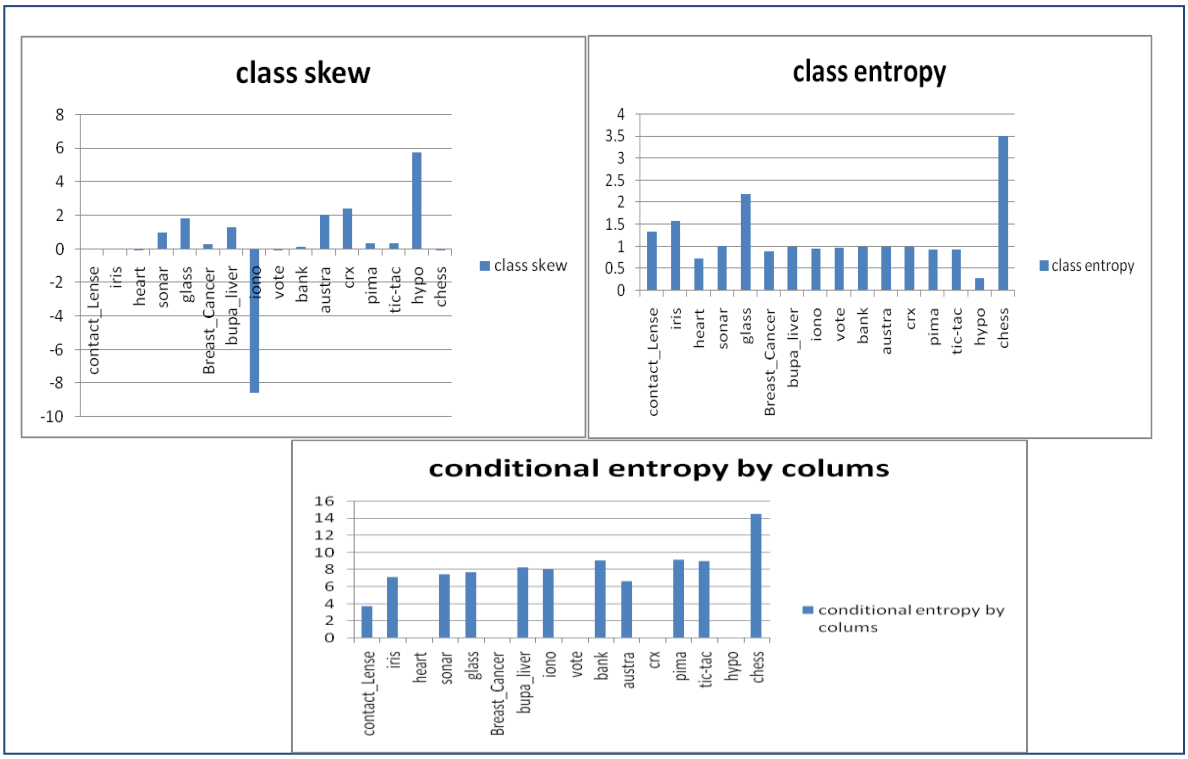

Figure 3-4: Information theoretic measures on different datasets

3.1.2 Base Learning Process

Section 3.1.1 above described the data characterisation box 1 in Figure 3-1. This section describes the next step, which involves the application of different sets of base learning algorithms. Section 3.1.2.1 describes the feature-selection methods considered, whilst Section 3.1.2.2 describes the cost-sensitive learning methods.

3.1.2.1 Feature-Selection

Some datasets are very large and contain features that are either irrelevant or that have no impact on the learning process. It is well known in the machine-learning community that irrelevant features can have a negative effect on a learner's predictive power, and has some disadvantages (Almuallim & Dietterich, 1991; Kira & Rendell, 1992; Vanschoren, 2010). Irrelevant features can cause greater computational demand, especially if there are a lot of features requiring additional processing.

In the literature, a wide range of approaches are defined (see Section 2.3), albeit without a solid answer for the question that arises here:

Amongst those different feature-selection approaches, which one should I use for a particular dataset? (Q2 P46)

Considering the importance of feature-selection, this research includes use of meta-learning for feature-selection; thus, meta-learning also explores the impact of feature-selection methods based on dataset characteristics, with the aim of finding which algorithm performs best if a specific feature-selection method is applied for a specific task. In this research, a variety of feature-selection methods are used with the aim of learning meta-knowledge that predicts which feature-selection method is most suitable for a specific dataset. An overview of feature-selection methods—and related searching methods—was presented in Section 2.3. There are two different approaches, namely wrapper methods and filter methods: in the case of the former, the inducer itself is used to evaluate which subset of the feature is more predictive, eliminating those that are less predictive or irrelevant; in the latter, data is independently characterised from the learning algorithm, depending on the data characteristics, its correlations and dependencies.

There are different search strategies defined in the literature: greedy search, which performs either forward or backward search by starting either with all attributes in the case of backward search, and eliminates those attributes that reduce or otherwise have no effect on classifier performance, whilst forward search starts with no attributes, and adds attributes one by one that increase classifier performance until some stopping criteria are met (Kohavi & John, 1997). On the other hand, rankers rank a set of attributes according to their evaluation values, either gain ratio, entropy, etc.

3.1.2.2 Cost-Sensitive Meta-learning

Chapter two described cost-sensitive learning methods. In this section, the methods selected for inclusion in this study are briefly summarised. In general, there are two types of method for developing cost-sensitive classifiers: direct and indirect. In a direct approach, an algorithm is altered or developed in such a way so as to take into account costs; in an indirect approach, wrappers over accuracy-based algorithms are used to convert accuracy-based classifiers to produce cost-sensitive classifiers.

In this research, we use the wrapper approach with a wide variety of base learners. The advantages of using a wrapper approach include the amendment of the data input space rather than making any changes in any part of the classifier. In addition, it provides more flexibility to use any new learning algorithm without concerning its internal behaviour. The following cost-sensitive methods are used:

- Minimum expected cost: In accuracy-based learning, the aim is minimising error by producing a classifier that minimises the number of misclassified instances. In general, each misclassified instance is counted according to its occurrence; therefore, if 50 instances are misclassified in a dataset that contains 500, then the error rate is 10%. In contrast, the optimal class in cost-sensitive learning is the class that provides the minimum expected cost as opposed to the minimum misclassification number of instances by taking into account the cost of misclassification. This idea of optimal cost produced is by Elkan (2001). The expected costs are calculated using ***(3-5):***

$$\sum p(j|x)C(i,j,x) \qquad \textbf{(3-5)}$$

C (i,j,x) is the cost of misclassification example x as class i, when the true class is j, p (j|x) is the probability of classifying instance x as class j

Accordingly, the cost-sensitive classifier tends to classify each instance to the class that it gives its minimum misclassification cost.

- Re-Weighting: In the re-weighting method, each instance is allocated a weight that reflects its importance in the learning process, meaning each instance in the training set is multiplied by its misclassification costs. In this way, the costly class is given more weight owing to the fact that the misclassification of costly class is more severe (carries heavier penalties) than misclassifications in a dominant class. This leads to generating a model that is less likely to misclassify the costly instances.

- Bagging (MetaCost): This is one of the algorithms that utilises bagging with resembling (Domingos, 1999). In MetaCost, the training set is divided into different samples (sample with replacement), with a specific base learner then applied on each sample in order to generate different models. The decision made by each model is

combined so as to produce the aggregate final prediction. Subsequently, each instance then is assigned a label that provides its minimum cost—even if it is not its true label. The re-labelled examples are re-processed through the insensitive cost learner (accuracy-based learners) in an effort to produce a sensitive cost classifier.

3.1.3 Performance Evaluation

This is very important process in meta-learning, it involves evaluating the result of applying different base learners (cost-sensitive and insensitive) on a given datasets. In this research, accuracy and cost are used to evaluate the performance of the learning process by using 10 folds cross validation. The idea is to have a 10 % subsets of the dataset called the validation set, nine are used for building the model and one is used for testing, then each subset that is taken for testing is retuned back to enter a training process and another subset is taken for testing. As a result the average of the ten trials is taken as the classifier performance.

In our experiments different costs are generated using different cost matrix values, the value of the cost matrix is taken from 1 to 4 in round for each class and the cost is calculated by multiplying the number of misclassified instance from the cost matrix by the cost obtained from generated cost matrix using the following equation (Elkan, 2001):

$$\boldsymbol{Cost} = \frac{\sum \boldsymbol{P(i,j)} * \boldsymbol{C(i,j)}}{\boldsymbol{N}} \qquad \textbf{(3-6)}$$

Where $P\ (i,j)$ is the probability of classifying instance j into i, $C(i,j)$ is the cost of misclassifying an instance j into i, N is the total number of instances.

3.1.4 Meta-Learning Process

As mentioned previously, meta-learning is the process of learning about the learning process, which can be achieved using the accumulative experience learned from previous applications of different learners on datasets with a wide range of characteristics, with these then used to help recommend future uses. In meta-learning, the meta-features are computed for all datasets

(Step 1 of Figure 3-1), then combined with the result of accuracy and cost from the empirical evaluations of different base learning classifiers on those datasets, and accordingly provided as input to the meta-learner (Box 5 in Figure 3-1) in order to create the meta-knowledge. This meta-knowledge can then be used to select one or more algorithms given the characteristics of the datasets under analysis. The question that arises here is:

Which learning algorithm should be used for this learning task? (Q4 P46)

In theory, several alternative methods could be employed in an effort to adhere to this recommendation:

- K-nearest neighbour: Given a new dataset, one could compute the k-nearest neighbour based on the characteristics of the new dataset, and accordingly use this as a basis of a recommendation.
- Neural networks: One could train a neural network, which, when given the characteristics and a proposed learner, predicts accuracy and cost.
- Decision tree learners: A decision tree in which each internal (non-leaf) node is labelled with an input feature, given a new dataset, allowing one to predict the accuracy and cost of the given dataset by simply tracing through the given tree checking its attributes until a leaf is reached.

All of the above-cited methods could have been used in this research; however, an important aim of this research is developing meta-knowledge that could be transparent and comprehensible. Hence, in this research, the decision tree induction is used as a meta-learner. More specifically, J48 is used because it is readily available, easy to use, and produces rules which are comprehensible. J48 is used as meta-learner that predicts the performance of a given classifier for a specific dataset according to its computed features and accordingly is linked to different base learners' performance. The meta-knowledge developed using J48 is converted into rules, such as:

If classEntropy<1.5 && number of Instances < 30, & number of classes =2,

then naiveBayes classifier accuracy is >80.

If feature-selection = wrapper, search strategy =bestFirst.

The meta-learning procedure described above has been implemented in the WEKA (Hall et al., 2009) system, WEKA is open source software which is popular for machine learning algorithms to apply data mining tasks, WEKA contains a collection of visualization tools and algorithms for data mining tasks, and data analysing using different predictive models, with graphical user interface for easy accessing those machine learning. The results of an evaluation are presented in chapter 4. The next section will tackle a specific problem faced in the development of meta-learning, which is the problem of improving the speed of learning good-meta knowledge.

3.2 Development of Active Learning Based on Clustering

The process of developing meta-knowledge, as described above—namely the generation of different examples by applying different classifiers and evaluating their performance—is an important first step in this research, but could prove time-consuming, particularly for large datasets. More specifically, not all datasets will contribute the same amount of new knowledge: for example, if a new dataset is very similar to previous datasets, there would be little benefit in using it if the existing predictions are accurate, and one might get more knowledge from another dataset. This problem of seeking out the right examples for learning is not new, and several authors have proposed the use of active learning, such as Cohn *et al*. (1994), Lindenbaum, Markovitch, & Rusakov (2004) and Tong & Koller (2002). Active learning has emerged in some domains, where labelled data is not available or otherwise requires time and effort to be labelled.

The central problem of active learning is how to choose data that will be labelled (Roy & McCallum, 2001; Settles, 2011) . The new data is determined using different query strategies:

1. ***Uncertainty Sampling***: In this framework, the query seeks the data that the learner is uncertain (or least certain) on how to classify (Cohn et al., 1994; Huang, Jin, & Zhou, 2010). Through this technique, data which has the maximum uncertainty measure is selected to enter the learning process. It is claimed that, when data uncertainty is high, the classifier does not have sufficient knowledge for its classification. As a result, including these data will improve the learning process (Settles, 2011). This approach

is also referred to as *closest-to-boundary* criterion (Nguyen & Smeulders, 2004) owing to the fact that the data that are nearest to the current learner boundaries are often the most difficult to classify with certainty, and therefore they may produce new knowledge.

2. ***Query-By-Committee***: Through this framework, more than one classifier is used. Each learner is asked to vote on the labelling of the new data, where the data chosen is the data that has the largest number of disagreements in terms of how it should be labelled (Settles, 2011; Seung et al., 1992). This concept follows the uncertainty sampling approach described above, where data around the boundaries are taken to be the most informative data for labelling.
3. ***Expected Error Deduction***: This depends on the estimation of how much classification error is likely to be reduced using the new nominated dataset (Roy & McCallum, 2001)
4. ***Active Learning with Clustering***: All of the above methods ignore the prior data distribution in choosing the first pool of dataset to be labelled. Hence, some researchers have suggested active learning with clustering that takes into consideration the prior data distribution e.g.(Nguyen & Smeulders, 2004; Urner et al., 2013).

The approach adopted in this thesis is to use active learning based on clustering with the aim of improving accuracy more rapidly as new datasets are added (points 3 and 4). As mentioned previously, most active learning approaches (namely points 1–3) ignore data distribution by applying a specific learner or set of learners on a small number of samples with the aim of labelling under the assumption that those chosen samples are good representatives of the whole dataset. In most cases, this assumption is violated owing to the small number of samples taken for labelling and a large number of unlabelled data (Zhu et al., 2008b). A potential solution for this problem is finding clusters of dataset that could represent the whole data space.

Active learning based on clustering is motivated by the idea of establishing good representative samples for the potentially wide range of datasets needed to create the meta-knowledge in the meta-learning process. In applying this approach, the following questions arise:

- How can we assess the cluster ‘goodness’?
- In considering that a ‘strong enough’ cluster is generated, would all data in the same cluster behave in the same way to a specific learner?
- Can we unify them in one cluster label that reveals their performance?

In an effort to answer these questions, the concept of weak and strong clusters is introduced, along with the development of an active learning algorithm for meta-learning. Figure 3-5 illustrates the idea of strong and weak clusters, plus sign refers to the example that has correct classification result and minus for the example that has wrong classification result, the figure shows various possible clusters that have both positive and negative examples:

- Cluster 1: All positive examples are concentrated around the centre. The further away we are from the centre, the more negative examples are found (because they may belong to another cluster). This cluster has good predictive power and can be considered as a *strong cluster*.
- Cluster 2: This cluster has a mixture of negative and positive examples spread near and far from the centre. This cluster is not good because no decision boundaries are fixed, and it has low predictive power owing to the fact that data with similar patterns are classified differently, and hence is a *weak cluster*.
- Cluster 3: Most of the negative examples are concentrated around the centre, whilst most of the positive examples are spread around the cluster boundaries, meaning this cluster is poor, and we expect that the positive examples on the boundaries should be belong to a different cluster.

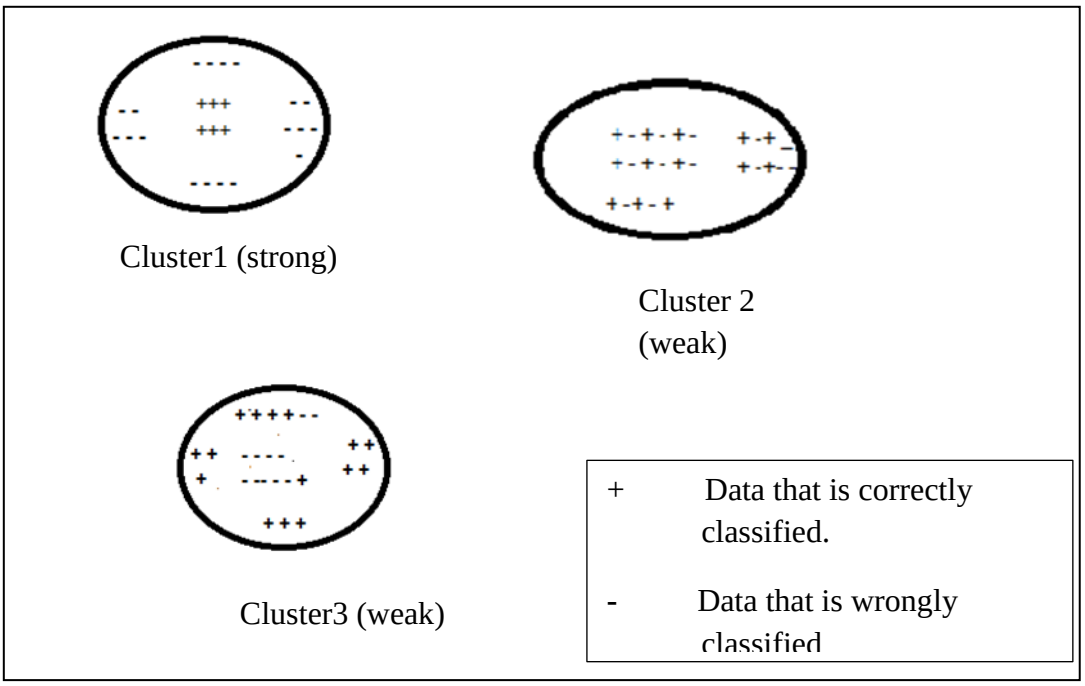

Figure 3-5: Examples of strong and weak clusters

As the examples in Figure 3-5 illustrate, there are two important points:

1. Data that are close to the cluster centres are more representative.
2. Data samples in the same clusters are likely to have the same classifier performance, meaning the closer the two instances, the less likely they are to have different performance (Urner et al., 2013) ; this will help us in assigning one label to each cluster.

As described above, our aim is finding how far each data is from its cluster centre because we believe that (in the case of strong cluster) data in the cluster's centre are more representative and should have the lowest error (more positive examples). Through this approach, we use a statistical measure, known as the z score, to measure the distance of a specific dataset from the cluster centre. The z score indicates the amount of standard deviations of an element from the mean. A dataset with higher z score is nearest to the cluster boundary. The formula for calculating a z score is given in ***(3-7)*** (Abdi, 2007) :

$$z = \frac{x - \bar{x}}{S} \qquad \textit{(3-7)}$$

Where x is the attribute value, $\bar{x}$ is the mean value, and S is the standard deviation.

In order to assess whether a cluster is strong or weak, we first divide each cluster into three z score zones (Zone 1, Zone 2 and Zone 3), where each zone highlights the distance from the cluster mean. In order to achieve a strong cluster, we expect to have more positive examples in Zone 1 and more negative examples in Zone 3. According to this, creating a good cluster can be done through minimising the error prediction for each cluster by seeking more data in the cluster's centre that are positive.

Thus, searching for a new datasets will be done in the following cases:

1. There are no datasets in a specific zone, which means no data represents this type of knowledge.
2. The error rate is increasing whilst we are moving toward the cluster centre, which means the cluster is not good enough in terms of prediction.

The question that arises here is how a data character that goes into a specific z_score in a specific cluster can be chosen. The minimum distance between dataset characters and cluster z_score is used to find the dataset that fits into the specific cluster z_score.

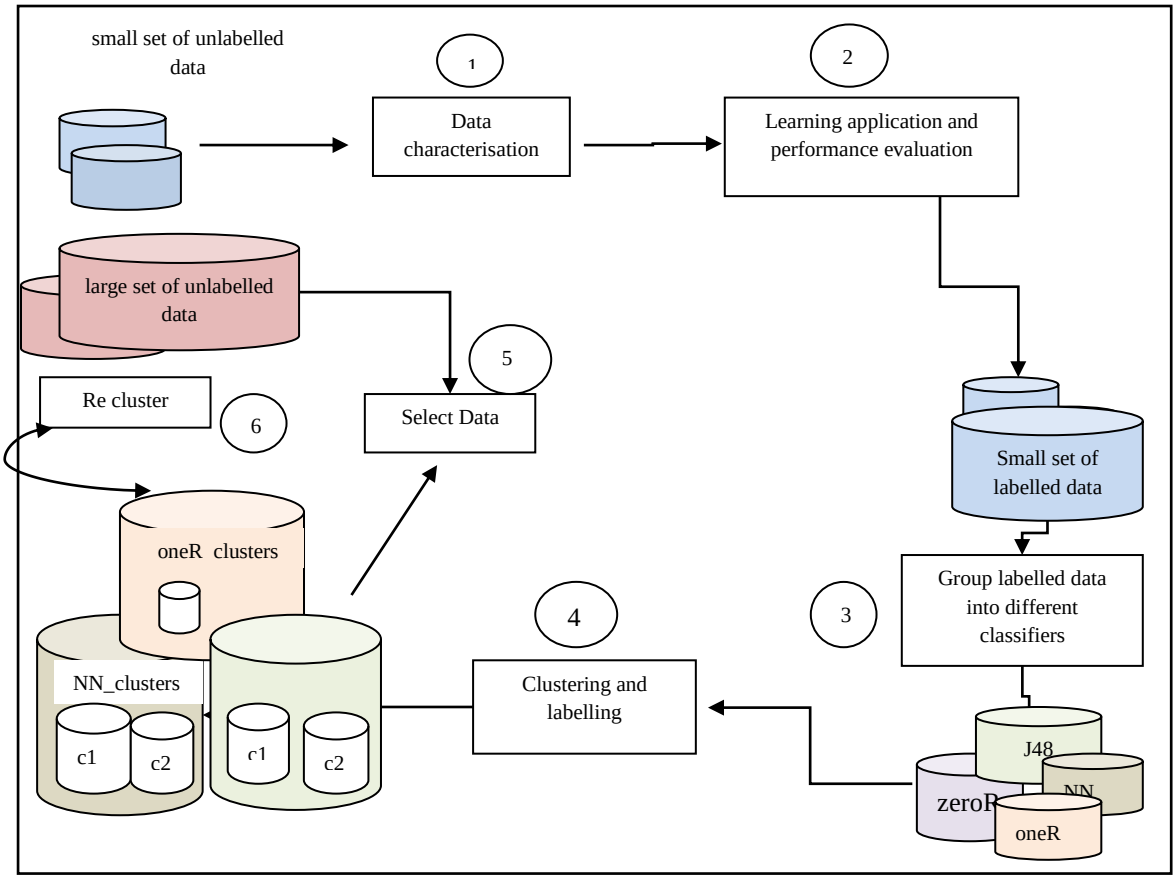

Figure 3-6: Active learning based on clustering

The above considerations lead to the active learning process, as presented in Figure 3-6, which includes the following steps:

1. Dataset characterisation: Each dataset is characterised using different simple mathematical and statistical methods with the aim of clustering each dataset with its natural groups.
2. Learning application and performance evaluation: Different base learners are applied to the characterised data with the aim of evaluating the performance of different classifiers, such as applying J48, naive Bayes, neural network, oneR, zeroR. Each dataset has evaluation accuracy and misclassification cost resulting from applying different classifiers. All the results are evaluated using 10-fold cross-validation.

3. Divide each labelled datasets into its classifier groups: The datasets are grouped according to the learning algorithm used (J48, naïve Bayes, neural network, part, and zeroR).
4. Clustering: Each classifier's group is clustered based on the data characterisation. Subsequently, each cluster is given an initial label that is the average of its datasets labels. As shown in Figure 3-7, J48 group clusters are generated by applying J48 learner on data, with the average label of all datasets within a specific cluster then assigned as 'cluster label', with the average categorised in ranges (for example: 50–60 in Cluster 1 and 80–90 in Cluster 2 for J48 accuracy group, and 5–10 for Cluster 1 and 30–40 for Cluster 2 for Naive Bayes cost group). As a result, each classifier's group has its own clusters, each representing the expected results of data in that cluster. Figure 3-8 depicts two different clusters showing the different results and data characteristics; thus, if data is in Cluster 1—which has number of classes = 2, number of attributes = 14, class skew = 0.5—then J48 can be expected to produce better results than if it was in Cluster 2—where number of classes = 3, number of attributes = 5, class skew = 2.

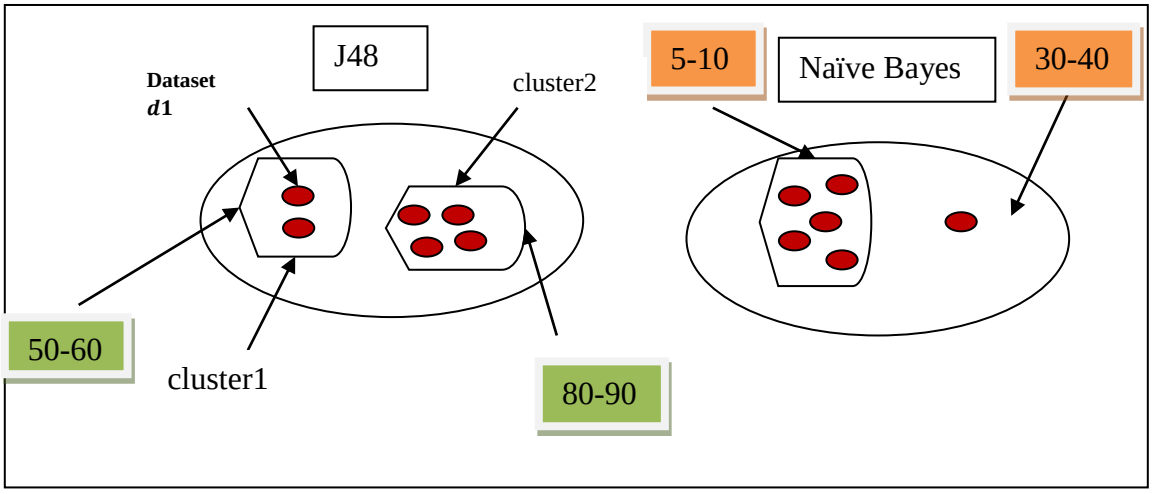

Figure 3-7: Illustration of classifiers specific clusters for J48 and naïve Bayes

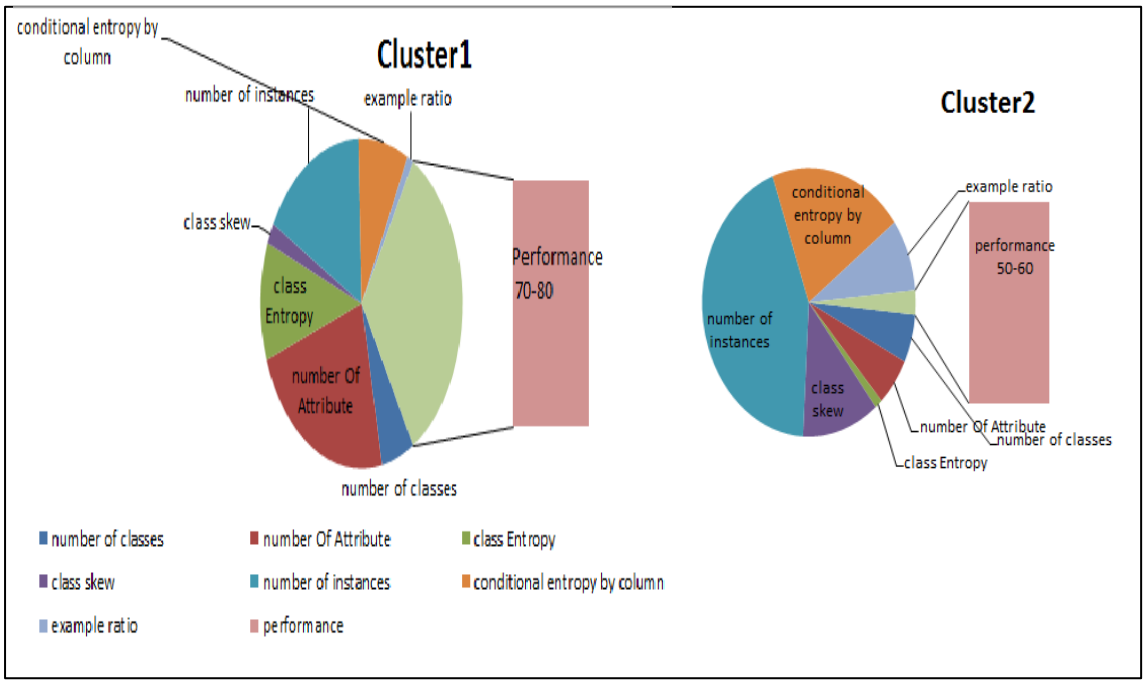

Figure 3-8: Illustration of two clusters and their characteristics, both for J48 and naïve Bayes

For each learning method (J48, naïve Bayes, neural network, Part, oneR, and ZeroR)

- Evaluate quality of *clusters* based on z_score (described previously in this section)
- Determine the characteristics of the data needed to improve the predicted accuracy of clusters.
- Search for new data
- Add the selected data into labelling datasets

5. Select data: More informative data for each cluster is selected by re-clustering, which is done with the aim of finding better clusters after new data is added.
6. Re-cluster with the new data added.

The above outline procedure is presented in greater detail as per the algorithm below:

- Figure 3-9 presents the top level of the active learning algorithm,
- Figure 3-10: The query formulation steps,
- Figure 3-11 presents dataset determination steps, and
- Figure 3-12 presents dataset search steps.

***ALBC*(X_i, X_u)**

// Active learning based on clustering algorithm

Input:

- X_i *set of small unlabeled dataset*
- X_u : pool of large unlabelled datasets

Output:

- X_{Mc} set of informative datasets for each classifier c, where $X_{Mc} \subseteq X_u$ and $X_{Mc} \neq \{\}$.

Procedure

1. Characterise each data set X_i into X_{ichar} :

   X_{ichar}={(x,noAttributes(x),noClasses(x),classSkew(x), noOfInstances(x) ,classEntropy(x) , conditionalEntropy(x) , performance (x, c), classifier(c) | x ∈ X_i, c ∈ { j= J48, NB, NN , oneR , part, ZeroR }.

   Where x is the dataset, noAttributes returns number of dataset attribute, noClasses returns number of dataset classes, classSkew returns dataset class skew, noOfInstances returns number of dataset instances, classEntropy returns dataset class entropy, conditionalEntropy returns dataset conditional entropy.

2. Group the characterized datasets X_{ichar} in subgroups, one for each classifiers:

   G = {g_{J48},g_{NB}, g_{NN}, g_{oneR}, g_{zeroR}, g_{part}} such that g_i subset of X_{ichar}

3. For each classier *c* member of { J48, NB, NN , oneR , part, ZeroR } do

   3.1 X_{Mc} = Select query (g_c, X_u);

4. EndLoop

5. Return X_{Mc}

Figure 3-9: The top level of the active learning algorithm

Select query (g, X_u).

Input:

- g : is set of characterization obtained for each dataset associated with labels (accuracy and cost) grouped into its classifier c
- X_u : pool of unlabelled datasets

Output

- Set of informative data X_M , where $X_M \subseteq X_u$ and $X_M \neq \{\}$.

Procedure

1. Cluster g into k ***clusters*** , where k is the number of clusters , $k > 0$
   // evaluate each cluster goodness using z_score value and cluster prediction error rate //
2. For each *cluster* $c_j \in$ ***clusters*** do , where $j=1,2,...k$
3. For each dataset d_i, $d_i \in c_j$
   3.1 Calculate Average accuracy $acc_c = \sum_{d_i \in c} acc(d_i) / |d|$,where acc_c is the cluster accuracy , $acc(d_i)$: dataset accuracy from empirical evaluation , | d | is number of dataset in cluster c
   3.2 Calculate Average cost $cost_c = \sum_{d_i \in c} cost(d_i) / |d|$, where $cost_c$ is the cluster misclassification cost, $cost(d_i)$: dataset misclassification cost from empirical evaluation, |d| is number of dataset in cluster c_j
   3.3 Calculate the accuracy error for each dataset using the following equation

$$E_{acc_{d_i}} = \sqrt{(acc(d_i) - acc_c)^2}$$

   3.4 Calculate the cost error for each dataset using the following equation

$$E_{cost_{d_i}} = \sqrt{(cost(d_i) - cost_c)^2}$$

3.5 Calculate $z\_score_{d\,i}$ for each dataset d_i in a cluster using the following

$$z_{score_{d_i}} = \sum_{i=1}^{|d|} \sum_{n=1}^{N} \sqrt{\frac{(T_{ni}-M_{ni})^2}{std_{ni}}}, \quad \textit{where } T_{ni} \textit{ is dataset attribute value , N: number of data attributes}$$

M_{ni}: *cluster mean value for specific attribute n,*

std_{ni}: *cluster standard deviation for specific attribute n*

4. End loop // dataset loop
5. Divide each cluster into three zones,

$z\_score_1 = \{d_i \mid 0 \le z_{score_1}(d_i) \le 1\}$,

$z_{score_2} = \{d_i \mid 1 < z_{score_2}(d_i) \le 2.5\}$,

$z\_score_3 = \{d_i \mid z_{score_3}(d_i) > 2.5\}$.

6. For each z_score_i, $1 \le i \le 3$
   - 6.1 Calculate the average accuracy error of all data that are in a specific cluster z_score_i using the following equation

     $AvgEr_{acc\_z_{scorei}} = \sum_{d \in z\_score_i} E_{acc_{d_i}} / |\, z\_score_i \,|$, where $|\, z\_score_i \,|$ is number of dataset in a specific z_score.
   - 6.2 Calculate the average cost error of all data that are in a specific cluster z_score_i using the following equation

     $AvgEr_{cost\_z_{scorei}} = \sum_{d \in z\_score_i} E_{cost_{d_i}} / |\, z\_score_i \,|$
7. End loop // z_score loop

   // determine dataset charters where the selected query of yet unlabelled data X_u that are most informative for the next learning process
8. X_M = ***DetermineDataset*** $(clusters, AvgEr_{acc\_z_{scorei}}, AvgEr_{cost\_z_{scorei}})$
9. End loop // clusters loop
10. Return (X_M)

Figure 3-10: The query formulation steps

//Determine dataset from available unlabelled datasets algorithm//

DetermineDataset $(clusters, AvgEr_{acc\_z_{scorei}}, AvgEr_{cost\_z_{scorei}})$

Input:

- error rate for each cluster's z_score $AvgEr_{acc\_z_{scorei}}$
- Average error rate for each cluster's z_score *where* $i, 1 \leq i \leq 3$
- Set of generated clusters

Output:

- Set of informative datasets X_M, where $X_M \subseteq X_u$ and $X_M \neq \{\}$.

Procedure:

1. For each cluster $c \in$ ***clusters*** do the following
2. For i =1....2 do
   - 2.1 IF ($AvgEr_{acc\_z_{scorei+1}} > AvgEr_{acc\_z_{scorei}}$)
   - 2.2 Then goodCluster *// no need to do anything*
   - 2.3 ElSE *// search for a new dataset that goes into a specific cluster with specific z_score //*
     - 2.3.1 X_{m_i} = ***Search for new Data*** (c ,z_score_i)
     - 2.3.2 $X_{m_{i+1}}$ =***Search for new Data*** (c , z_score_{i+1})
     - 2.3.3 $X_M = X_{m_i} \cup X_{m_{i+1}}$
   - 2.4 End ElSE
   - 2.5 End IF
3. End loop
4. For j =1....3 do
   - 4.1 IF ($z\_score_j = \{\}$) *// no data inside z_score*
   - 4.2 X_{m_j} =***Search for a new Data*** (clusters, c, z_score_i) .
   - 4.3 $X_M = X_M \cup X_{m_j}$
5. End Loop
6. End loop
7. Return X_M;

Figure 3-11: Dataset determination steps

Search for new Data algorithm *(clusters, c,* z_score_i*)*

***Input*:**

- Needed cluster **c**, *needed* z_{score} $\boldsymbol{z\_score_i}$.
- Set of generated clusters

Output:

- Informative dataset $\boldsymbol{X_m}$

Procedure:

1. For each j=1 …to $\boldsymbol{k}$, $\boldsymbol{k}$ is the number of clusters
2. For each $i = 1 \ldots . \boldsymbol{U}, \boldsymbol{U}$ is the number of unlabelled example X_u
   - 2.1 Calculate the distance from X_{ui} to each cluster $\boldsymbol{c_j}$

     $\boldsymbol{DistFunction_{ij}} = \frac{\sqrt{\sum_{i=1}^{U}\sum_{j=1}^{k}(T_{ni}-M_{nj})^2}}{N}$ *where* T_n: The value of dataset attribute, where n=1.....*N, N*: number of dataset attributes, M_n: *cluster mean value for a specific attribute*
   - 2.2 Assign X_{ui} to its closest cluster C_i which gives value **min** $\boldsymbol{(DistFunction_{ii})}$
   - 2.3 Calculate z_score_{ui} of new unlabelled dataset using z_score equation
   - 2.4 If ($C_i = \boldsymbol{c}$) && ($z\_score_{ui} = \boldsymbol{z\_score_i}$)
   - 2.5 $X_m = X_m \cup X_{ui}$
3. End loop
4. End loop
5. Return X_m

Figure 3-12: Dataset search steps

A significant assumption in the above algorithm is that, as more data becomes available, the performance of the predictions improves; however, some initial trials suggest that the error rate does not always decrease as the number of labelling data grows for all datasets in

different clusters (more details can be found in Section 4.3). Figure 3-13 shows what can happen when the error rate starts to increase. In an effort to address this problem, rather than always clustering on the last state, the procedure is extended by storing the cluster information that gives the minimum error prediction rate and re-clusters the labelled dataset based on 'best cluster'. The results of these algorithms are shown in detail in Section 4.3

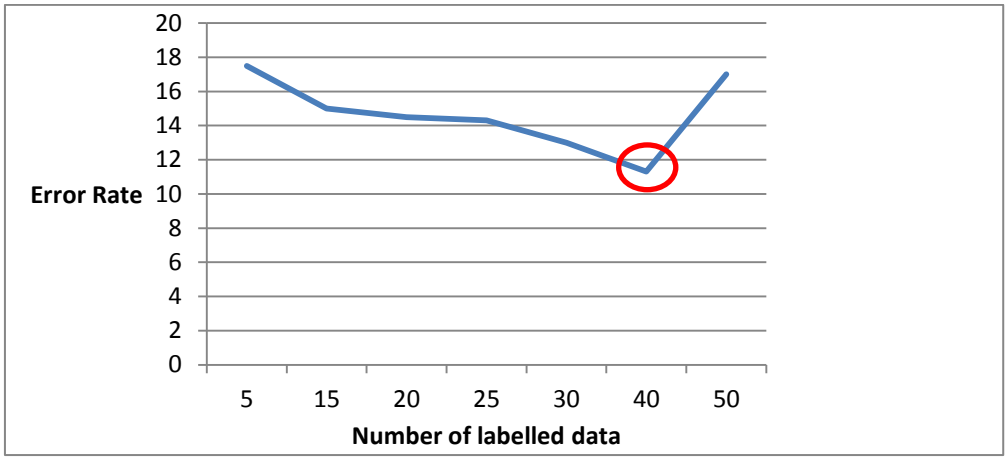

Figure 3-13: Error Rate versus number of labelled data using ALBC

Clustering on Best Cluster

For each learning method (J48, naïve Bayes, neural network, Part, oneR, and ZeroR):

- Find the cluster prediction error
- Find the cluster with highest performance
- Predict the data needed based on c_{min}using *Active Learning Based on Clustering Algorithm* in Figure 3-9
- Re-cluster based on new data.

Figure 3-14 illustrates clustering on the best cluster method. As shown in Figure 3-14, each cluster prediction performance is calculated using ALBC algorithm, then the cluster with highest performance is used for the prediction then the data is predicted using the specified cluster (highest performance) then the clustering is done based on this cluster.

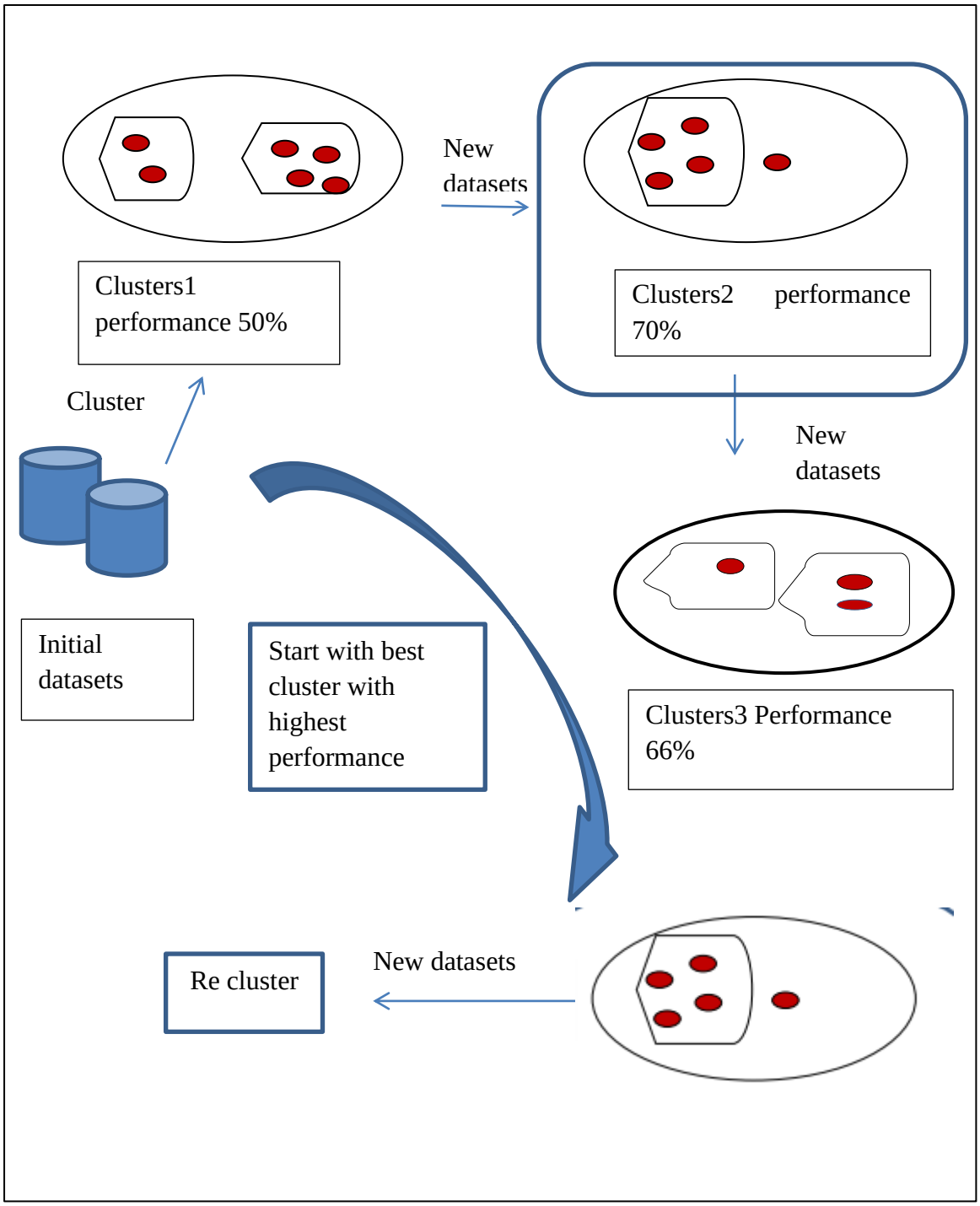

Figure 3-14: Best cluster method

3.3 Summary of Cost-Sensitive Meta-Learning Work and Active Learning

This chapter has presented the design of a meta-learning system for recommending feature selection methods and cost-sensitive learning methods. The central idea is to apply various feature selection methods and learning algorithms to determine their performance on several data sets. The performance and characteristics of the data are combined to form examples which are then used as input J48 to generate the meta-knowledge in the form of decision trees. This meta-knowledge can then be used to predict the performance of the feature selection methods and algorithms on new data, and hence allow the user to select the most suitable methods.

The chapter also explores how one can speed up the meta-learning process by identifying the next most suitable dataset(s) to digest into the meta-learning process. This is done by choosing a small set of labelled data, the labelled data are clustered according to its meta-features, and then any new data that will enter the learning process will be assigned to the clusters that are nearer to its meta-features. The quality of the clusters is determined by using z-scores and the characteristics of data sets that can improve the clusters is identified and provides the basis for selecting the new data sets to ingest.

The next chapter presents a detailed evaluation of the methods developed in this chapter, including a comparison with the METAL system for meta-learning.

A full detailed experiments description and comparison between those two solutions and other existing work will be done in chapter 4.

CHAPTER FOUR: EMPRICAL EVALUATION OF NEW COST SENSTIVE META-LEARNING SYSTEM

Chapter 3 presented the design of a system for meta-learning knowledge concerning cost-sensitive learners. It proposed the use of data characterisation coupled with performance data in the generation of examples, providing input to a learning algorithm that induced knowledge, which subsequently could be used to predict the performance of learning algorithms on given datasets. Chapter 3 also proposed a new approach to active learning in an effort to speed-up the meta-learning process.

This chapter presents an empirical evaluation of the following ideas:

- Section 4.1 evaluates the meta-learning process to learn the meta-knowledge for recommending feature-selecting methods.
- Section 4.2 evaluates the meta-learning process to learn the meta-knowledge for recommending cost-sensitive algorithms, and accordingly presents a comparison with those obtained with METAL—a project with similar goals.
- Section 4.3 presents an evaluation of the use of active learning by comparing with the use of a random process for the selection of meta-example.

Each section is structured so that the aim of the experiment and methodology adopted is explained, with the results presented and then discussed.

4.1 Feature Selection Experiment

As described in Section 3.1, feature selection is an important element of data-mining since there may be several features available in a dataset that are irrelevant or correlated, and it is known that the use of features that are correlated or irrelevant can have an adverse effect on some learning algorithms (Hall, 1999a; Kim, Street, & Menczer, 2003; Kohavi & John, 1997). Hence, Section 4.1.1 evaluates whether or not feature selection has an effect and whether there is a single best method. Subsequently Section 4.1.2 aims at building meta-

knowledge that recommends the most appropriate feature selection approaches for a given dataset.

4.1.1 Evaluating Feature Selection Approaches

This section presents the results of an evaluation in assessing how well feature-selecting methods work, and further assesses whether there is, in fact, a best method. The experiments use 10 datasets from the UCI Machine Learning Repository(Bache & Lichman, 2013), namely contact lenses, diabetes, vote, iris, credit-g, labour, glass, ionosphere, breast-cancer and soybean. For each dataset, the following characteristics are identified: number of classes, number of instances, number of attributes, class entropy, class skew and class conditional entropy. The experiments utilise four target learners: J48, naïve Bayes, neural networks, and a cost-sensitive classifier using minimum expected cost with J48 as a base learner. The 10 cross-validation methodology is adopted in the experiments, with the results including standard error.

In order to evaluate the effects of using feature-selection methods on classifier performance, a comparison is carried out between using and not using a feature-selection method. Three different feature-selection methods are considered, with the results presented below.

- Table 4-1 summarises the improvements in classifier accuracy after using a wrapper feature-selecting method, known as the WrapperSubsetEval in Weka (Kohavi & John, 1997), which uses a greedy search method for adding and removing attributes. The results are presented in three columns for each base learning method: a column with the accuracy prior to use of feature selection, a column after utilising the wrapper method with feature subset selection (columns with a FS suffix) and a column highlighting the improvement (columns labelled IM). Results are presented for a decision tree learner, known as J48, naïve Bayes (NB), neural networks (NN) and the use of cost-sensitive learning with J48 (J48_Cost). Figure 4-1 presents improvements in the form of a bar chart, showing improvements in almost all cases, except J48_cost-sensitive classifier in Diabetes and Vote datasets, and NN in both Glass and Weather datasets.

- Table 4-2 summarises the improvements in classifier performance using a filtering feature-selection method known as GainRatioEval in Weka. This method uses worth of an attribute by utilising an information theoretic measure, which is used in decision tree learning algorithms, known as the gain ratio, with respect to the class. Figure 4-2 presents the improvements in the form of a bar chart, highlighting the improvements in almost all cases, except J48 classifier in Ionosphere, Soybean and Vote datasets, naïve Bayes in Weather, Soybean and Vote datasets, NN shows a decrement or no changes in all cases, except Contact-lenses, Ionosphere, Vote and Cancer datasets, and J48_Cost-sensitive shows decrement in all cases except credit-g, Glass, Labour and Cancer.
- Table 4-3 and Figure 4-3 summarise the improvements in classifier performance using a feature-selection wrapper method, known as cfsSubEval in Weka, which evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature, in addition to the degree of redundancy between them (Hall, 1999a), J48 shows a decrement or no changes in all cases, except Diabetes, Glass and Labor datasets. Naïve Bayes shows accuracy improvements in all cases except Credit-g, Glass and Weather datasets. NN shows accuracy improvements in all datasets except Contact-lenses, Glass and Weather whereas; J48_Cost-sensitive shows increment or no changes in all cases except Contact-lenses, Diabetes and Cancer.

Table 4-1: Changes in classifiers performance accuracy after using WrapperSubSetEval with Greedy Search (accuracy % ∓ standard error)

Dataset	*J48*	*J48 FS*	*IM*	*NB*	*NB FS*	*IM*	*NN*	*NN FS*	*IM*	*J48_ Cost*	*J48_ Cost _FS*	*IM*
Contact Lenses	83.3 ∓22.8	87.3 ∓7.2	4	70.8 ∓7.5	87.5 ∓7.2	16.7	70.8 ∓6.8	83.3 ∓7	12.5	79.2 ∓6.1	83.3 ∓6.7	4.1
Credit-Card g	70.5 ∓1.18	74.6 ∓1.1	4.1	75.4 ∓1.6	74.4 ∓1.2	-1	71.6 ∓1.1	96 ∓0.9	24.4	72.6 ∓1.5	73.7 ∓2	1.1
Diabetes	73.8 ∓2.17	75 ∓1.8	1.2	76.3 ∓1.3	77.5 ∓1.4	1.2	75.3 ∓1.7	78.6 ∓1.3	3.3	75.1 ∓2.3	74.5 ∓3.1	-0.6
Glass	66.6 ∓3.6	71 ∓3.7	4.4	48.5 ∓4	58.1 ∓2.5	9.6	67.7 ∓3.5	67.2 ∓3.8	-0.5	70∓4	7∓4	1
Ionosphere	91.4 ∓2.0	93.7 ∓3	2.3	82.6 ∓3.5	92.0 ∓1.7	9.4	91.1 ∓2.2	92.5 ∓1.7	1.4	90.3 ∓2	90.5 ∓2.3	0.2
Labour	73.6 ∓4.4	82.4 ∓4.4	8.8	89.4 ∓2.6	94.7 ∓2.6	5.3	85.9 ∓3.4	94.7 ∓2.1	8.8	75 ∓2.4	82.4 ∓4	7
Weather	64.2 ∓14.1	71.4 ∓14.9	7.2	64.2 ∓12.3	71.4 ∓12.6	7.2	78.5 14.9	71.4 ∓14.2	-7.1	35.7 ∓5	35.7 ∓3	0
Soybeafn	91.5 ∓3.9	93.4 ∓3.6	1.4	92.9 ∓2.2	93.2 ∓2.3	0.3	93.4 ∓2.9	94.4 ∓2.8	1	91.5 ∓2.7	92.2 ∓2.7	0.7
Vote	96.3 ∓0.7	96.9 ∓0.7	0.7	90.1 ∓2.1	95.8 ∓1	5.7	94.1 ∓0.6	97.0 ∓0.74	2.9	96.3 ∓1.5	95.6 ∓2	-0.7
Cancer	75.5 ∓2.6	75.8 ∓2.7	0.3	71.6 ∓2.4	74.1 ∓2.4	2.5	64.6 ∓2.5	75.8 ∓2.5	11.2	73.4 ∓4	73.4 ∓3	0
Average of IM			34.4			56			57.9			12.8

Table 4-2: Changes in classifier performance accuracy after using GainRatioEval with Ranker (accuracy % ∓ standard error)

Dataset	J48	J48 FS	IM	NB	NB FS	IM	NN	NN FS	IM	J48_ Cost	J48_ Cost_	IM
Contact Lenses	83.3 ∓22.8	83.3 ±7.1	0	70.8 ∓7.5	83.3 ±7.2	12.4	70.8 ∓6.8	83.3 ±7.1	12.5	79.2 ±7	74.8 ±7	-4.4
Credit-Card g	70.5 ∓1.18	72.8 ±1.2	2.3	75.4 ∓1.6	75.5 ±1.5	0.1	71.6 ∓1.1	70 ±1.7	-1.6	72.6 ±2.1	73.1 ±2.4	0.5
Diabetes	73.8 ∓2.17	74.3 ±2	0.2	76.3 ∓1.3	76.4 ±1.1	0.1	75.3 ∓1.7	76.0 ±1.6	-0.7	75.1 ±2	72.6 ±1.8	-2.5
Glass	66.6 ∓3.6	68.3 ±4.7	1.6	48.5 ∓4	50 ±4.1	1.5	67.7 ∓3.5	67.2 ±2.3	-0.5	70 ±2.5	71±3	1
Ionosphere	91.4 ∓2.0	90.3 ±2.2	-1.1	82.6 ∓3.5	87.1 ±1.4	4.5	91.1 ∓2.2	91.4 ±2.1	0.3	90.3 ±3	90. ±4.1	-0.3
Labour	73.6 ∓4.4	77.1 ±5.6	3.5	89.4 ∓2.6	89.4 ±3.6	0	85.9 ∓3.4	85.9 ±3.6	0	75.4 ±4	78.9 ±4.2	3.5
Weather	64.2 ∓14.1	71.4 ±14.9	7.1	64.2 ∓12.3	57.2 ±11.8	-7	78.5 ∓14.9	78.5 ±14.2	0	35.7 ±3	35.7 ±3.2	0
Soybean	91.5 ∓3.9	85 ±8.4	-6.5	92.9 ∓2.2	85.9 ±4	-7	93.4 ∓2.9	86.2 ±8.4	-7.2	91.5 ±3	90.4 ±5	-1.1
Vote	96.3 ∓0.7	95±1	-1.3	90.1 ∓2.1	89 ±1.8	-1.1	94.1 ∓0.6	94.7 ±1.4	0.6	96.3± 4	95.4 ±5	-0.9
Cancer	75.5 ∓2.6	75.1 ±2.5	-0.03	71.6 ∓2.4	72.3 ±2.7	0.7	64.6 ∓2.5	69.5 ±2.8	4.9	73.4 ±3.4	74.8 ±4.1	1.4
Average of IM			5.8			4.2			8.3			-2.8

Table 4-3: Changes in classifier performance accuracy after using CfcSubEval with bestFirst (accuracy % ∓ standard error)

Dataset	J48	J48 FS	IM	NB	NB FS	IM	NN	NN FS	IM	J48_ Cost	J48_ Cost _FS	IM
Contact Lenses	83.3 ∓22.8	70.8 ∓6.5	-12.5	70.8 ∓7.5	70.8 ∓6.5	0	70.8 ∓6.8	66.6 ∓4.7	-4.2	79.2 ±5	50 ±4.3	-29.2
Credit-Card g	70.5 ∓1.18	70.5 ∓1.2	0	75.4 ∓1.6	74.4 ∓1.6	-1	71.6 ∓1.1	73 ∓3.6	1.4	72.6 ±2.4	72.7 ±3.5	0.1
Diabetes	73.8 ∓2.17	74.8∓ 2	1.1	76.3 ∓1.3	77.5 ∓1.5	1.2	75.3 ∓1.7	75.5 ∓5.2	0.2	75.1 ±1.5	73.5 ±5.3	-1.6
Glass	66.6 ∓3.6	68.9 ∓3.7	2.3	48.5 ∓4	47.6 ∓4.2	-0.9	67.7 ∓3.5	65.8 ∓3.3	-1.9	70 ±4	70 ±5.6	0
Ionosphere	91.4 ∓2.0	90. ∓2.35	-1.4	82.6 ∓3.5	92.0 ∓1.9	9.4	91.1 ∓2.2	93.4 ∓2	2.3	90.3 ±7	90.8 ±5.6	0.5
Labour	73.6 ∓4.4	77.1 ∓5.6	3.5	89.4 ∓2.6	91.2 ∓3.5	1.8	85.9 ∓3.4	85∓4	-0.9	75.4 ±4	80.7 ±6	5.3
Weather	64.2 ∓14.1	42.8 ∓14.9	-21.4	64.2 ∓12.3	57.1 ∓14.1	-7.1	78.5 ∓14.9	71.4 ∓14.2	-7.1	35.7 ±8	57 ±3	21.3
Soybean	91.5 ∓3.9	90.1 ∓4	-1.4	92.9 ∓2.2	92.2 ∓2.5	-0.7	93.4 ∓2.9	93.8 ∓3	0.4	91.5 ±8	90.7 ±5.4	-0.8
Vote	96.3 ∓0.7	96 ∓0.9	-0.3	90.1 ∓2.1	96 ∓0.9	5.9	94.1 ∓0.6	95.8 ∓0.8	1.7	96.3 ±3	95.4 ±2	-0.9
Cancer	75.5 ∓2.6	73.0 ∓2.7	-2.5	71.6 ∓2.4	72.3 ∓2.7	0.7	64.6 ∓2.5	72.6	8	73.4 ±4	57 ±5.6	-16.4
Average of IM			-32.6			9.3			-.01			-21.7

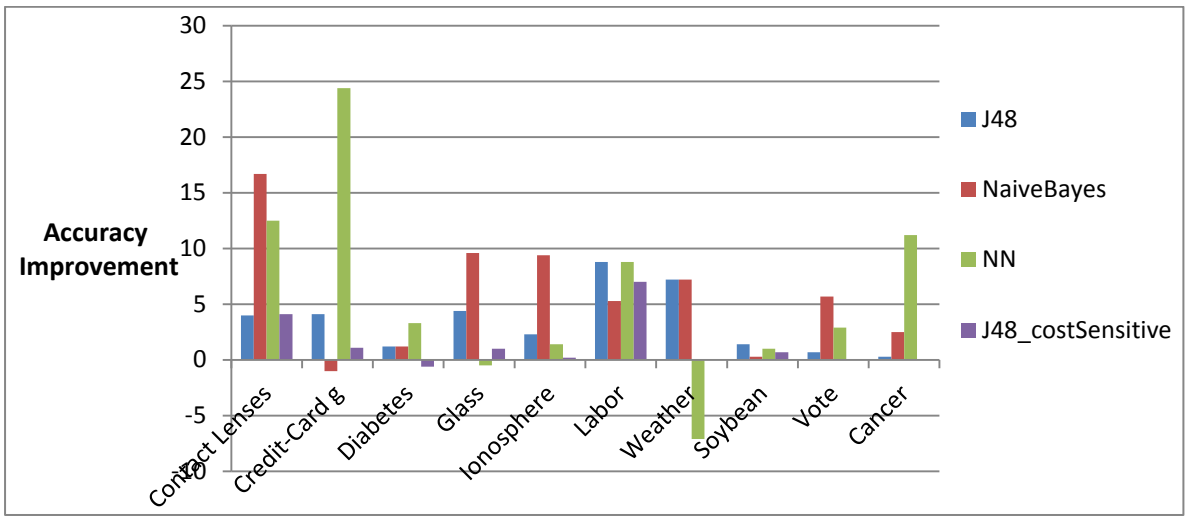

Figure 4-1: Changes in classifiers accuracy after using WrapperSubSetEval with Greedy Search

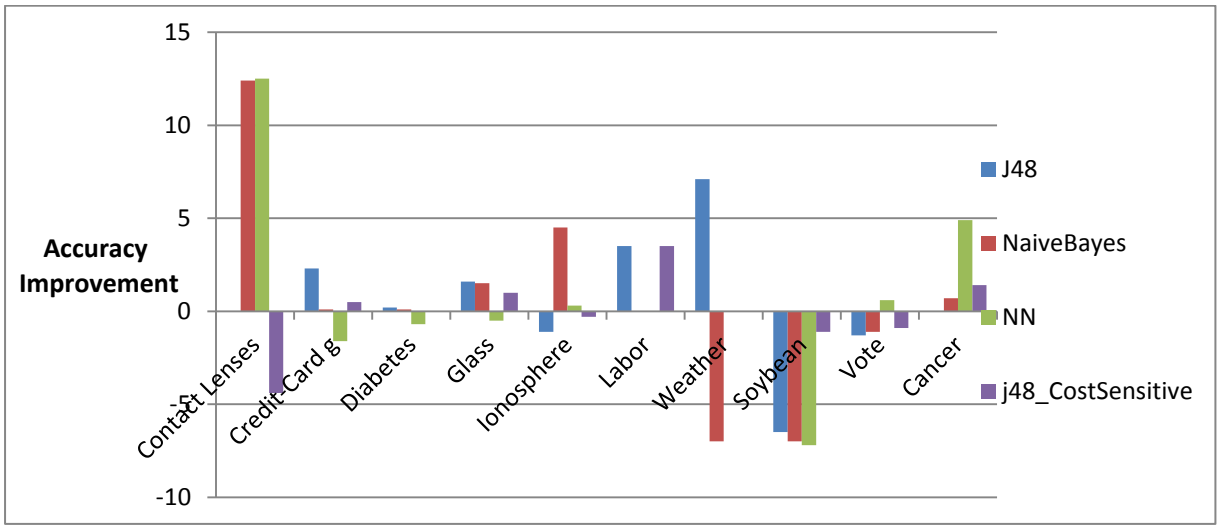

Figure 4-2: Changes in classifier accuracy after using GainRatioEval with Ranker

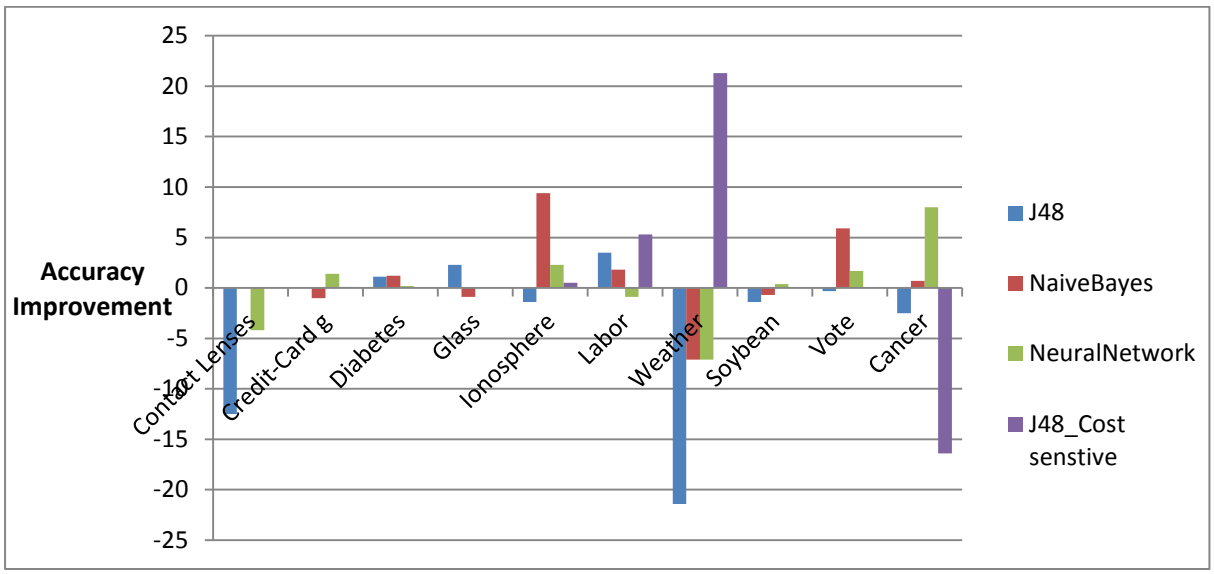

Figure 4-3: Changes in classifier accuracy after using CfcSubEval with bestFirst

The above results show that, in general, there is improvement in the learning algorithm performance when using feature selection positive values in IM columns), there are some negative values which shows accuracy decrements in some cases after using the feature selection which can result from losing some important features. Moreover, it also shows that there is no best feature-selection approach for all learning tasks: for example, in using the WrapperSubsetEval approach with greedy search, J48 performs better than the naïve Bayes classifier and neural network on Soybean dataset; however, naïve Bayes outperforms J48 and neural networks on the contact-lenses dataset once the WrapperSubSetEval feature-selecting method is used. J48 and naïve Bayes show the same performance improvement on Weather dataset, whereas neural networks, on the other hand, show a reduction in performance when this feature-selecting method is used. On the other hand, using cfsSubsetEval with a best-first strategy shows a decrement in performance in the same datasets (Weather, contact-lenses and Soybean).

What is required for a good data mining plan is good understanding of the data nature, and knowledge regarding which feature-selecting strategy works well for a specific dataset. Thus, the aim of our next experiment, as presented in Section 4.1.2, is finding the link between dataset characteristics, feature selection approaches and search techniques with different

classifier performance in an effort to establish which feature-selection method is best for a specific task.

4.1.2 Developing Meta-Knowledge for Feature Selection

Meta-knowledge for feature-selecting methods is developed by learning from the application of the methods on forty-two datasets from the UCI repository (Bache & Lichman, 2013). For this aim, different dataset characteristics are used for the purpose of understanding the nature of the data and to know what makes a feature-selecting method work well on a specific dataset. For each dataset, the following characteristics are identified: number of classes, number of instances, number of attributes, class entropy, class skew, and class conditional entropy. All of these characteristics are linked to different feature-selection approaches under different search techniques, along with the application of six classifiers (J48, naïve Bayes, oneR, part, zeroR, and neural networks). The results, together with the characteristics, are fed to J48 in an effort to create the meta-knowledge for predicting the performance of the methods, which then can be used for recommending what feature-selection approaches should be used on a specific dataset.

Table 4-4 summarises the feature-selection approaches, evaluation strategies and the search strategy used in the experiments. None indicates using a learning algorithm without feature-selection. There are two different feature-selection approaches used in the experiments: a wrapper method and filter method. In a wrapper method, a subset of features is evaluated using the learning algorithm that ultimately will be employed in the learning process. In a filter method, data is characterised independently from the learning algorithm, depending on data characteristics; therefore, the subset is filtered even prior to learning. Different search strategies are used, including forward greedy search, which involved starting with no features, adding them one by one, and evaluating performance. The search stops when adding features results in a drop in performance. Another search strategy used is best first, which contrasts with greedy search, where the searching process in best first does not terminate when the performance starts to drop; instead, a list of features that are chosen thus far are sorted in order of performance, whilst another search starts from different points, with a comparison carried out between different lists of feature stored previously (Hall, 1999a). Ranker search strategy is

used to rank attributes according to the measurements value used in filtering, such as entropy, gain ratio.

Table 4-4:Combination of feature selection approaches, search strategies, and evaluators

Feature Selection Approach	Attribute Evaluator	Search Strategy
None	None	None
Wrapper	ClassifierSubSetEval	GreedySearch
Wrapper	ClassifierSubSetEval	BestFirst
Wrapper	CfsSubSetEval	BestFirst
Filter	InfoGain	Ranker
Filter	GainRatioEval	Ranker`
Filter	OneAttributeEval	Ranker

In order to obtain the meta-knowledge for different feature-selecting methods, desired features are extracted for forty-two datasets. Each dataset, with its characteristics, is linked to 6 classifiers in an effort to evaluate their accuracy and cost after applying the combinations listed in Table 4-4; in other words, each dataset has 42 rows (6 classifier * 7 feature selection combination).

A sample of the data characteristics with different feature-selection approaches, along with the different classifiers accuracy and cost for contact-lenses dataset, is shown in Figure 4-4, where:

t1: number of classes, t2: number of attribute, t3: class entropy, t4: class skew, t5: number of instances, t6: class conditional entropy, f1: feature selection type, f2: evaluator, f3: search strategy

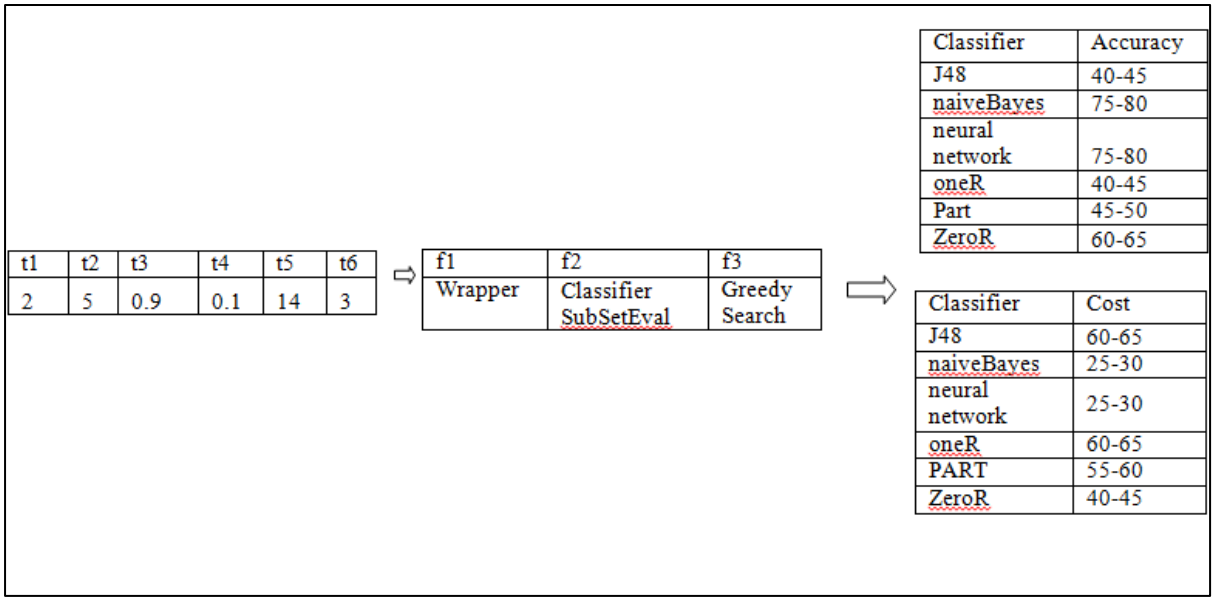

Figure 4-4: Sample of dataset characterises used to build meta-knowledge along with classifier performance for weather dataset

The following decision tree results, stemming from applying J48 as a meta-learner on the performance data and characteristics, are illustrated in Figure 4-5 and Figure 4-6 .

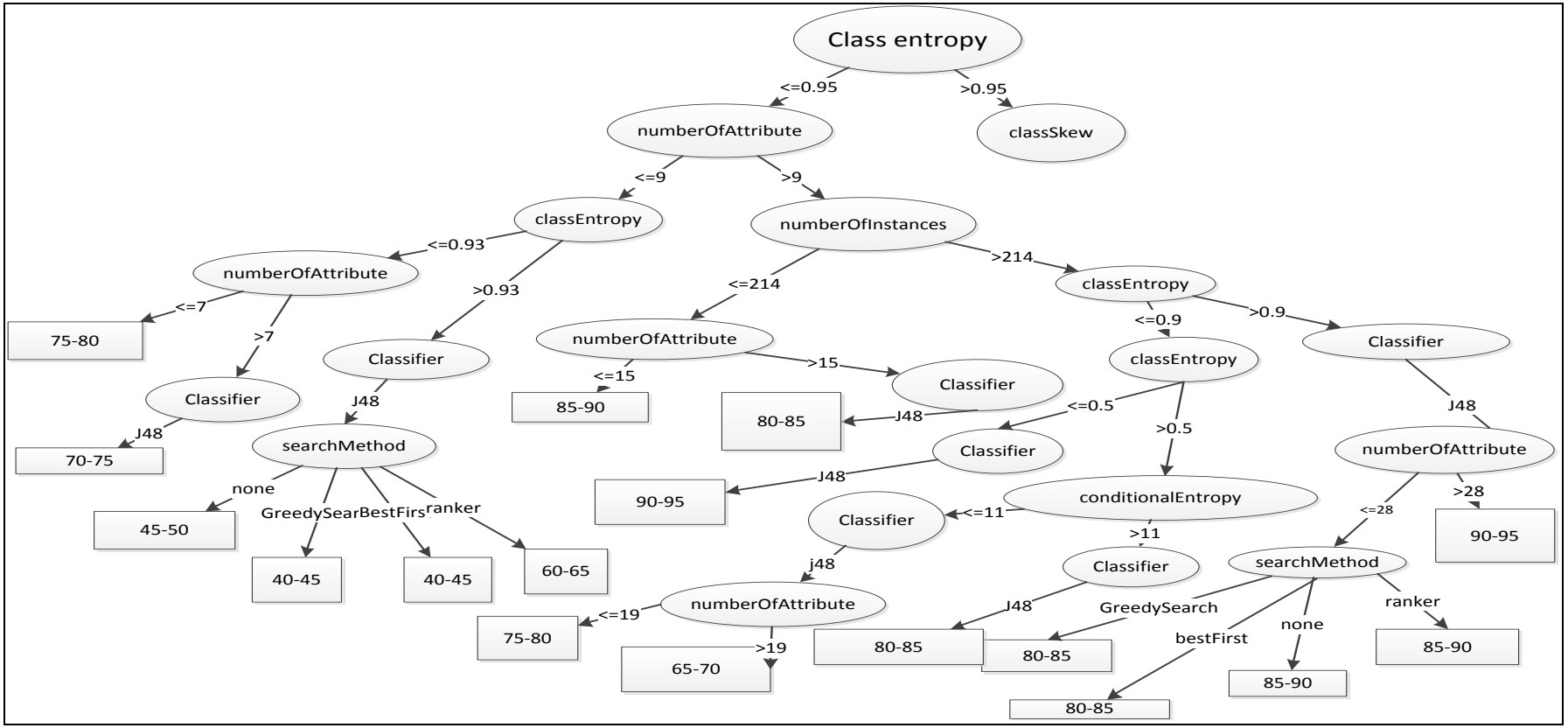

Figure 4-5 : Decision tree for feature selection with accuracy (Left)

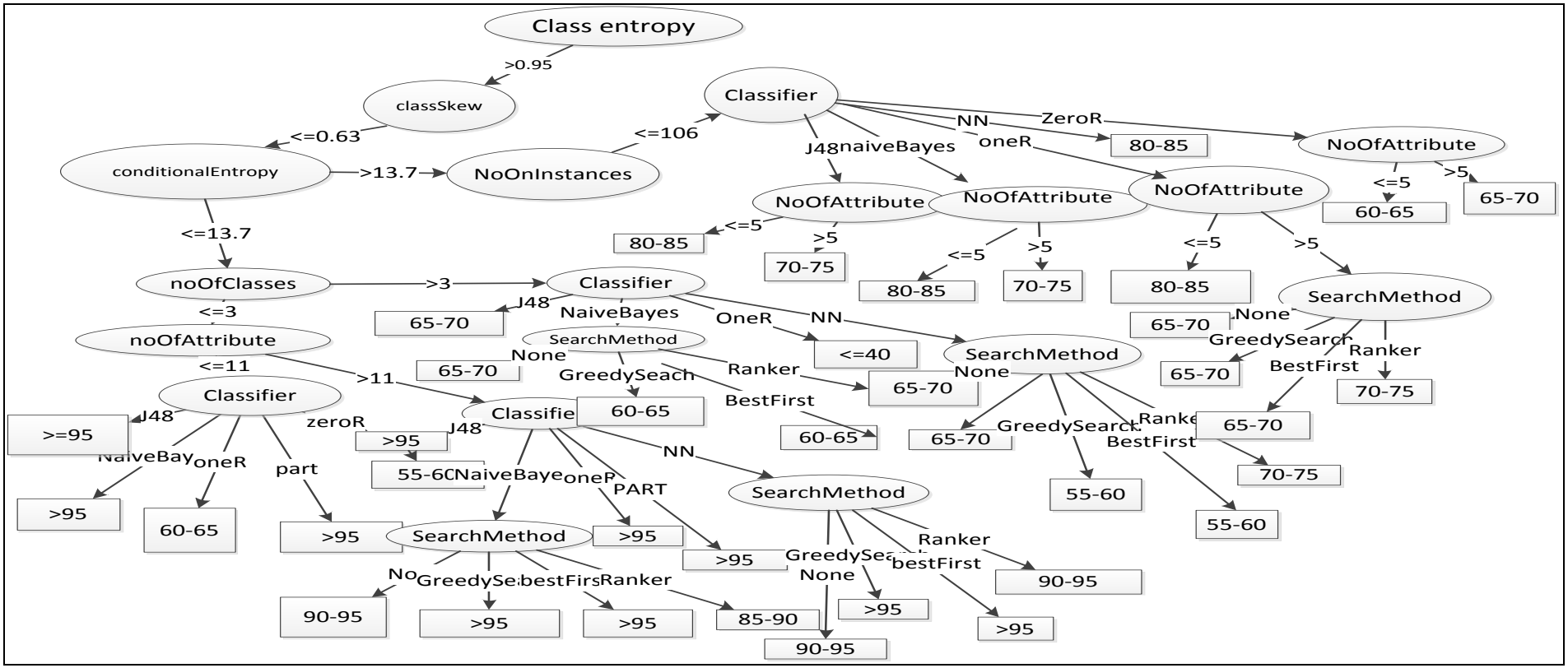

Figure 4-6: Decision tree for feature selection with accuracy (right)

The knowledge in the above decision tree is suitable as meta-knowledge, which can be used in deciding the feature-selecting methods best for a given dataset; that is, given a new dataset, its characteristics would be calculated and the tree traversed to predict the performance of specific algorithms. For example, if the class entropy for a given dataset is >0.95, class skew <=0.63, conditional entropy is <=13.7 and number of classes >3, then the predicted accuracy is 65–70 using J48 classifier, 65–70 using naïve Bayes with Ranker filter method and 60–65 using greedy search and best first method, oneR results in less than or equal to 40, NN will give 65–70 (without any feature selection), 55–60 with greedy wrapper search and Best first, and 70–75 with ranker filter methods. As these recommendations are given to a user, he/she will be able to choose the algorithm and feature-selecting method best suited to his case: for example, in this specific case, the user may choose to apply NN with ranker feature methods, as this gives the highest accuracy.

Notice that the classifier always appears as a splitting node in the decision tree, which suggests that the behaviour of the dataset (the accuracy and preferred feature selection) is highly affected by the learning algorithm used. The above decision tree is generated using all training data and the performance of applying 10 fold cross validation gives 78 %. The following observations can be made about these results:

- Feature-selecting methods are an important factor in the meta-knowledge decision tree, especially when the number of attributes is large: for example, if the number of attribute >11 (Figure 4-6), classifiers, such as naïve Bayes and NN, are highly affected by the feature selection search method used. In contrast, if there are less than 11 attributes and fewer than 3 classes, feature selection does not appear in the decision tree.
- Class skew presents a challenge to any induction algorithms. As mentioned previously in Section 2.1, when data is highly skewed, a classifier tends to classify each instance into dominant class to maximise accuracy. From the above decision tree, it can be noticed that feature selection is relatively more important in skewed data, as mentioned in (Forman, 2003). This appears in the above decision tree, where feature selection is not a factor in the case of class skew < 0.63, unless the number of attributes is relatively large (>11), whereas the classifier accuracy is affected by the

feature-selecting method if the class skew >0.63 for most of the learning algorithms used (J48, part, naïve Bayes, and NN). (See Appendix A1-2)

- Selecting the right set of features for classification is one of the most important problems in building a good classifier. Classifier performance is highly affected by irrelevant or correlated data; therefore, for example, J48 is less affected by using feature selection than naïve Bayes and neural networks because J48 is designed to choose the most relevant feature during construction. This appears in the previous decision tree, where the neural network and naïve Bayes accuracy are more affected when using feature selection, when a number of attributes is relatively large (>11) (Figure 4-6).
- Class Entropy measures the 'amount of mix' of the data: for example, if the class entropy=0, all instances belong to the same class (minimum mix) with high accuracy. We notice that, in the first tree (Figure 4-5), the accuracy of using J48 on a specific dataset with class entropy <0.93 is 70–80, whereas using J48 with feature selection on another dataset that has class entropy >0.93, accuracy ranges from 40–50 (using Greedy search method) and 60–65 using Ranker filter method. Hence, we can conclude that feature selection is not essential when using J48 if the classes are well discriminative (the class entropy is low).

The following decision tree results from applying J48 as a meta-learner on misclassification cost prediction, as illustrated in Figure 4-7 and Figure 4-8

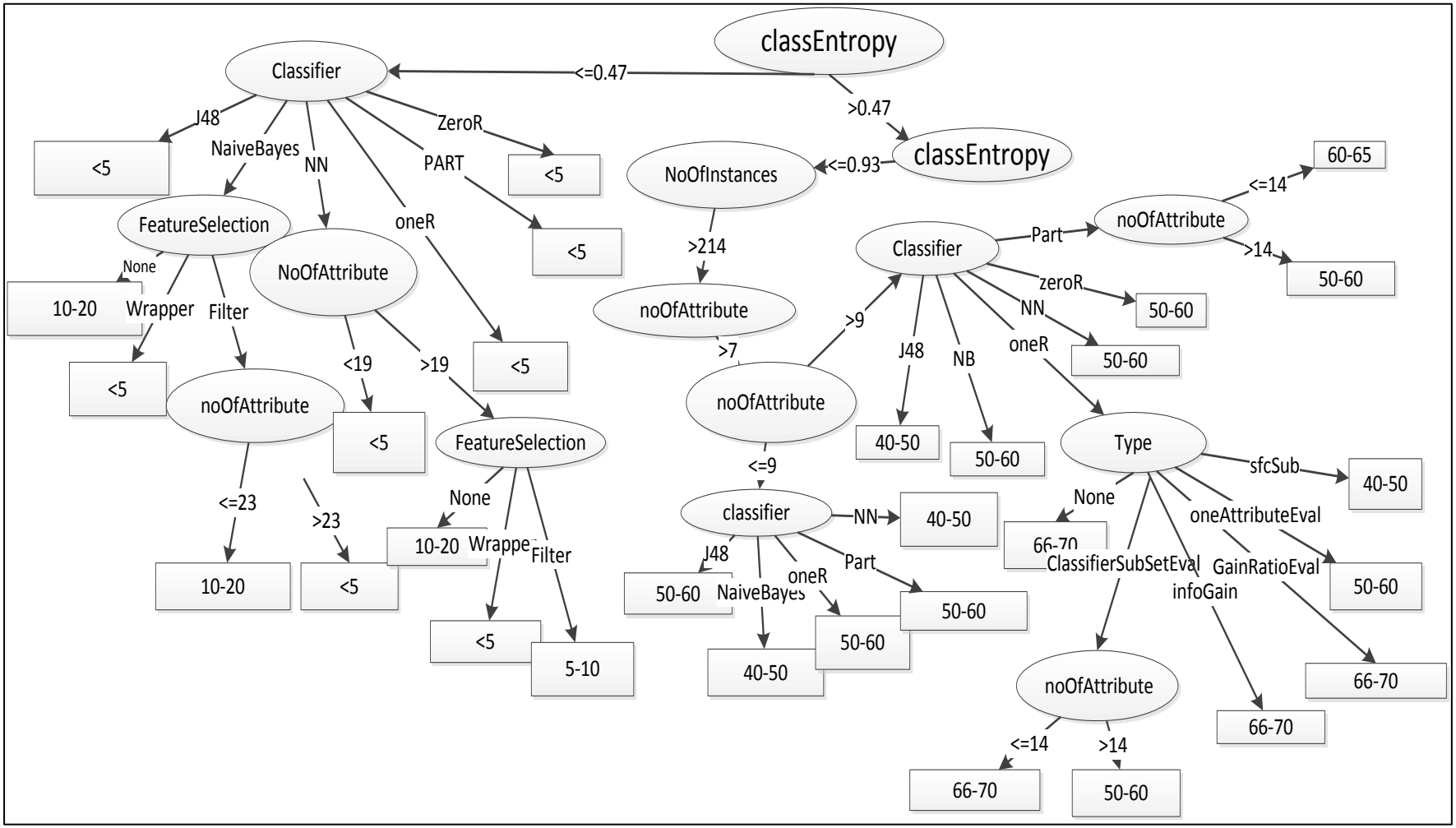

Figure 4-7: Decision Tree for feature selection with misclassification cost (left)

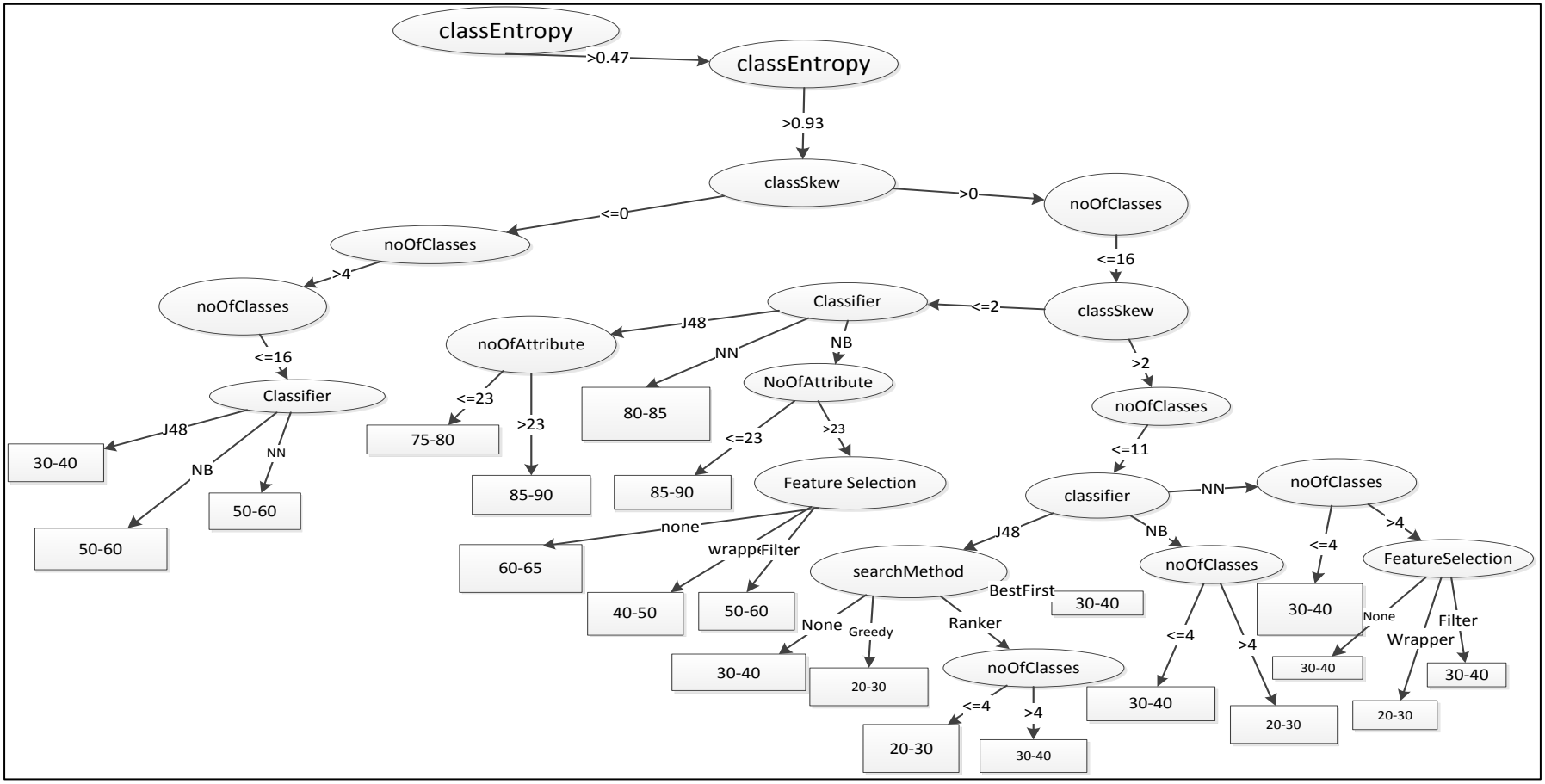

Figure 4-8: Decision Tree for feature selection with misclassification cost (right)

The knowledge in the above decision tree is suitable as meta-knowledge that can be used in deciding which feature-selecting methods are best for a given dataset in considering a misclassification costs; that is, given new data, its characteristics can be calculated and the tree traversed to predict the misclassification costs of a specific algorithm: for example, if the data class entropy >0.47 and number of instances >214, the number of attributes >9, then the user is recommended to use J48 or oneR algorithm with cfsSubsetEval feature selection, which gives a minimum misclassification cost. (See Figure 4-7).

The above decision tree is generated using all training data, and the following observations can be made:

- Class Entropy is an important characteristic affecting classifier performance and, accordingly, misclassification cost. In general, the misclassification costs for the set of examples with class entropy <0.5 are less than the misclassification costs for the datasets with a class entropy > 0.5 (see Table 4-5).

Table 4-5: Misclassification cost comparison between left and right branch of feature selection decision tree

<table>
<tr><td colspan="4">classEntropy<=0.5</td><td colspan="6">classEntropy >0.5</td></tr>
<tr><td colspan="4"></td><td></td><td colspan="3">Instances <=214</td><td colspan="2">instance>214</td></tr>
<tr><td>J48</td><td colspan="3"><5</td><td>J48</td><td>attribute <18</td><td colspan="2">attribute >18</td><td colspan="2">attribute</td></tr>
<tr><td rowspan="3"></td><td rowspan="3"></td><td rowspan="3"></td><td rowspan="3"></td><td rowspan="3"></td><td rowspan="3">30-40</td><td>none</td><td>50-60</td><td><=9</td><td>>9</td></tr>
<tr><td>wrapper</td><td>40-50</td><td rowspan="2">50-60</td><td rowspan="2">40-50</td></tr>
<tr><td>filter</td><td>40-50</td></tr>
<tr><td>NB</td><td>none</td><td colspan="2">10-20</td><td>NB</td><td colspan="3" rowspan="3">20-30</td><td>attribute <=9</td><td>attribute >9</td></tr>
<tr><td>wrapper</td><td colspan="2"><5</td><td rowspan="3"></td><td rowspan="3">40-50</td><td rowspan="3">50-60</td></tr>
<tr><td rowspan="2">filter</td><td>attribute <=23</td><td>10-20</td></tr>
<tr><td>attribute >23</td><td><5</td></tr>
<tr><td>NN</td><td colspan="2">attribute <=19</td><td>attribute >19</td><td>NN</td><td colspan="2">attribute <=18</td><td>attribute >18</td><td>attribute <=9</td><td>attribute >9</td></tr>
<tr><td colspan="2" rowspan="3"><5</td><td>none</td><td>10-20</td><td rowspan="3"></td><td colspan="2" rowspan="2">20-30</td><td rowspan="2">30-40</td><td rowspan="2">40-50</td><td rowspan="2">50-60</td></tr>
<tr><td>wrapper</td><td>< 5</td></tr>
<tr><td>filter</td><td>5-10</td></tr>
</table>

Table 4-5 shows the misclassification costs for J48, NB and NN applied on data with class entropy <=0.5 ranges from 5 to 20. On the other hand, the misclassification costs of applying the same classifiers on data with class entropy >0.5 range 40–60 (if the number of instances >214) and 30–60 (if the number of instances < 214) because datasets with lower class entropy are more easily discriminated, having higher accuracy and therefore a lower misclassification cost.

- From the above decision trees Figure 4-7 and Figure 4-8, we notice that feature selection is more critical when the number of attributes is relatively large. Table 4-5 shows that the misclassification costs are highly affected by feature-selecting methods if the number of attributes is relatively high: for example, when using feature selection with J48, the misclassification cost decreases from 50 to 40 if the number of attribute >18, and using feature selection with NN causes a decrement in the misclassification costs from 10 to 5 (if the number of attribute >19).
- Notice that the classifier always appears as a decision node, which states that the behaviour of the data (the cost of data misclassification) is highly affected by the learning algorithm used to build the model.
- Class skew is a critical characteristic affecting misclassification cost because class skew has a direct affect in classifier accuracy; class skew appears as a splitting point in Figure 4-8.

The next section presents the development of the meta-knowledge for cost-sensitive learning.

4.2 Cost-Sensitive Learning Experiment

This section presents an empirical evaluation of the meta-learning approach for recommending cost-sensitive methods. Section 4.2.1 describes the experiments performed to uncover the knowledge regarding the performance of various methods, as based on the characteristics of the data, i.e., the meta-knowledge that is able to guide future recommendations. Section 4.2.2 evaluates the results obtained by making a comparison with the METAL project.

4.2.1 Cost-Sensitive Experiment Methodology

This subsection describes the experiments that are conducted to learn a model that predicts the performance of cost-sensitive classifiers, given the characteristics of a dataset. Twenty-six datasets are characterised using simple and theoretical measurements, and are mapped to different cost-sensitive and -insensitive classifiers, evaluated using 10 folds cross-validation with the aim of finding under which circumstances a specific algorithm results in good accuracy and minimum cost. For cost-insensitive algorithms, the base learner is learnt, as it is without any wrapper methods, whilst for cost-sensitive algorithms, different wrapper phases are added to convert cost-insensitive classifiers to cost-sensitive. Thus, in each dataset, there are 264 rows (6 cost-insensitive classifiers *4 wrapper methods types and 11 cost ratios). As an example, Table 4-6 shows the characteristics used for the contact-lenses dataset, and Table 4-7 shows the cost ratios used, where the cost ratio ranges from 1 to 4 for each class. Table 4-8 summarises the different cost-sensitive and -insensitive approaches, Table 4-9 and Table 4-10 show how accuracy and cost are categorized.

Table 4-6: Dataset characters for contact-lenses dataset

NoOfClass	***NoOfAttribute***	***Class Entropy***	***Class Skew***	***NoOfInstances***	***Conditional Entropy***	***Example Ratio***	***Cost Ratio***
3	5	1.32	0	24	1.24	0.62	2

Table 4-7: Cost ratios

CostRatio	1	2	3	4	0.5	0.25	0.67	0.33	0.75	1.3	1.5

Table 4-8: Cost-sensitive approaches

Cost Insensitive classifiers	Cost-sensitive approaches	Cost-sensitive classifier
J48	None, Sampling,MinimumExpected cost, Bagging(MetaCost)	Meta-J48
OneR		Meta-OneR
ZeroR		Meta- ZeroR
NeuralNetwork		Meta-NeuralNetwork
PART		Meta- PART
NaïveBayes		Meta-naïve Bayes

Subsequently, the J48 classifier is used as a meta-learner to predict the accuracy and cost of different datasets, as characterised by different meta-features (shown in Table 4-6) and mapped to different cost-sensitive and -insensitive approaches (shown in Table 4-8).

The accuracy of the classifiers is categorised into the following bands (Table 4-9) in order to enable the application of learning algorithms, such as J48 for meta-learning:

Table 4-9: Accuracy categories

Accuracy	<=40	40-45	45-50	50-55	55-60	60-65	65-70	70-75	75-80	80-85	85-90	90-95	>=95

The misclassification cost for one instance is categorised into the following bands (Table 4-10) in order to enable the application of learning algorithms, such as J48 for meta-learning:

Table 4-10: Cost Categories

Cost	0-10	10-20	20-30	30-40	40-50	50-60	60-65	65-70	70-75	75-80	80-85	85-90	90-95	95-100	100-150	150-200	>=200

The characteristics, cost ratios and performance obtained can be combined to form examples. Table 4-11and Table 4-12 provide examples of the kind of data that can be obtained.

Table 4-11: Accuracy of cost-sensitive approaches with different dataset characteristics for contact-lenses dataset at a cost ratio of 0.5

t1	t2	t3	t4	t5	t6	t7	t8	Classifier	Wrapper Type	Accuracy
3	5	1.3	0	24	1.2	0.62	0.5	J48	None	80-85
									Bagging	80-85
									Sampling	85-90
									Min_ Expected_ Cost	80-85

Table 4-12: Cost of cost-sensitive approaches with different dataset characters for contact-lenses dataset at cost ratio 0.5

t1	t2	t3	t4	t5	t6	t7	t8	classifier	WrapperType	Cost
3	5	1.3	0	24	1.2	0.62	0.5	J48	None	20-30
									Bagging	20-30
									Sampling	10-20
									Min_expected_Cost	20-30

Figure 4-9 and Figure 4-10 show the meta-knowledge obtained when J48 is applied to the table of examples. Meta-knowledge like this can be used to make recommendations. Accordingly, if we take the contact-lenses dataset as an example, this has 3 classes, with 5 attributes, example ratio of 0.6, and if we consider the cost of rare classes is four times the cost of dominant classes (cost ratio=4), the system will recommend using J48 or part classifier without any wrapper method if the accuracy is the main user concern, and using neural network with sampling or minimum expected cost, which gives minimum cost.

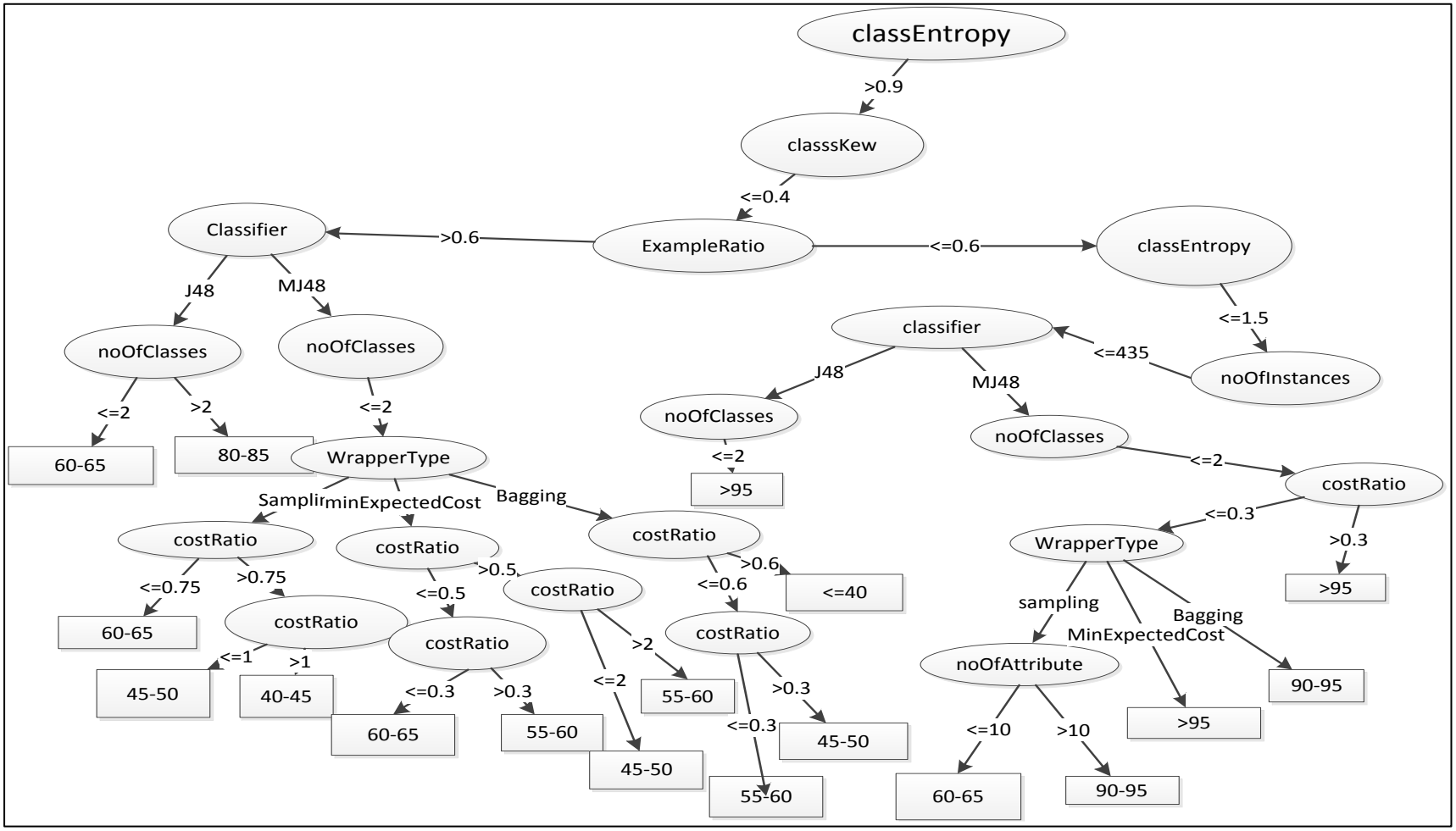

Figure 4-9: Decision tree for cost-sensitive methods accuracy (right)

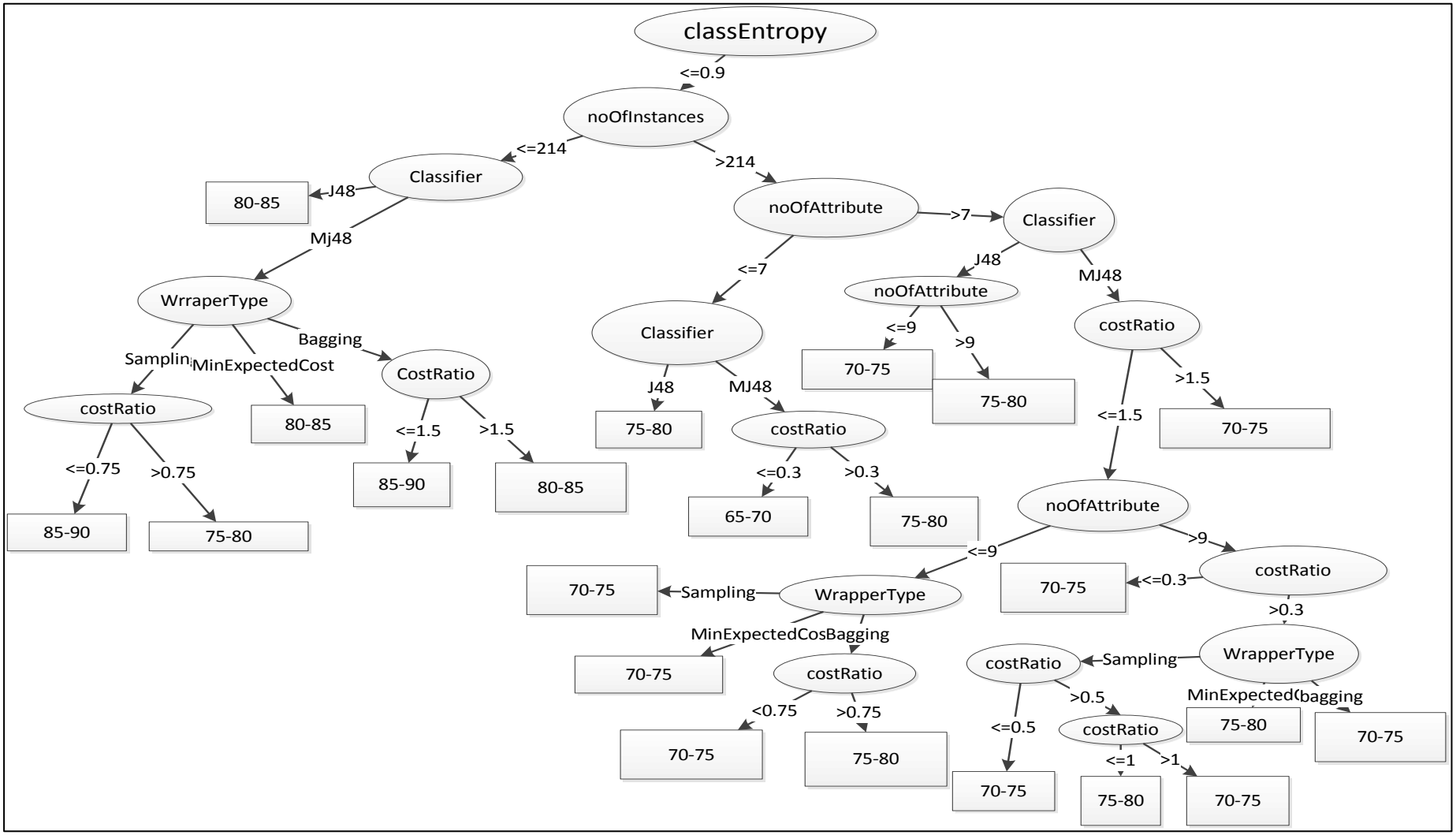

Figure 4-10: Decision tree for cost-sensitive methods accuracy (left)

The knowledge in the above decision tree is suitable as meta-knowledge that can be used in deciding which cost-sensitive methods are best for a given dataset. The following can be observed from the decision tree of Figure 4-9 and Figure 4-10:

- Class entropy is the most important factor used to predict classifier accuracy. As mentioned previously, class entropy measures the homogeneity of the data, meaning data with lower class entropy is expected to have higher accuracy because it is well discriminated. This appears in the previous decision Figure 4-9 and Figure 4-10.
- Classifier type is an important factor used to predict the accuracy of adopting a specific-cost-sensitive and -insensitive classifier on a specific dataset: for example, if we compare between using NN and ZeroR with wrapper methods in the same dataset, which have class entropy >0.9, example ratio <=0.6, and number of attribute>14, we notice that the accuracy of using ZeroR with sampling and bagging range from 40 to 65, while using NN give 90 to >95 for sampling and 90-95 for bagging (see Table 4-13 and the decision tree in Appendix A2-5).

Table 4-13: Different dataset accuracy using different cost-sensitive classifiers

<table>
<tr><th colspan="5">classEntropy>0.9 & classSkew<=0.4,exampleRatio<=0.6,noOfAttribute>14</th></tr>
<tr><td rowspan="2">NzeroR</td><td colspan="2">Sampling</td><td colspan="2">Bagging</td></tr>
<tr><td>CR<=1.5</td><td>CR>1.5</td><td>CR<-1.5</td><td>CR>1.5</td></tr>
<tr><td></td><td>60-65</td><td><=40</td><td>60-65</td><td><=40</td></tr>
<tr><td rowspan="3">NN</td><td colspan="2">Sampling</td><td colspan="2" rowspan="2">Bagging</td></tr>
<tr><td>CR<=2</td><td>CR>2</td></tr>
<tr><td>90-95</td><td>>95</td><td colspan="2">90-95</td></tr>
</table>

- Example ratio is the ratio of the number of instances belonging to the frequent class to the number of instances belonging to the rare class; the ratio measures the balance of the dataset. As mentioned previously in Section 2.1, classifiers applied to imbalanced data tend to classify each instance to the frequent class. The meta-knowledge in Figure 4-9 shows that the example ratio is an important factor. Now, comparing

balanced datasets (example ratio <=0.6) and imbalance datasets (example ratio >0.6) (both have number of classes =2), overall, the accuracy of the balanced datasets is more than the accuracy of the imbalanced dataset using the same classifiers with the same wrapper cost-sensitive methods. Table 4-14 shows that the accuracy of applying J48 and J48 with different wrapper methods on imbalanced data results in accuracy ranges spanning 40–65. On the other hand, applying the same classifiers on more balanced datasets gives a higher accuracy, ranging from 60 to 95. (See Figure 4-9).

Table 4-14: Comparison between J48 and MJ48 classifier accuracy between balance and imbalanced dataset

<table>
<tr><td colspan="9">ClassEntropy>0.9 && ClassSkew<=0.4,noOfClasses<=2</td></tr>
<tr><td colspan="5">ExampleRatio >0.6</td><td colspan="4">ExampleRatio <=0.6</td></tr>
<tr><td>J48</td><td colspan="4">60-65</td><td>J48</td><td colspan="3">>95</td></tr>
<tr><td rowspan="6">MJ48</td><td rowspan="2">Sampling</td><td>CR <=0.75</td><td colspan="2">1=>CR>0.75</td><td>CR>1</td><td rowspan="6">MJ48</td><td rowspan="2">Sampling</td><td>Attribute <=10</td><td>Attribute >10</td></tr>
<tr><td>60-65</td><td colspan="2">45-50</td><td>40-45</td><td>60-65</td><td>90-95</td></tr>
<tr><td rowspan="2">MinExCost</td><td>CR <=0.3</td><td>0.3<CR <=0.5</td><td>0.5<CR<=2</td><td>CR>2</td><td rowspan="2">MinExCost</td><td colspan="2" rowspan="2">>95</td></tr>
<tr><td>60-65</td><td>55-60</td><td>45-50</td><td>55-60</td></tr>
<tr><td rowspan="2">Bagging</td><td>CR<=0.3</td><td colspan="2">0.3<CR<=0.6</td><td>CR>0.6</td><td rowspan="2">Bagging</td><td colspan="2" rowspan="2">90-95</td></tr>
<tr><td>55-60</td><td colspan="2">45-50</td><td><40</td></tr>
</table>

- As one would expect, the cost ratio appears to be a significant decision node in the tree because the classifier performance is highly affected by the cost ratio characterisation. Table 4-14 shows that cost ratio does not appear in the decision node where the example ratio <= 0.6 as much as in datasets with example ratio > 0.6 because when data is imbalanced in the second case, the cost ratio becomes a critical factor that affects classifier accuracy (see Section 2.3 for imbalanced data problem). In addition, Table 4-14 shows the importance of the cost ratio in building a cost

sensitive classifier: for example, when using the MJ48 classifier with different wrapper methods (Sampling, MinExpectedCost and Bagging) in the imbalanced dataset, the accuracy decreases when applying a higher cost ratio (higher penalties on the misclassification of rare classes). Figure 4-11 shows the decrement in accuracy using different cost ratios over J48, naïve bayes and neural network wrapper methods on the contact lenses dataset as an example.

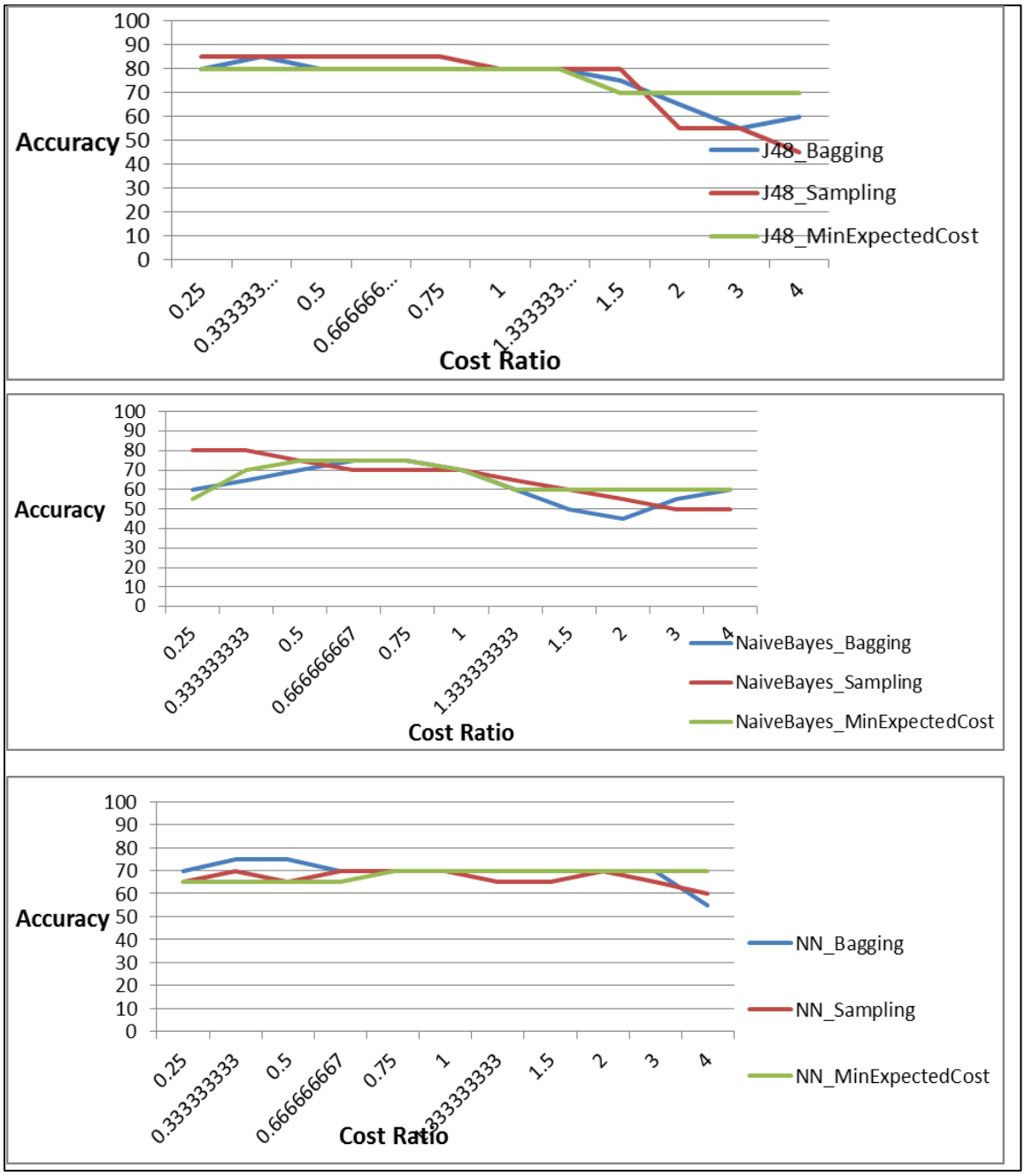

Figure 4-11: Cost ratio with classifier accuracy

Figure 4-12 and Figure 4-13 show the decision tree generated when applying J48 as meta-learner on the set of examples measuring the cost of misclassification.

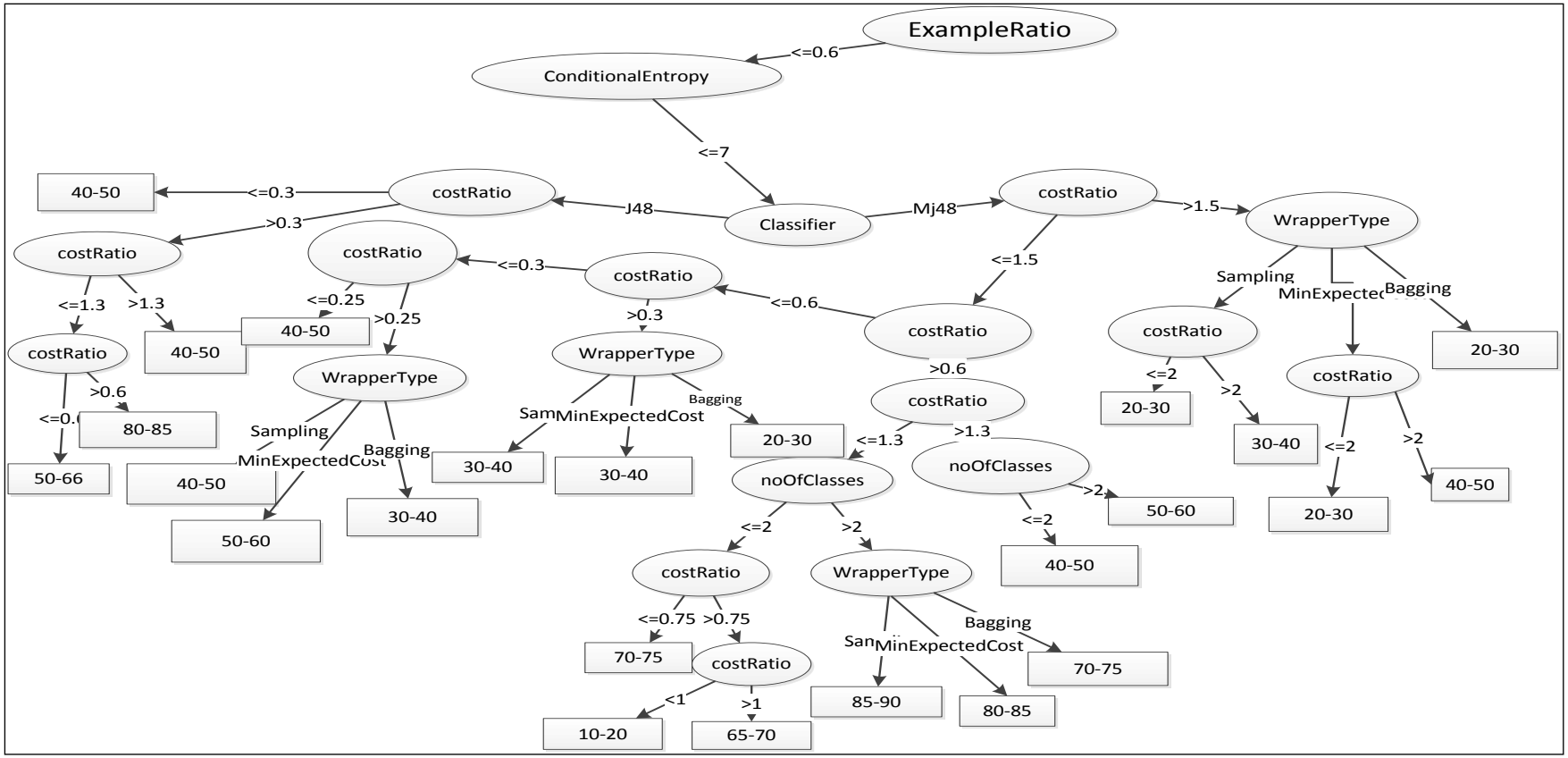

Figure 4-12: Decision tree for cost-sensitive methods in predicting cost (left)

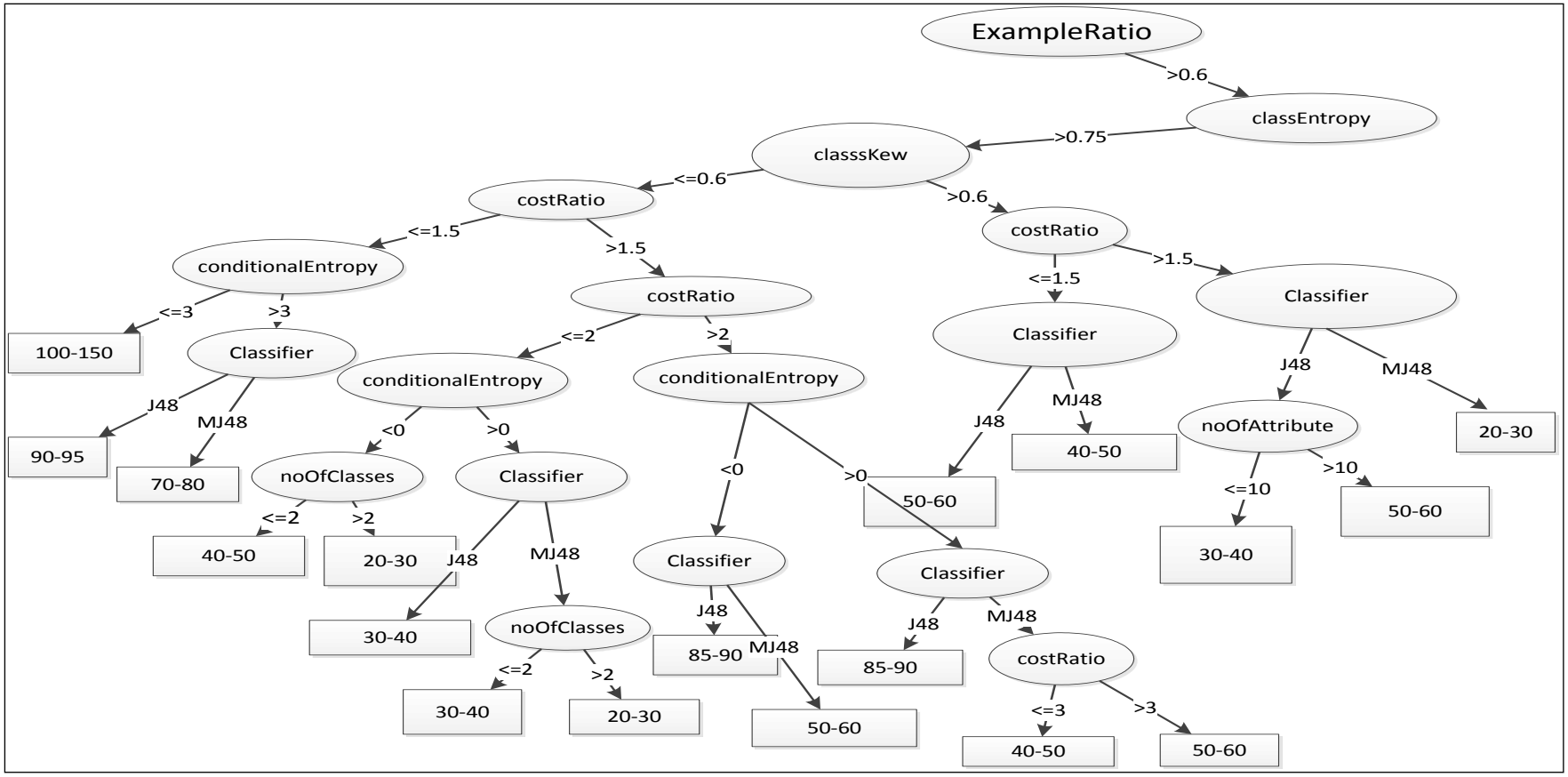

Figure 4-13: Decision tree for cost-sensitive methods in predicting cost (right)

The following can be observed from the decision tree in Figure 4-12 and Figure 4-13:

1. The ratio of examples and cost ratio are important characteristics affecting both classifier accuracy and, accordingly, its misclassification cost.
2. Class skew is an important factor affecting classifier misclassification cost, as shown in the previous decision tree (Figure 4-13), which shows more highly skewed data (class Skew >0.6)
3. We notice that changing cost ratio will have an impact on misclassification cost more so than on accuracy. Figure 4-14 presents the misclassification costs of using different wrapper methods (Bagging, Sampling, MinExpectedCost) with different classifiers (J48, naïve bayes, neural network) in contact-lenses dataset, and also shows the considerable cost changes (decrement or increment) for different cost ratios.

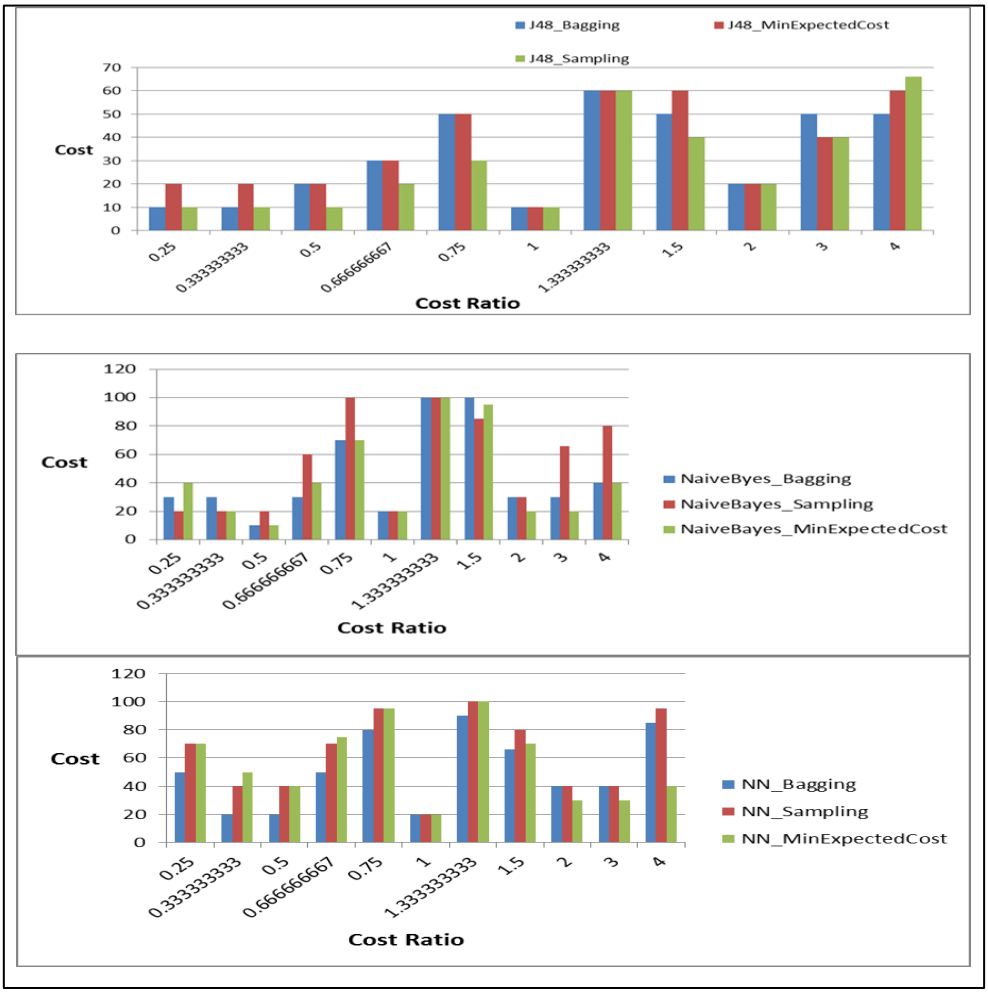

Figure 4-14: Cost ratio with misclassification cost

4.2.2 Cost-Sensitive Meta-Learning System Evaluation

This subsection presents a comparison of the results obtained with an alternative meta-learning system METAL (Berrer, Paterson, & Keller, 2000; Brazdil et al., 2003). The following begins with an overview of METAL, and accordingly presents the results of comparing the METAL meta-learning project and our system.

METAL is a meta-learning assistant project for providing user support in machine-learning with similar goals to the research presented in this thesis, and hence is a good basis for comparison. The main difference with the work presented in this thesis is that METAL uses instance-based learning to rank the recommendations that best suit a specific problem. In METAL, Brazdil et al. (2003) claim that using instance-based reasoning has the advantage of being able to extend the meta-knowledge as soon as new meta data, with its performance, becomes available. The distance function used, $Dis(DS_j, DS_k)$ between two datasets (j *and* k) is given in ***(4-1)***:

$$\boldsymbol{Dis(DS_j, DS_k) = \sum_{i=1}^{m} \sqrt{\left(\frac{DS_j(i) - DS_k(i)}{Stdev(i)}\right)^2}} \qquad \textbf{(4-1)}$$

$Stdev(i)$ *is the standard deviation of the* i*th attribute over all datasets and* $DS_j(i)$ *is* i^{th} Value if the meta features of $\boldsymbol{DS_j}$.

Seven datasets are used for comparison. In this research, data characterisation methods, such as simple mathematical and statistical methods, are similar to those used in the METAL project, except that we also use the ratio of the number of examples in the classes ***exampleRatio*** and the ratio of costs of misclassification ***costRatio***, given that we are interested in misclassification cost as a measure of performance, in addition to classifier accuracy. These two characteristics are important owing to the fact they reflect the importance of one class over the other, which is considered essential in cost-sensitive learning. The datasets used for comparison are taken from the UCI Machine Learning

Repository (Bache & Lichman, 2013), and the misclassification cost is calculated by multiplying the cost matrix values with the confusion matrix values.

In this research, the misclassification cost is also included in the learning process, given the fact that one of the primary aims of this study is contributing knowledge in regard to cost-sensitive learning.

The comparison here is made between the results obtained from our cost-sensitive meta-learning system and the METAL system: the prediction error for each dataset is compared, including each cost-sensitive and in-sensitive method for both accuracy and misclassification cost, and listed in Table 4-15 to Table 4-23. The empirical evaluation is carried out using 10 folds cross-validation. The prediction error is calculated using ***(4-2)***:

$$\textbf{Prediction Error} = \sqrt{(\boldsymbol{p_p} - \boldsymbol{p_e})^2} \qquad \textbf{(4-2)}$$

p_p *is the performance Predicted by the system that uses the meta-knowledge, taking the edge point of ranges get from the meta-knowledge*

p_e *is the actual performance when the algorithm is applied.*

The results of each of the datasets, in comparison to METAL, are presented below.

Diabetes Dataset

The diabetes dataset is a medical dataset taken from the UCI Repository (Bache & Lichman, 2013). The data records whether a patient tests positive or negative for diabetes. The factors used include personal data, such as age, number of times pregnant and the results of medical examination, such as mass index and blood test. There are 768 examples, with 500 examples with Class 1 and 268 examples from Class 2, and 8 attributes. The prediction error, as defined by ***(4-2)*** and using the meta-knowledge described in Section 4.1 for cost insensitive and sensitive classifiers, accuracy and cost prediction errors for the diabetes dataset using

different cost-sensitive and -insensitive methods, is shown in Table 4-15 for both our system and METAL.

METAL prediction error is calculated using ***(4-1)*** with k=3 and the final prediction is the average of the 3 nearest points.

Below is the evaluation result for both accuracy and cost-prediction error, where the shaded box indicates the lowest error prediction.

Table 4-15: Accuracy and cost-prediction error in diabetes dataset

Prediction error	***J48***		***NB***		***NN***		***OneR***		***PART***		***ZeroR***		***Avg Acc***
Diabetes Dataset	**acc**	**cost**	**acc**	**cost**	**acc**	**acc**	**acc**	**cost**	**Acc**	**cost**	**acc**	**cost**	
Bagging	3	6	1	7	2	1	0	1	5	8	7	4	3.0
Sampling	1	3	1	2	2	1	2	3	8	8	2	1	2.7
MinExpected Cost	1	3	1	2	2	1	0	3	8	8	7	4	3.2
Cost-sensitive (none)	1	4	1	2	3	1	0	3	5	8	2	1	2.0
Cost in-sensitive	12	26	1	23	13	24	15	27	10	27	7	34	9.7
METAL	10		3		8	1		3	11		3		6.3
Wrapper methods	***Bagging***		***Sampling***		***MinExpected Cost***				***Cost-sensitive none***				***Cost in-sensitive***
Avg Cost	4.5		3		3.5				3.2				26.8

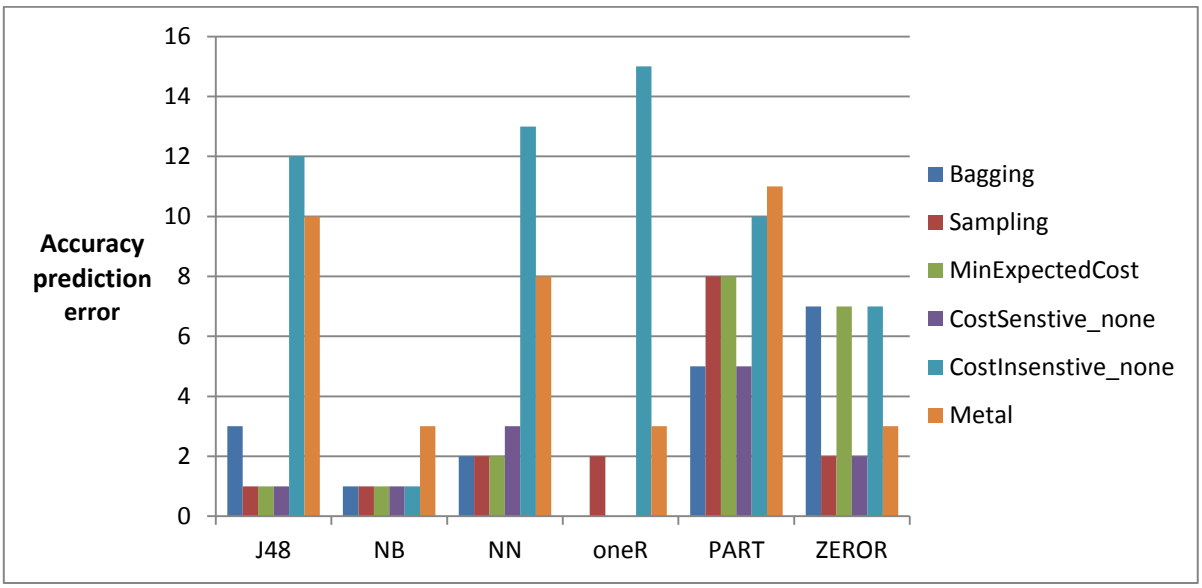

Figure 4-15: Accuracy-prediction error in diabetes dataset

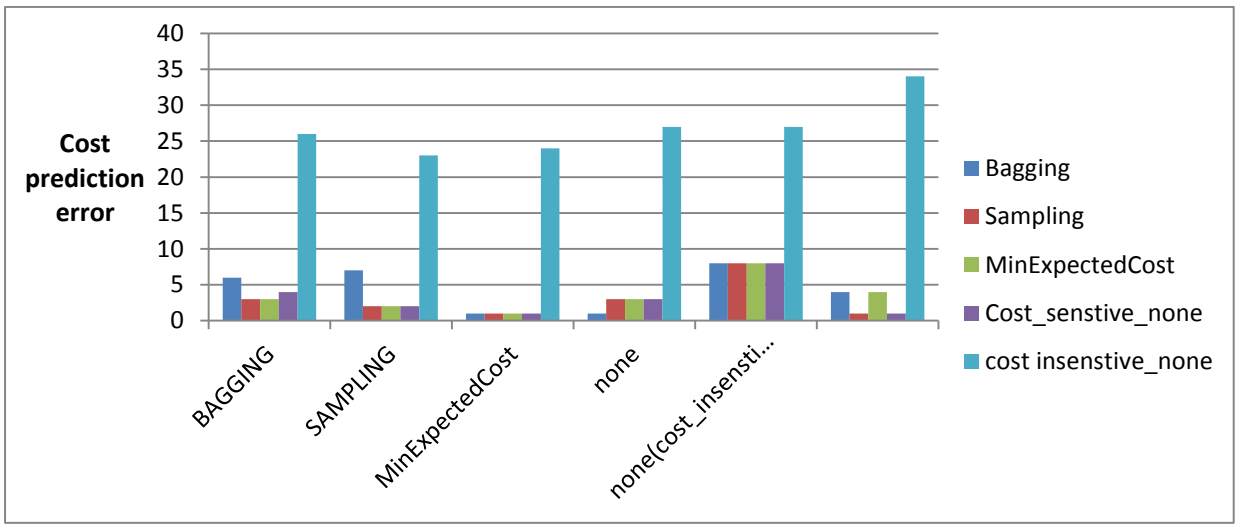

Figure 4-16: Cost-prediction error in diabetes dataset

As shown in Table 4-15, Figure 4-15 and Figure 4-16, Sampling and the Cost-sensitive method without any wrapper method gives the lowest error rate prediction for accuracy, whilst METAL and Cost-insensitive give the highest error rate prediction for accuracy. Sampling and Cost sensitive without any wrapper methods, as well, give the lower error rate for misclassification cost-prediction.

Credit-g Dataset

The problem posed in this dataset is classifying a bank customer as good or bad, which depends on the risk of refunding bank credit. A total of 1,000 examples, with 700 for Class 1 (good) and 300 with Class 2 (bad), are considered, with the data imbalanced with an example ratio (70%, 30%). The cost matrix for this dataset is shown in Table 4.16 taken from UCI repository (Bache & Lichman, 2013). A cost-sensitive meta-knowledge developed is used to predict the accuracy and cost of this dataset based on cost ratio and provided in this cost matrix.

Table 4-16: Cost matrix for Credit-g dataset

	Good	Bad
Good	0	1
Bad	5	0

Figure 4-17 shows the results for the German credit dataset (2 classes, 24 attributes, 1000 observations).

Table 4-17: Accuracy and cost-prediction error in credit-g dataset

Prediction error Credit-g	***J48***		***NB***		***NN***		***OneR***		***PART***		***ZeroR***		***Avg Acc***
	acc	**cost**	**acc**	**cost**	**acc**	**Cost**	**acc**	**cost**	**acc**	**cost**	**acc**	**cost**	
Bagging	14	12	14	9	3	3	6	48	25	26	40	0	17.0
Sampling	10	16	14	2	6	16	16	32	22	30	40	0	18.0
MinExpected Cost	15	27	14	4	4	13	2	63	25	42	35	5	15.8
Cost-sensitive (none)	5	27	5	15	6	21	0	61	9	23	40	0	10.8
Cost_ In-sensitive	0	32	10	20	9	6	0	61	9	23	40	0	11.3
METAL	10		2		3		5		3		30		8.8
Wrapper methods	***Bagging***		***Sampling***		***MinExpected Cost***		***Cost sensitive (none)***		***Cost insensitive***				
Avg Cost	16.3		16		25.7		24.5		23.7				

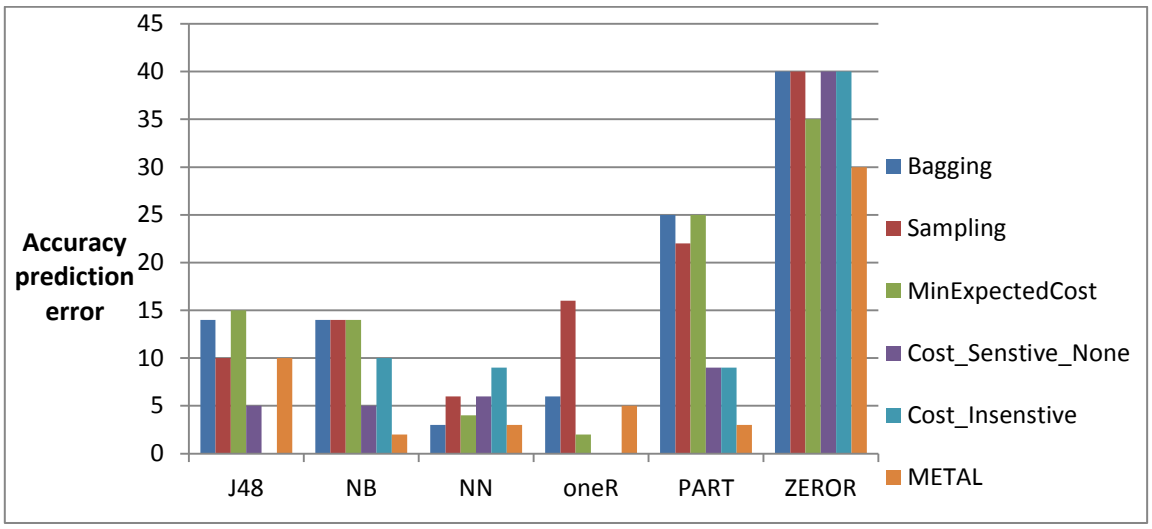

Figure 4-17: Accuracy-prediction error in credit-g dataset

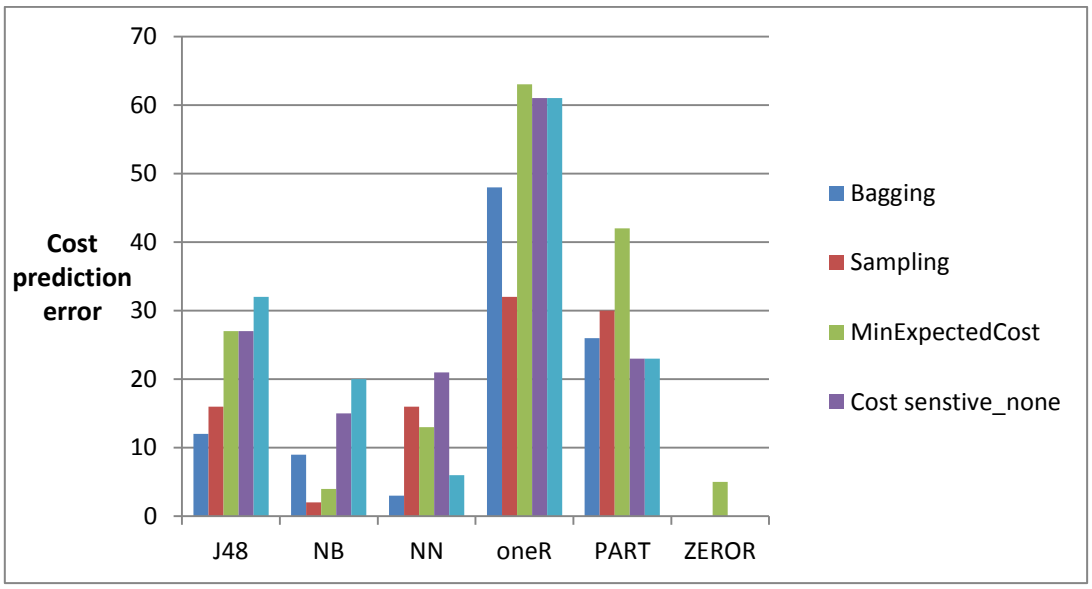

Figure 4-18: Cost-prediction error in credit-g dataset

As shown in Figure 4-17, Figure 4-18 and Table 4-17, METAL and Cost-sensitive methods, without any wrapper, show better prediction accuracy than others, whereas Bagging and Sampling, on the other hand, show better cost-prediction accuracy.

Glass:

This dataset contains examples of types of glass and their attributes. It consists of 214 instances, 8 attributes and 7 classes.

Table 4-18: Accuracy and cost-prediction error in Glass dataset

***Prediction error Glass Cost ratio*=1**	***J48***		***NB***		***NN***		***OneR***		***PART***		***ZeroR***		***Avg Acc***
	acc	**cost**	**acc**	**cost**	**acc**	**cost**	**acc**	**cost**	**acc**	**cost**	**acc**	**cost**	
Bagging	2	3	5	5	23	23	8	7	37	38	20	19	15.8
Sampling	1	2	0	1	17	18	30	29	27	28	20	19	15.8
MinExpected Cost	1	1	3	4	17	18	30	29	27	28	20	19	16.3
Cost-sensitive(none)	1	0	17	17	0	1	2	3	1	1	5	4	4.3
Cost in-sensitive	1	0	3	3	7	8	8	7	28	28	30	29	12.8
METAL	9		25		15		9		10		23		15.2
Wrapper methods	***Bagging***		***Sampling***		***MinExpected Cost***		***Cost-sensitive (none)***		***Cost in-sensitive***				
Avg Cost	15.8		16.2		16.5		4.3		12.5				

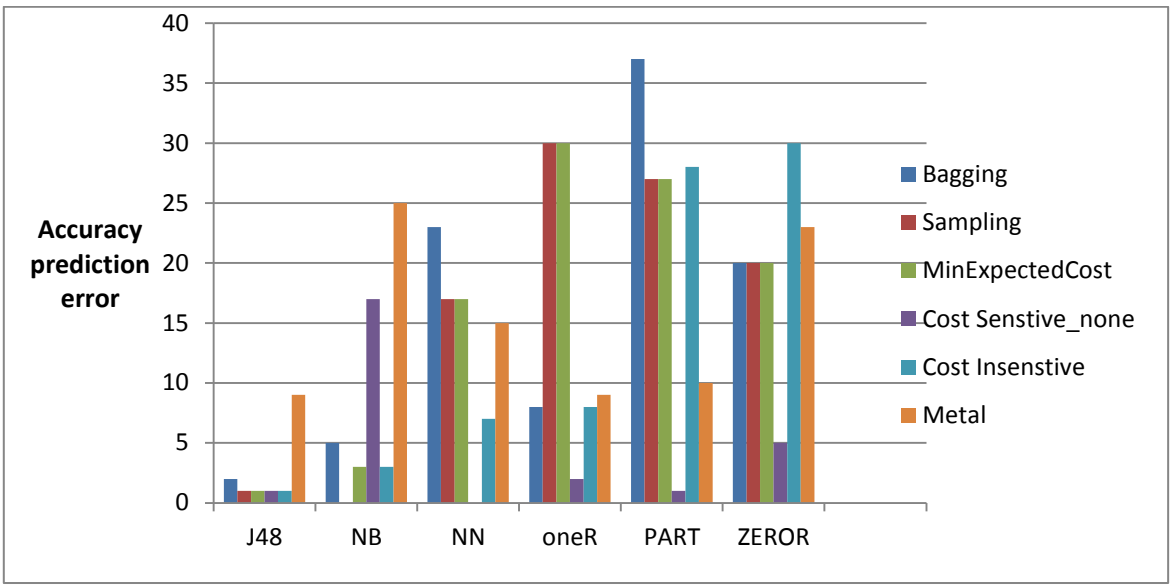

Figure 4-19: Accuracy-prediction error in Glass dataset

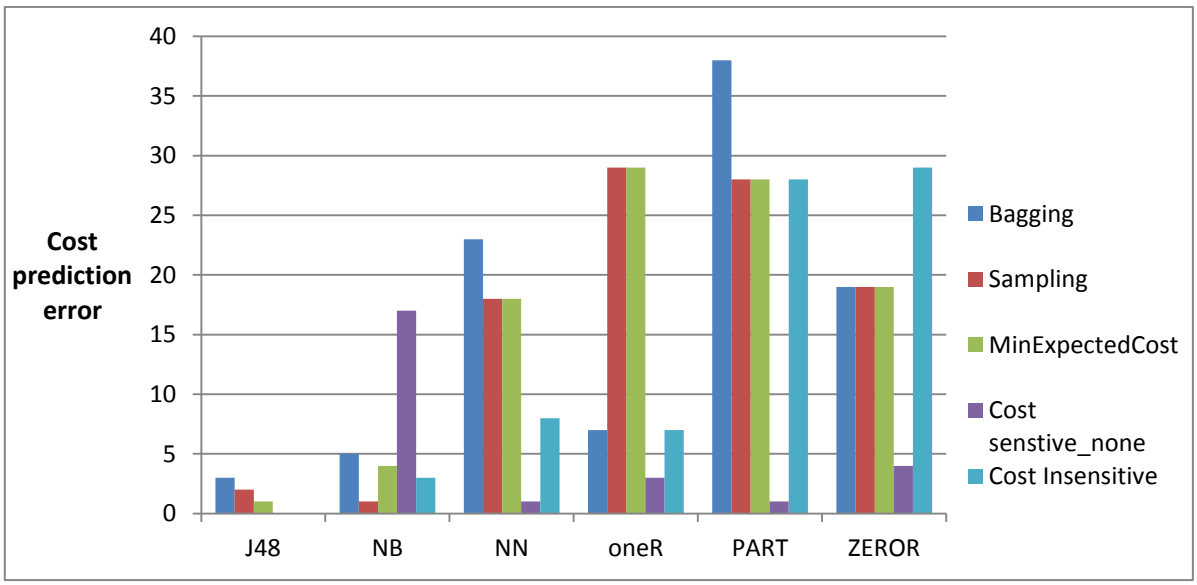

Figure 4-20: Cost-prediction error in Glass dataset

Figure 4-19, Figure 4-20 and Table 4-18 show the accuracy and cost prediction error over different classifiers with different wrapper cost-sensitive methods. Table 4-18 shows that Cost-sensitive methods, without any wrapper, and Cost-insensitive classifiers show better prediction accuracy than others, and also show better cost-prediction for this dataset.

Transfusion Dataset:

The blood transfusion data is from the Blood Transfusion Service Centre (Yeh, King-Jang, & Tao-Ming, 2009). The dataset number of instances= 748, with the number of attributes = 5 and classes= 2.

Table 4-19: Accuracy and cost-prediction error in Transfusion dataset

Prediction error Transfusion Cost ratio=1	*J48*		*NB*		*NN*		*oneR*		*PART*		*ZeroR*		*Avg Acc*
	acc	**cost**	**acc**	**cost**	**acc**	**cost**	**acc**	**cost**	**acc**	**cost**	**acc**	**cost**	
Bagging	2	2	0	0	3	3	1	1	3	3	1	1	1.7
Sampling	2	3	0	0	2	2	1	1	3	3	1	1	1.5
MinExpected Cost	2	2	0	0	2	2	1	5	3	3	1	1	1.5
Cost-sensitive (none)	2	4	0	0	2	2	1	1	3	3	1	2	1.5
cost_ in sensitive	2	2	0	0	3	3	1	1	3	3	1	6	1.7
METAL	3		5		2		4		2		4		3.3
Wrapper methods	***Bagging***		***Sampling***		***MinExpectedCost***				***Cost-sensitive none***		***Cost insensitive***		
Avg cost	1.7		1.7		2.2				2		2.5		

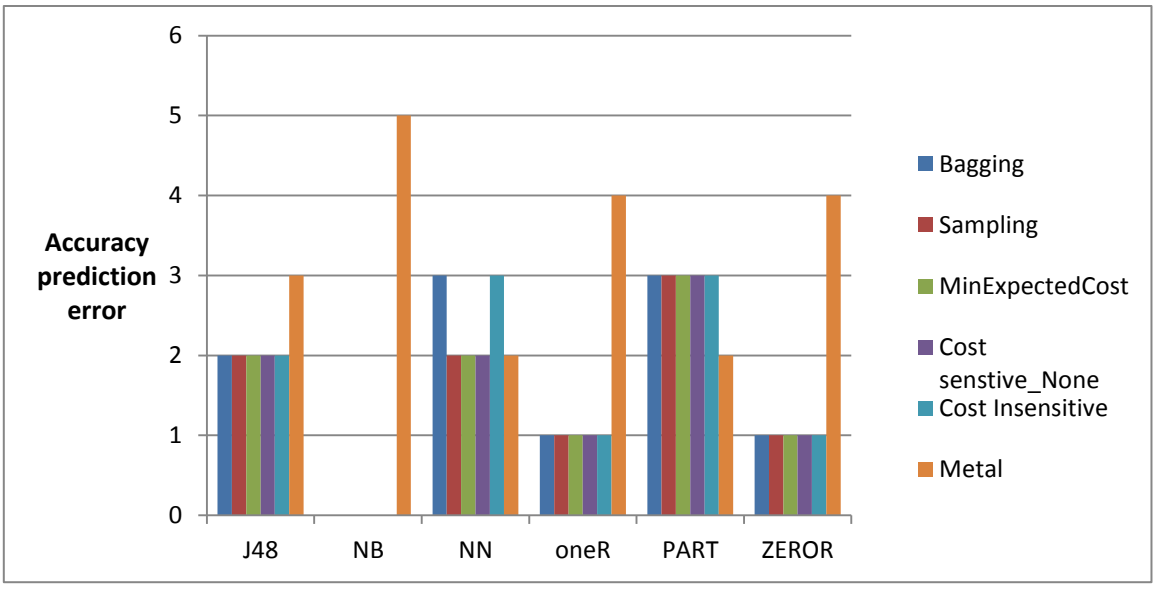

Figure 4-21: Accuracy-prediction error in transfusion dataset

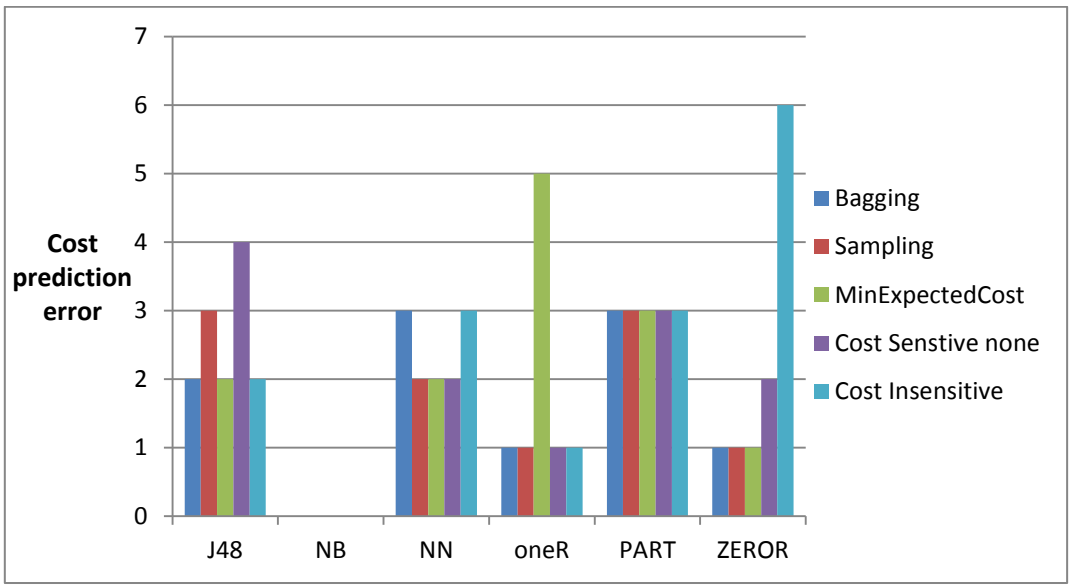

Figure 4-22: Cost-prediction error in transfusion dataset

As shown in Figure 4-21, Figure 4-22 and Table 4-19, Sampling, MinExpectedCost and Cost-sensitive without any wrapper method show best prediction performance, whilst METAL shows the highest error. For cost-prediction, Bagging and Sampling, on the other hand, best prediction performance is achieved. The accuracy and cost-prediction errors are very low for this dataset, which means that the characteristics used to characterise this dataset are very effective in understanding the dataset's behaviour and accordingly gives good prediction performance.

Heart Dataset:

The purpose of this dataset is centred on predicting the presence or absence of heart disease given different medical tests. This data has 2 classes and 13 attributes, and there 270 examples. The cost matrix for heart disease is presented in Table 4-20.

Table 4-20: Heart cost matrix

	Absent	Present
Absent	0	1
Present	5	0

Table 4-21 shows the accuracy prediction error and cost prediction error for heart disease.

Table 4-21: Accuracy and cost-prediction error in heart dataset

Prediction error heart Cost ratio=5	*J48*		*NB*		*NN*		*OneR*		*PART*		*ZeroR*		*Avg Acc*
	acc	**cost**	**acc**	**cost**	**acc**	**cost**	**acc**	**cost**	**acc**	**cost**	**acc**	**cost**	
Bagging	20	9	15	8	8	5	10	22	10	3	20	5	13.8
Sampling	10	9	10	1	12	8	20	47	15	1	10	5	12.8
MinExpected Cost	28	24	9	1	11	11	10	49	12	17	21	5	15.2
Cost-sensitive (none)	15	50	12	6	13	13	24	29	15	24	10	172	14.8
cost_ in sensitive	0	20	13	20	32	47	1	39	20	26	25	147	15.2
METAL	11		8		9		20		10		40		16.3
Wrapper methods	***Bagging***		***Sampling***		***MinExpectedCost***				***Cost-sensitive none***		***Cost insensitive***		
Avg cost	8.7		11.8		17.8				49		49.8		

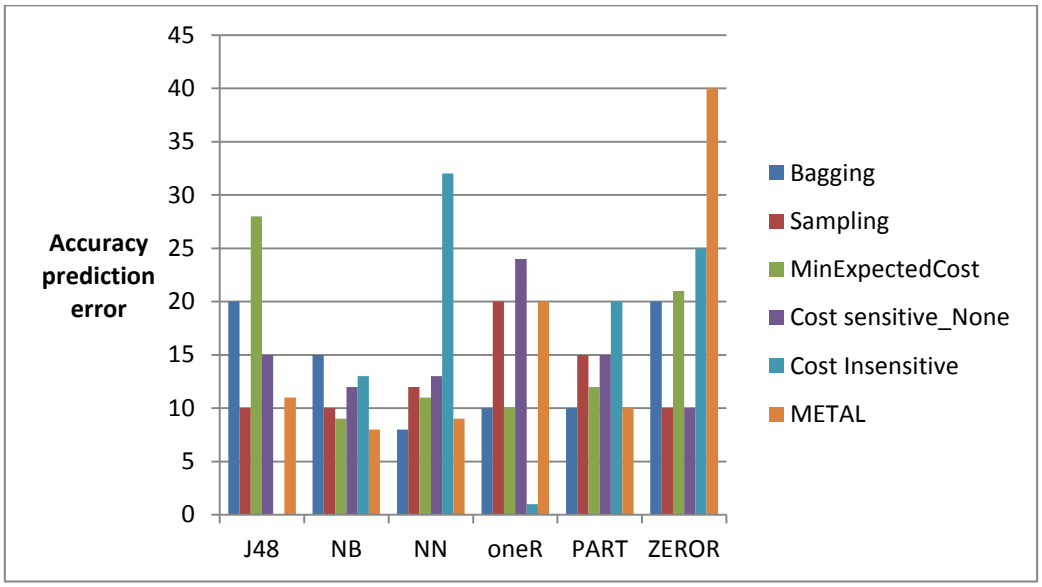

Figure 4-23: Accuracy-prediction error in heart dataset

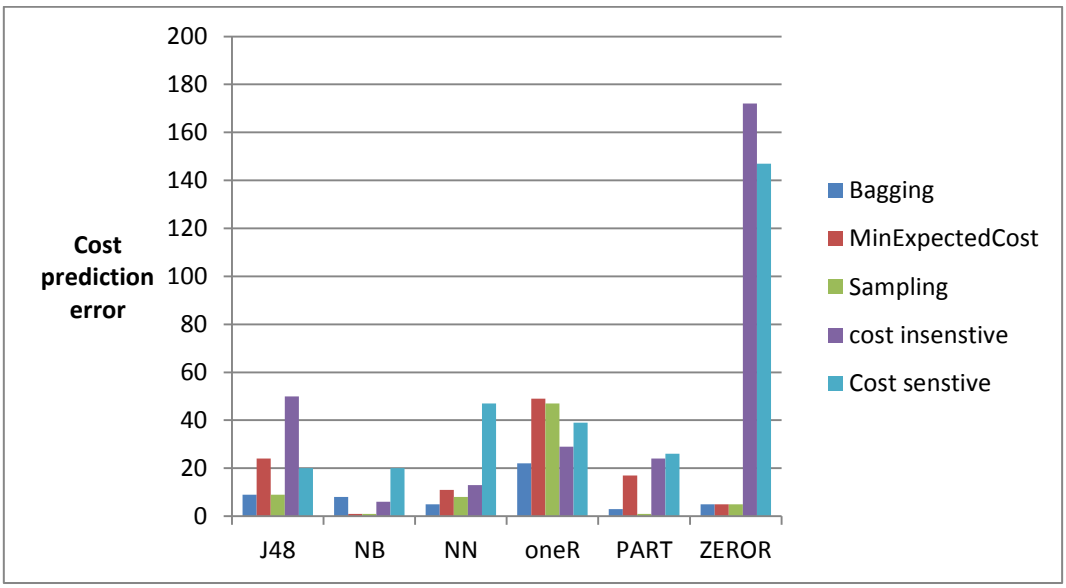

Figure 4-24: Cost-prediction error in heart dataset

As shown in Table 4-21, Figure 4-23 and Figure 4-24, Bagging and Sampling show the best prediction performance for both accuracy and cost for this specific dataset, whereas METAL shows the highest accuracy-prediction error.

Vehicle Dataset

A problem in object recognition is establishing a method of distinguishing 3D objects within a 2D image by the application of an ensemble of shape feature extractors to the 2D silhouettes of the objects (statLog). This contains 940 instances, 19 attributes and 4 classes. Figure 4-22 shows the result of accuracy and cost prediction error for this dataset.

Table 4-22: Accuracy and cost-prediction error in vehicle dataset

Prediction error Vehicle Cost ratio=1	*J48*		*NB*		*NN*		*OneR*		*PART*		*ZeroR*		*Avg Acc*
	acc	**cost**	**acc**	**cost**	**acc**	**cost**	**acc**	**cost**	**acc**	**Cost**	**acc**	**cost**	
Bagging	1	5	0	7	37	32	2	3	5	8	14	10	9.8
Sampling	6	10	1	5	13	10	2	2	5	8	14	15	6.8
MinExpected Cost	6	10	1	5	13	14	3	3	5	5	14	15	7.0
Cost-sensitive(none)	6	10	1	5	13	14	3	3	5	5	14	16	7.0
cost_ in sensitive	14	13	1	5	13	14	38	35	35	38	64	60	27.5
METAL	4		31		2		26		6		47		19.3
Wrapper methods	***Bagging***		***Sampling***		***MinExpectedCost***				***Cost-sensitive none***		***Cost insensitive***		
Avg Cost	10.8		8.3		8.6				8.8		27.5		

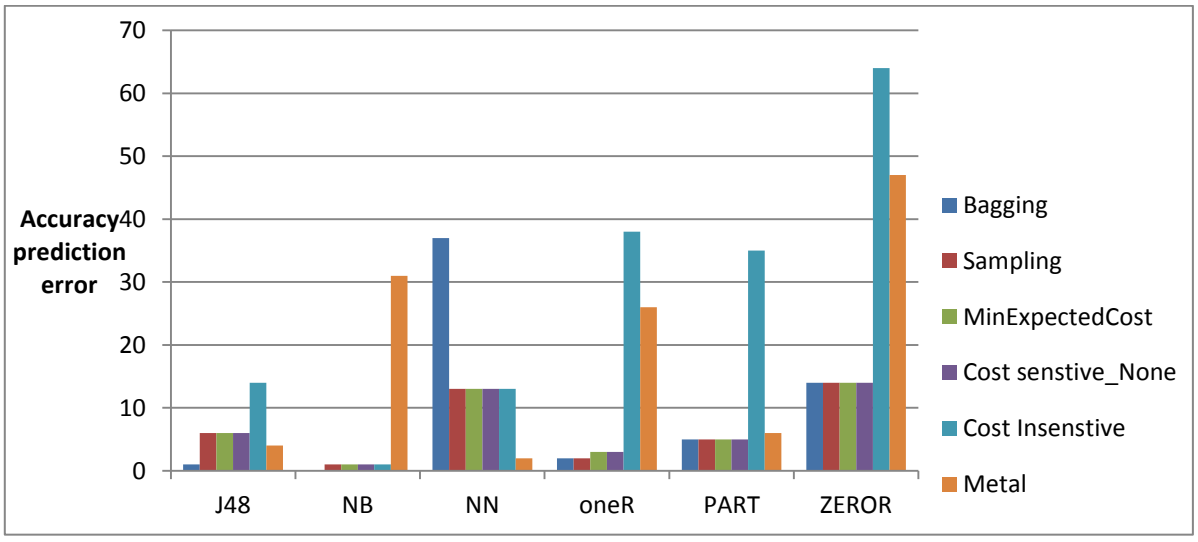

Figure 4-25: Accuracy-prediction error in vehicle dataset

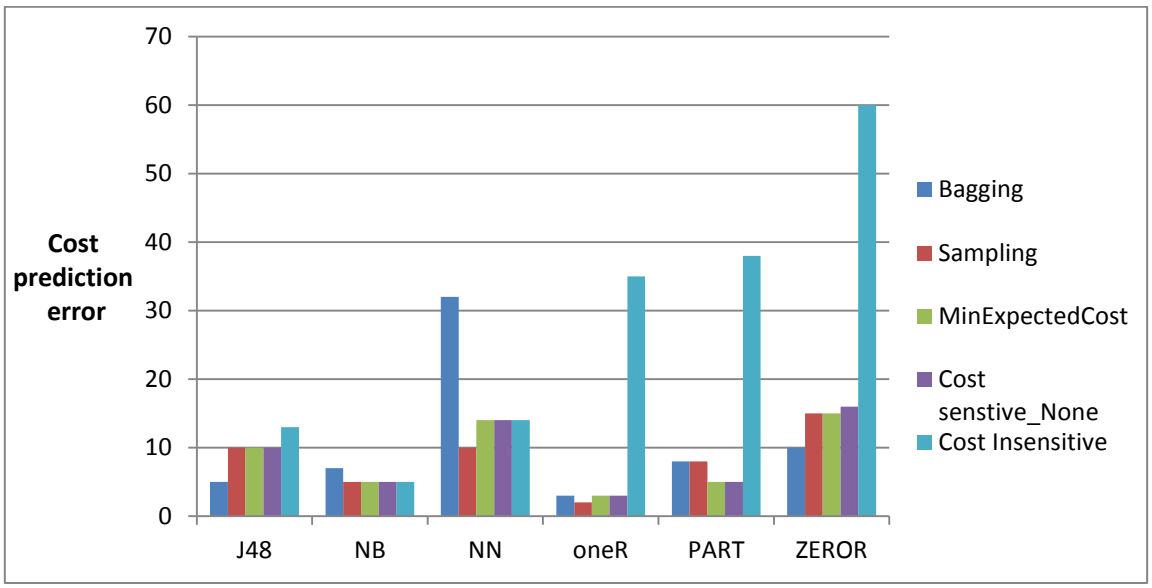

Figure 4-26: Cost-prediction error in vehicle dataset

As shown in Table 4-22, Figure 4-25 and Figure 4-26, Sampling, MinExpectedCost and Cost-sensitive without any wrapper methods show the best prediction performance for accuracy, while Sampling, and MinExpectedCost show the best cost prediction performance for this specific dataset.

Vote Dataset

This dataset classifies a vote type for election and contains 435 instances, with the number of attributes amounting to 16 and classes totalling 2 (democrat and republican).

Table 4-23: Accuracy and cost-prediction error in vote dataset

Prediction error Vote Cost ratio=1	*J48*		*NB*		*NN*		*oneR*		*PART*		*ZeroR*		*Avg Acc*
	acc	cost	acc	cost	acc	cost	acc	cost	acc	cost	acc	cost	
Bagging	1	4.8	0	16.5	1	4.1	0	4.5	4	3.8	4	37	1.7
Sampling	1	4.5	1	17	5	2.5	0	4.1	0	1.5	34	37	6.8
MinExpected Cost	1	4.5	0	17	1	2.5	1	4.5	4	1.5	34	37	6.8
Cost-sensitive(none)	0	5	5	4	1	2.5	1	4.5	4	11.5	34	37	7.5
cost in-sensitive	1	5	5	4	1	2.5	1	4.5	29	66	29	2	11.0
METAL	31		19		13		3		23		2		15.2
Wrapper methods	***Bagging***		***Sampling***		***MinExpectedCost***				***Cost-sensitive none***		***Cost in-sensitive***		
Avg Cost	11.8		11.1		11.2				10.8		14		

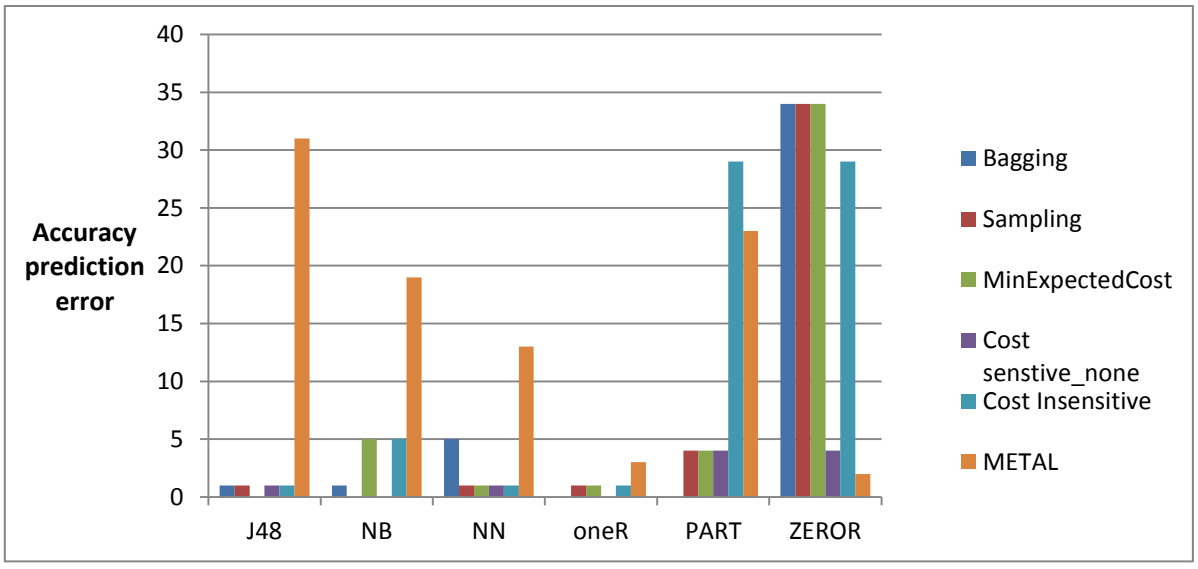

Figure 4-27: Accuracy-prediction error in vote dataset

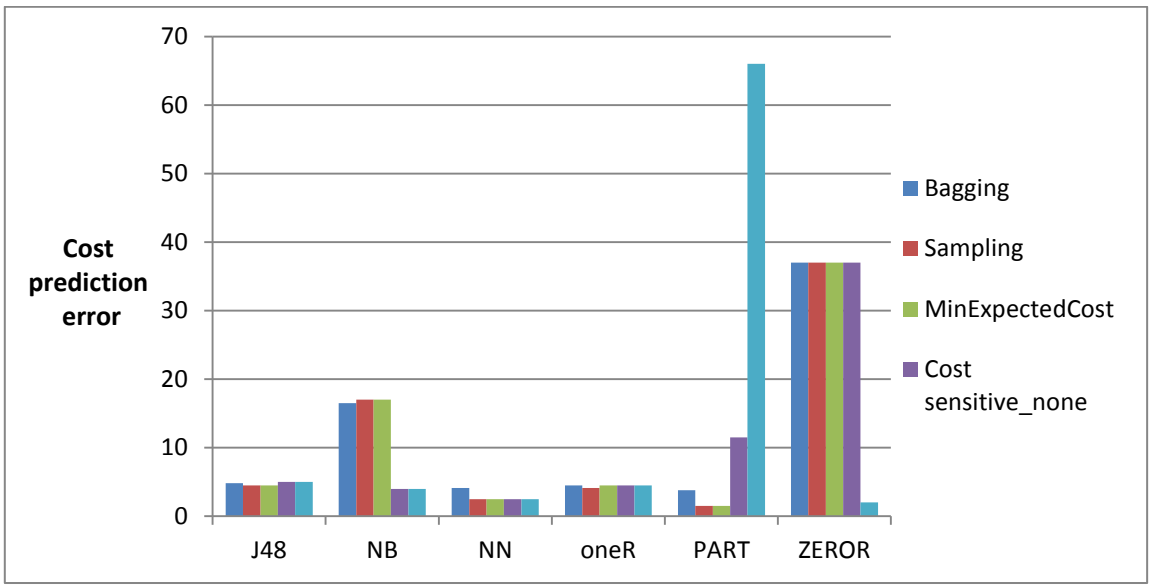

Figure 4-28: Cost-prediction error in vote dataset

Table 4-23, Figure 4-27 and Figure 4-28 shows that Bagging, Sampling, and MinimumExpectedCost show lowest accuracy and Sampling, MinExpectedCost and Cost sensitive without any wrapper methods show lowest cost-prediction error.

4.2.3 Conclusion

This section presents an evaluation of the meta-knowledge learned using the approach developed in this thesis in comparison to the METAL system. The following table (Table 4-24) summarises the relative prediction performance for METAL and our approach.

Table 4-24: Average accuracy-prediction error for all methods used in all compared datasets

Dataset	*Bagging*	*Sampling*	*MinExpected Cost*	*Cost-sensitive none*	*Cost insensitive*	*METAL*
Diabetes	3	2.7	3.2	2	9.7	6.3
Credit-g	17	18	15.8	10.8	11.3	8.8
Glass	15.8	15.8	16.3	4.3	12.8	15.2
Transfusion	1.7	1.5	1.5	1.5	1.7	3.3
Heart	13.8	12.8	15.2	14.8	15.2	16.3
Vehicle	9.8	6.8	7	7	27	19
Vote	6.8	6.8	7.5	1.7	11	16
Avg accuracy-prediction error	9.7	9.2	9.5	6.1	12.6	12.1

Table 4-24 provides the average prediction error for each dataset, utilising different cost-sensitive and -insensitive techniques. This table shows that our cost-sensitive system provides better accuracy-prediction error in all tested datasets than METAL, except in the case of Credit-g. It has been shown in many papers that credit-g performs well using K-NN as meta-learner because, in the credit-g dataset, the attributes are carefully selected, and K-NN works very well when there are no irrelevant attributes.

As has come to our knowledge, this is the first work that uses meta-learning methods to recommend which cost-sensitive plan is best for a specific task. Importantly, this is motivated by the idea that, although a wealth of work has been carried out on converting the learning process from accuracy-based algorithm to cost-based algorithm, there is no systemic meta-knowledge that provides a prediction in terms of which cost-sensitive plan should be used for a specific dataset.

The result shown in this section can be concluded that the use of J48 as a meta-learner provides better prediction power over using K-NN, where this is very sensitive to irrelevant attributes, as any irrelevant performance evaluations mean new predictions are highly affected by k number. We conducted our experiments on k=3, meaning we may have a very different result if k is assigned to a different number. The conclusion obtained is the same

result revealed by Prudancio, De Souto, & Ludermir. (2011) where K-NN as a meta-learner is used for a small number of meta examples and a large number of meta examples (some of them generated by using datasetoids (explained in Section 2.4.2), comparing the results of both shows that K-NN as meta-learner has better performance when the number of examples is small as K-NN is sensitive to the irrelevant meta examples potentially produced by datasetoids. In addition, K-NN is very sensitive to the choice of k number.

4.3 Active Learning

Chapter 3 presented the development of an active learning methodology that proposed an approach for improving the meta-learning process by selecting the most informative data for the learning process. The Literature review in Section 2.4 revealed that there is some works in active learning over meta-learning, with the aim of speeding-up the meta-knowledge development process (see Section 2.4.2). A new active learning based on the clustering algorithm is proposed and evaluated. The aim of this evaluation is evaluating the effect of using active learning based on the clustering method in meta-knowledge prediction error by comparing the meta-knowledge prediction error created by adding informative data (selected by ALBC (Active Learning Based on Clustering) and between using passive learning that randomly selects data.

Active learning is implemented and tested using 56 datasets with a variety of meta-features. The datasets are divided into two sets: a total of 6 are taken as the initial set of labelled examples, whilst the remaining 50 are taken as unlabelled examples from which the algorithm will choose. The experiment is carried out as follows: first, the data is characterised, with different classifiers then applied to the initial datasets. Subsequently, each dataset is evaluated using 10 folds cross validation and assigned a label (accuracy and cost), with the labelled datasets then grouped into different classifiers, where each data group is clustered on its meta-features, with new data needed for each cluster then selected from the unlabelled pool using the ALBC algorithm described in Section 3.2. These stages are described in the subsections below.

1. Initial datasets are characterised using different dataset characterisation methods, where Table 4-25 shows the initial datasets randomly chosen for labelling:

Table 4-25: Dataset characterisation for small pool of dataset

Number Of classes	*Number of attribute*	*Class entropy*	*Class skew*	*Number of instances*	*Conditional entropy*
2	10	0.88	0.29	286	0
3	5	1.33	0	24	3.72
2	9	0.93	0.38	768	9.21
7	10	2.18	1.82	214	7.72
2	35	0.94	-8.59	351	8.01
3	5	1.58	0.06	150	7.14

2. Learning application and performance evaluation: To label the initial data, different classifiers are applied to the previous datasets, with the performance (accuracy and cost) of each dataset then evaluated using 10 folds cross-validation. Table 4-26 shows the accuracy obtained from different classifiers for one dataset aligned with its data meta-features.

Table 4-26: Accuracy of applying different classifiers to the previous datasets aligned with its meta-features

Number of classes	*Number of attribute*	*Class entropy*	*Class skew*	*Number of instances*	*Conditional entropy*	*Classifier*	*Accuracy*
2	5	0.94	-0.06	14	3.07	J48	45-50
						Naive Bayes	70-75
						OneR	40-45
						Part	55-60
						ZeroR	60-65
						Neural Network	70-75

3. Data is grouped into different classifiers group:
   Data is divided into different groups as shown in Table 4-27.

Table 4-27: Accuracy results of applying J48 into the first set of data

Number of classes	*Number of attribute*	*Class entropy*	*Class skew*	*Number of instances*	*Conditional entropy*	*Classifier*	*Accuracy*	*Cost*
2	10	0.87	0.19	277	0	J48	75-80	20-30
3	5	1.32	0	24	0	J48	80-85	10-20
2	9	0.93	0.37	768	9.20	J48	70-75	30-40
2	5	0.94	-0.06	14	3.07	J48	45-50	40-50
2	35	0.94	-8.58	351	8.01	J48	90-95	5-10
3	5	1.58	0.06	150	7.14	J48	>95	<5

4. *Clustering*

As mentioned previously, clustering is done in order to form the initial representative data and accordingly seek more informative data for each cluster.

5. *Select Query*

Each informative dataset is selected using the select query algorithm, as described in Figure 3-10. The results obtained are discussed below.

4.3.1 ALBC Comparison Result with Random Selection

In order to evaluate the ALBC algorithm, a comparison is made between using this algorithm on initial clusters formed from a small pool of labelled datasets. The main aim of this part is seeking more data from large unlabelled datasets (50 datasets), using both algorithms (ALBC and random), and then drawing a relation between meta-knowledge prediction error and a number of labelled examples. Meta-knowledge prediction error is calculated using the following steps:

Initial labelled datasets are used to create a meta-knowledge that predicts the performance of applying a specific classifier into a given dataset. Leave-one-out experiment is

performed to evaluate the performance of meta-learner in accuracy and cost prediction. on each step one dataset is left out for testing and the rest is taken to create the meta-knowledge, the meta-knowledge prediction performance is calculated using the average accuracy of all cases giving the chance for all dataset to enter the testing process, leave one out method is used here rather than 10 folds cross validation because we started with a small number of meta-examples (eventually 6). More datasets are added to the meta - example pool using both methods (ALBC and random selection),
and each time, the meta-knowledge is created and prediction performance is evaluated. The relation between meta-knowledge prediction error and the number of meta-examples for both methods (ALBC and random selection) is illustrated for each learning algorithm (Figure 4-29 and Figure 4-30).

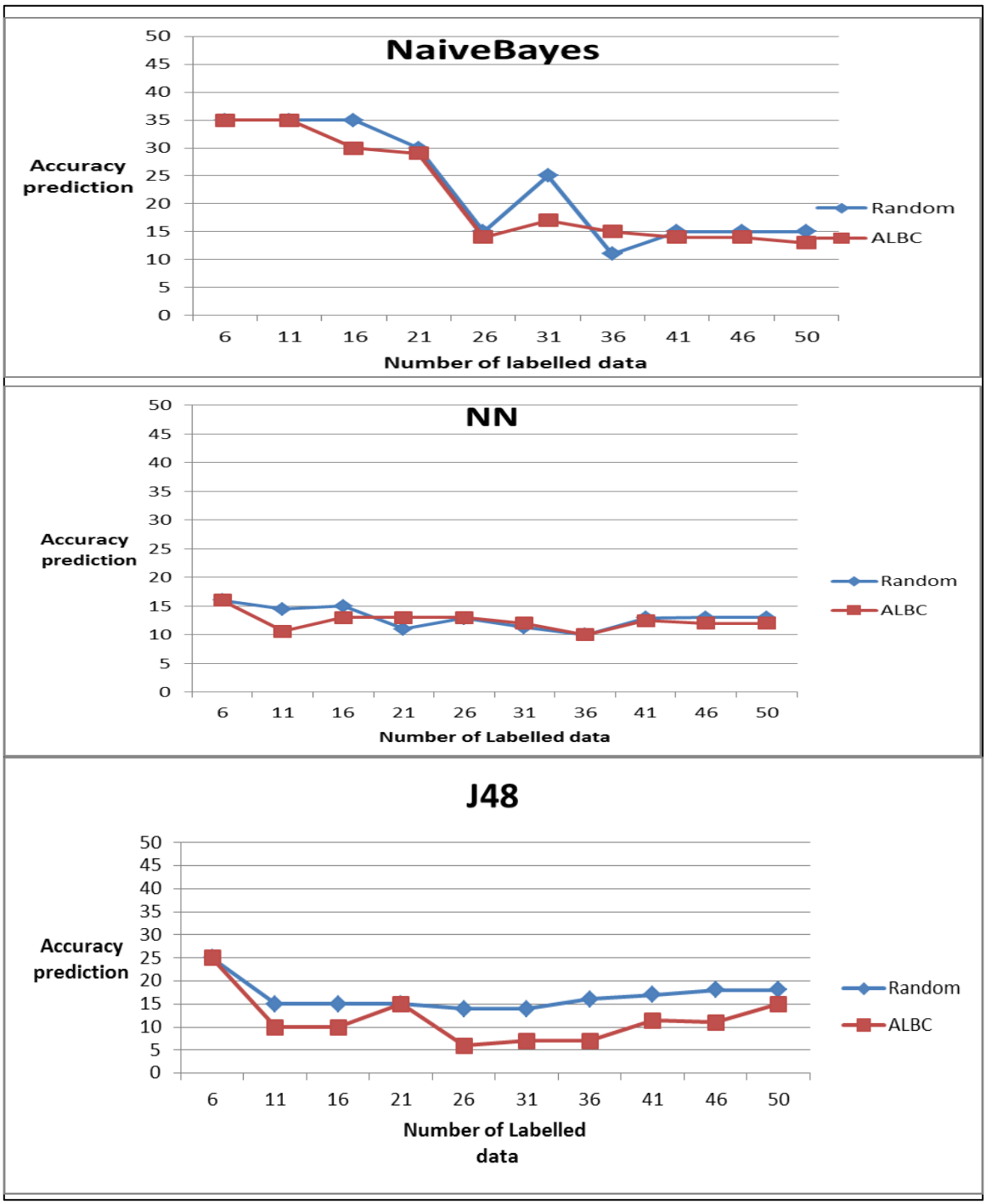

Figure 4-29: Accuracy prediction error with number of labelled examples

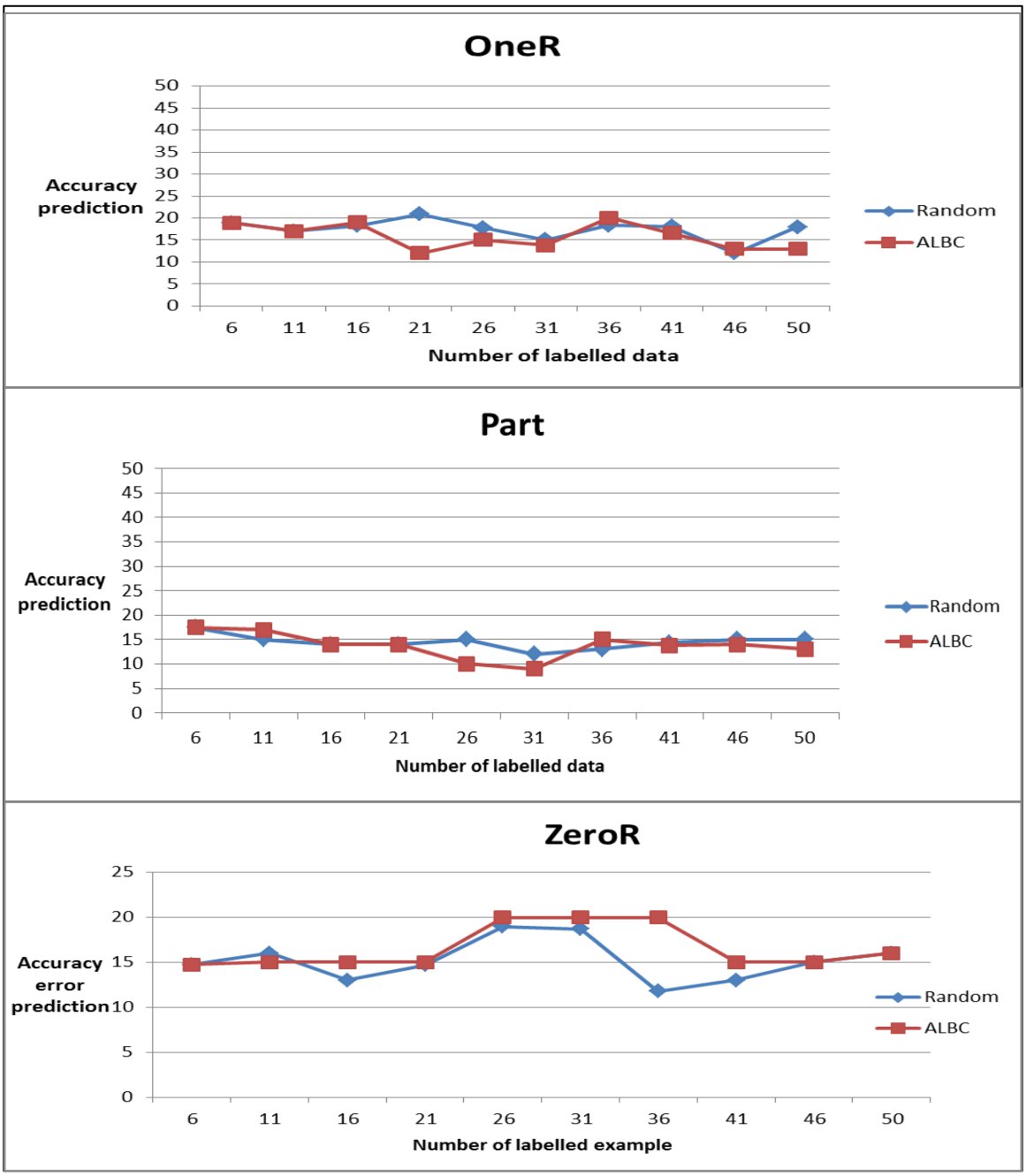

Figure 4-30: Accuracy prediction-error with number of labelled examples

4.3.2 Results Discussion

In absolute terms, the results obtained through the use of ALBC are better than those obtained from randomly choosing the data to be labelled for most of the used classification methods except the ZeroR classifier (see Figure 4-29 and Figure 4-30).

Ideally, the error rate should decrease as the number of labelled examples increases; however, this is not the case in all time, in some cases the curves goes up and down. In order to address this, as described in chapter 3, the active learning process was further refined by adopting a best-first search method; that is, the best cluster, which gives the minimum error rate during all active learning processes to date, is selected and expanded (see Section 3.2).

4.3.3 Clustering Based on Best Cluster

The experiment adopted here is to compare the ALBC method, which is based on the 'best cluster' methods that is described in Section 3.2, with one that randomly chooses the labelled data. This experiment is carried out on 56 datasets, where 6 datasets are initially taken as the first set of labelled data, with those datasets then clustered based on the best cluster that provides the minimum error rate from previous experiments. Subsequently, more data is selected using the ALBC algorithm as summarised in Section 3.2.

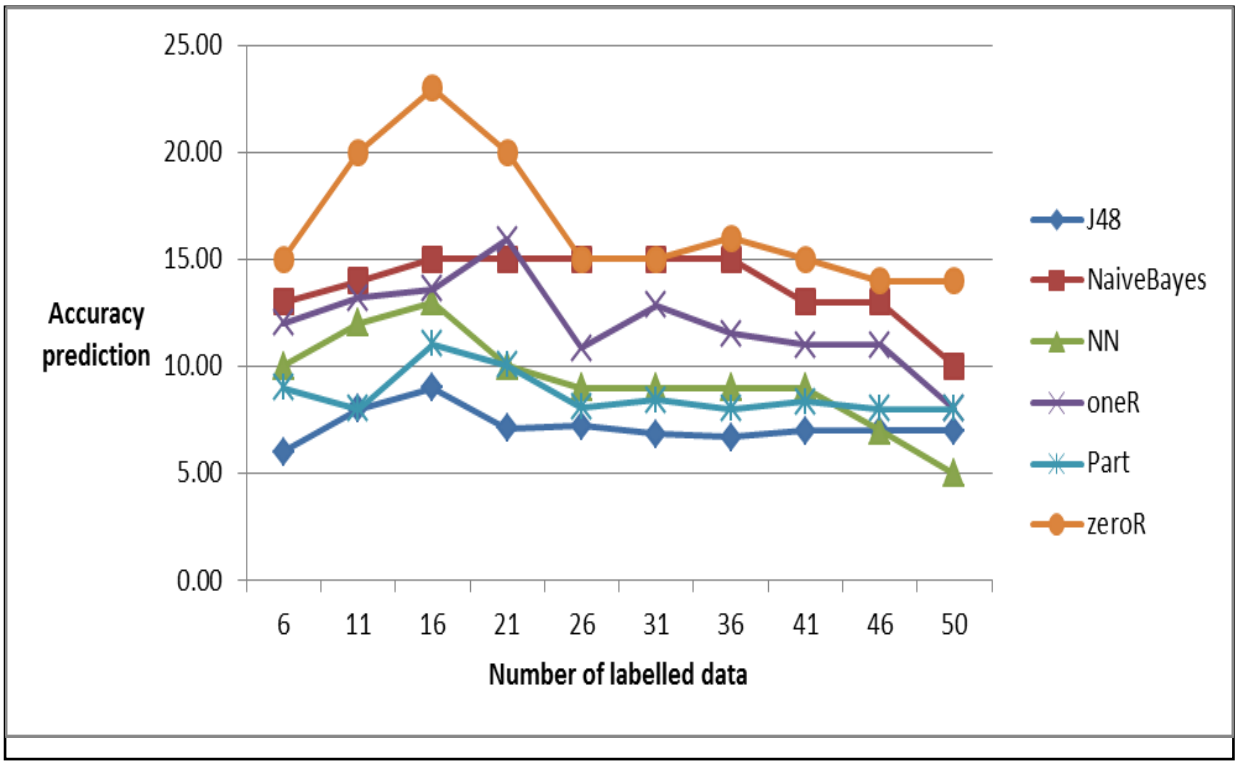

Figure 4-31: The results for learner's accuracy-prediction error based on 'best cluster' methods

Saving the clusters that gives minimum error rate and clustering each new data on this cluster gives better results than previous experiments in terms of the error rate decreasing whilst increasing the number of labelled data, as shown in Figure 4-31.

4.3.4 Conclusion

In this section, we present the use of active learning to support the classifier to query more informative examples, which will be used to enter the labelling process. The work done here is different to other active learning works, covered in Section 2.4. The clustering in previous works has been done over unlabelled examples with the aim of finding the most representative data (usually around centres) to enter the labelling process (Nguyen & Smeulders, 2004; Xu et al., 2003; Zhu et al., 2008a). In contrast to previous work, this methodology depends on clustering the first set of labelled data and accordingly seeking

more informative data that improves each cluster's predictive power. Importantly, this can be done by dividing each cluster into three zones, depending on the distance between cluster centres and cluster boundaries, and accordingly searching for data characters that best improve cluster predictive power.

In this section, the ALBC algorithm is applied and tested using six labelled datasets and 50 unlabelled datasets, and accordingly is evaluated using leave one out validation process with 6 classifiers. The results obtained through the ALBC approach, in general, are better than those achieved through the random selection method in most cases. The methodology used in increasing a meta-knowledge predictive power contributes to both the meta-learning and active learning fields

The next chapter gives a conclusion that summarizes the achievements and contributions made by this study.

CHAPTER FIVE: CONCLUSION AND FUTURE WORK

Supervised learning is one particular area of data mining where learning is carried out through the use of examples with class labels with the aim of learning a model that is adopted in an effort to estimate unseen examples' class labels; however, the learning process encompasses a number of costs, including misclassification costs, the costs associated with obtaining data, and test costs.

In the field of data mining, cost-sensitive learning is recognised as an active area of research that focuses on handling different types of costs. In the literature, a number of different approaches have been devised in an effort to deal with different kinds of costs, such as the cost of tests and misclassification costs. A number of academics have directed their efforts towards developing approaches and classifiers that consider misclassification costs; however, the most suitable cost-sensitive classifier for a given data set and problem remains unknown. Hence this thesis has aimed to study the use of meta-learning in an effort to establish the link between problem characterisation and the cost-sensitive method that is most appropriate for the learning area in question.

Meta-learning is defined as creating knowledge that associates the problem characteristics with the data mining algorithm's performance, considering this process in much the same way as any other learning process that centres on 'learning from learning'.

A number of the feature selection approaches and cost sensitive data mining approaches have been devised and introduced during the last decade; however, establishing which are the most valuable is not simple, with no best method recognised amongst the options. Accordingly, this study has examined whether it is feasible to devise a meta-learning system that can be adopted in mind of selecting a cost-sensitive data mining algorithm for a particular data set, taking into account misclassification costs.

Considering the ever-growing volume of data mining algorithms, along with the presence of large-scale data that may be prone to misleading or irrelevant characteristics, the thesis has explored the following questions:

> *Which of the cost-sensitive learning algorithms should be adopted for a particular data set, and what feature selection approach should be applied in an effort to eradicate distracting or otherwise irrelevant attributes?*

Section 5.1 summarises the objectives set to address this question and reviews the extent to which the objectives have been achieved.

5.1 A Review of the Research Objectives

This section discusses the research objectives, and accordingly reviews the degree to which they have been fulfilled:

1. **To carry out an in-depth, comprehensive literature review centred on the present data mining approaches, their use in meta-learning, cost-sensitive learning, and active-learning**. A literature review for cost-sensitive learning methods has been carried out, with meta-learning methods and their use in the data mining field also widely covered and associated with the study area. A general recognition for the value of cost sensitive learning is found in the literature with a growing volume of data mining algorithms that are cost sensitive, but there is no single cost sensitive learning method that is found to be the best for all cases. Chapter two also presents an in-depth review of the feature selection methods, search strategies and their impact on the learning process. The literature review highlighted that there are a number of feature selection methods though there is no single method to suit all cases. Likewise, the literature review identified several cost-sensitive methods but no single method that could be applied successfully in all cases. The literature includes some studies of meta-learning such as the METAL project which recommends the algorithm for a given problem using K-NN meta-learner. However, none of these focus on the cost-sensitive learning algorithm.

2. **To devise a meta-learning system with the capacity to make suggestions concerning data-mining approaches that take cost into consideration.** Chapter three

develops a new meta-learning system with the ability to suggest the most suitable feature selection method with a set of classifiers (or just one) that are best suited to a particular data set. The meta-knowledge for recommending feature selection was learned by applying different feature selection methods on forty two data sets and the results together with the characteristics of the data sets combined to form the meta-examples. A decision tree learner was then used on the meta-examples to obtain the meta-knowledge for feature selection. A similar approach was used for learning the meta-knowledge for recommending cost-sensitive learning but with the added complication that costs had to be considered. The meta-knowledge for recommending cost sensitive algorithms was learned by applying different wrapper cost sensitive methods on twenty six data sets and the results together with the characterization of the datasets combined to form the meta-examples. A decision tree learner was then used on the meta-examples to obtain meta-knowledge that can be used to rank the performance of different cost sensitive learning algorithms, allowing a user to select the most appropriate algorithm for their needs.

3. **To devise an active learning approach that provides learners with the ability to choose the most informative data for the learning process, and accordingly quicken the learning process**: Section 3.2 presents a new and innovative active learning algorithm that aims to accelerate the meta-learning process. The approach is based on clustering the meta-features and performance for each learning algorithm and seeking new data sets that can improve the clusters' predictive power. The identification of knowledge in this manner reduces the time and the effort the learner has to expend in considering data that doesn't contribute to the generated meta-knowledge. The approach adapted in this thesis is to use active learning based on clustering by establishing good representative samples for a potentially wide range of datasets that needed to create meta-knowledge. These clusters seek data that rapidly improves meta-knowledge accuracy each time data sets are added. The algorithm proposed in this thesis is novel in that it uses z-scores to provide bands in a cluster that aid in assessing the quality of the clusters and enable identification of suitable data sets that could be added to help improve the predictive power of future clusters.

4. **To conduct an empirical evaluation of the meta-learning approach, and accordingly contrast the findings with another meta-learning system—namely METAL:** The cost sensitive meta-learning system developed provides answers to the questions detailed earlier by learning the knowledge about the performance of different data mining algorithms on a number of different data sets. This meta-learning system encompasses a number of different algorithms, including J48, naive bayes, neural network, part, zeroR and oneR, the performance of which is calculated by completing tests on a variety of datasets, including a large number of characteristics.

   Chapter 4 presents the result of an empirical evaluation of the meta-learning system for recommending feature selection methods, and meta-learning for recommending cost-sensitive methods. The results for feature selection show the importance of factors such as the number of attribute, class entropy, class skew and the classifier used. Those factors affect the knowledge about which feature selection method would be recommended to be used in a specific dataset. On the other hand, the knowledge produced about the performance of cost-sensitive learning algorithms, shows the importance of factors such as ratio of classes, and cost ratio. For example, using sampling for cost sensitive learning with J48 on a dataset with example ratio > 0.6 gives an accuracy of 40-45% if the cost ratio > 1, whilst using sampling with J48 on a dataset with example ratio <=0.6 results in accuracy in the range 60-65% if number of attribute <=10, both having class skew <=0.4 and class entropy >0.9 (see Table 4.14).

   In a comparison to METAL, an evaluation is conducted on seven datasets by comparing the prediction error, which is the difference between classifier performances using a true empirical evaluation based on 10 folds cross validation and between classifier performances guided by the developed meta-knowledge. The results show that cost-sensitive meta-learning outperforms METAL in the majority of the cases.

5. **To assess the active learning approach devised in this research by drawing a contrast between the findings obtained and those from a passive learning approach that randomly selects data**. The generated knowledge provided in point 3 undergoes consistent improvement through the learning process, particularly by making use of an active learning method that utilises a wealth of available unlabelled data, and a small volume of labelled data, in order to choose the most informative data for the

learning process. Section 4.3 presents an evaluation of the active learning process developed in this study. The methodology adopted included 50 unlabelled datasets and 6 labelled datasets utilising different algorithms, namely J48, neural network, naïve bayes, part, zeroR and oneR. The evaluation highlights that active learning based on clustering outperforms the use of passive learning based on random selection for the majority of the tested data. The active learning based on clustering algorithm is improved through devising an active learning approach centred on a new, innovative 'best cluster' method, and one which provides a minimum error rate. The findings show that in the following of this clustering on a minimum cluster method, the error rate is seen to continuously decline when labelled data is incorporated into the learning process.

To conclude, the key contributions of this study are as follows:

1. A meta-learning system has been developed that learns the meta-knowledge capable of suggesting the most applicable algorithms for feature selection and cost-sensitive data mining. The work done has been contrasted with the METAL scheme—a widely recognised project in the domain of meta-learning. Importantly, an empirical evaluation shows that the cost-sensitive meta-learning system developed in this study outperforms METAL in relation to the majority of the datasets undergoing testing.

2. Based on the literature search, this is the first study that develops and utilises a meta-learning system in deducing knowledge about the performance of cost-sensitive learning methods.

3. The research has introduced an innovative, active learning system based on clustering, with the algorithm accelerating meta-learning through choosing the new datasets for labelling in such a way that it enhances cluster quality. The active learning system for meta-learning has been compared with a passive method that selected data sets randomly and the findings highlight that active learning based on clustering performs better across all of the tested data.

5.2 Future Work

This section will focus on a number of different directions for future work.

1. Alternative cost-sensitive approaches

There are a number of different cost-sensitive algorithms, and the system and method devised in this study have made use of a number of these for inclusion as one element of the meta-learning process. Those implemented in the research make use of wrapper-based methods that produce cost-sensitive classifiers from cost-insensitive ones. The benefit of this approach is that it is able to deal with the algorithm as a closed box, without the need to implement any changes in regard to its internal behaviour. Future works could encompass experiments with various other algorithms, including ICET (Turney, 1995), EG2 (Nunez, 1991) and CSNL (Vadera, 2010) for example.

2. Alternative Meta-Learner

In this research, there has been the adoption of J48 as a meta-learner owing to the fact that J48 is simple to use, its results are simplistic and therefore can be easily converted into rules. A different type of meta-learner could be adopted, such as RandomForest, for example, which is an ensemble approach, comprising boosting with replacement (bagging) with decision tree base learner that grows upon randomly assigning parameters, where each single tree on each sample is created through randomly choosing features and accordingly calculating the most suitable divide in relation to the randomly chosen features. Importantly, this classifier is suggested by Breiman (2001) and gives a good result in different studies such as (Sun, 2014) .

3. Model Selection

In addition to choosing an algorithm, there is also the need for a data mining user to choose the most suitable parameters for an algorithm. For instance, in the context of decision tree learning, there is the necessity to experiment with the amount of pruning. There is also the

need to choose the learning rate for a neural network. In order to circumvent this issue, a number of scholars have adopted meta-learning in an effort to provide a map between the performance of different classifiers with different parameter settings for each classifier, and dataset characteristics (Alexandros & Melanie, 2001; Sun, 2014). This would be done by doing the following:

> (1) a set of meta-features that characterises the problem under the domain is developed; (2) different parameter settings are chosen, along with their performance on different classifiers at base-level learning; and (3) a meta learner is used to build a model that predicts the performance of different algorithm settings on different problems or otherwise to predict the best algorithm parameters values (amongst a set of candidates), producing the best performance based on each data problem meta-features.

In conclusion, this thesis has contributed to the field of cost-sensitive data mining and meta-learning by developing a novel system for meta-learning, including the use active learning for accelerating learning.

References

Abdi, H. (2007). Z-scores. *Encyclopedia of measurement and statistics. Thousand Oaks, CA: Sage*.

Aha, D. W. (1992). Generalizing from case studies: A case study. *Proc. of the 9th International Conference on Machine Learning*, 1-10.

Aha, D. W., & Bankert, R. L. (1994). Feature selection for case-based classification of cloud types: An empirical comparison. *AAAI-94 Workshop on Case-Based Reasoning*, 106-112.

Alexandros, K., & Melanie, H. (2001). Model selection via meta-learning: a comparative study. *International Journal on Artificial Intelligence Tools, 10*(04), 525-554.

Almuallim, H., & Dietterich, T. G. (1991). Efficient algorithms for identifying relevant features. *Proc. of the 9th Canadian Conference on Artificial Intelligence*, 38-45.

Aminian, M. (2005). Active learning for reducing bias and variance of a classifier using Jensen-Shannon divergence. *Machine Learning and Applications, 2005. Proceedings. Fourth International Conference*, 1-6.

Anderson, M. L., & Oates, T. (2007). A review of recent research in metareasoning and metalearning. *AI Magazine, 28*(1), 12.

Angluin, D. (1988). Queries and concept learning. *Machine learning, 2*(4), 319-342.

Bache, K., & Lichman, M. (2013). UCI machine learning repository. *URL http://archive. ics. uci. edu/ml, 901*.

Balte, A., Pise, N., & Kulkarni, P. (2014). Meta-Learning With Landmarking: A Survey. *International Journal of Computer Applications, 105*, 47-51.

Baxter, J. (2000). A model of inductive bias learning. *J. Artif. Intell. Res.(JAIR), 12*, 149-198.

Behja, H., Marzak, A., & Trousse, B. (2012). Ontology-Based Knowledge Model for Multi-View KDD Process. *lnternational Journal of Mobile Computing and Multimedia Communications (IJMCMC], 4*(3), 21-33.

Berrer, H., Paterson, I., & Keller, J. r. (2000). Evaluation of machine-learning algorithm ranking advisors. *In Proceedings of the PKDD-2000 Workshop on DataMining, Decision Support, Meta-Learning and ILP: Forum for Practical Problem Presentation and Prospective Solutions*, 1-13.

Bhargava, N., Sharma, G., Bhargava, R., & Mathuria, M. (2013). Decision tree analysis on j48 algorithm for data mining. *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering, 3*(6).

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence, 97*(1), 245-271.

Bond, T. G., & Fox, C. M. (2013). *Applying the Rasch model: Fundamental measurement in the human sciences*: Psychology Press.

Brazdil, P., Christophe, G.-C., Carlos, S., & Ricardo, V. (2008). *Metalearning: Applications to Data Mining*: Springer Publishing Company, Incorporated.

Brazdil, P., Soares, C., & Da Costa, J. P. (2003). Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Machine Learning, 50*(3), 251-277.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research, 16*(1), 321-357.

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine learning, 15*(2), 201-221.

Cohn, D., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of artificial intelligence research*, 129-145.

Dasgupta, S. (2011). Two faces of active learning. *Theoretical computer science, 412*(19), 1767-1781.

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent data analysis, 1*(3), 131-156.

de Miranda, P. B. C., Prudancio, R. B. C., de Carvalho, A. C. P. L. F., & Soares, C. (2012). An experimental study of the combination of meta-learning with particle swarm algorithms for svm parameter selection. In *Computational Science and Its Applicationsâ€“ICCSA 2012* (pp. 562-575): Springer.

Devasena, C. L., Sumathi, T., Gomathi, V. V., & Hemalatha, M. (2011). Effectiveness evaluation of rule based classifiers for the classification of iris data set. *Bonfring International Journal of Man Machine Interface, 1*(Special Issue Inaugural Special Issue), 05-09.

Dietterich, T. (1995). Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR), 27*(3), 326-327.

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

Drummond, C., & Holte, R. C. (2005). Severe class imbalance: Why better algorithms arenâ€™t the answer. In *Machine Learning: ECML 2005* (pp. 539-546): Springer.

Elkan, C. (2001). The foundations of cost-sensitive learning. *International joint conference on artificial intelligence, 17*, 973-978.

Fan, W., Stolfo, S. J., Zhang, J., & Chan, P. K. (1999). AdaCost: misclassification cost-sensitive boosting. *ICML*, 97-105.

Fayyad, U. M., & Irani, K. B. (1992). On the handling of continuous-valued attributes in decision tree generation. *Machine learning, 8*(1), 87-102.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research, 3*, 1289-1305.

Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1993). Information, prediction, and query by committee. *Advances in neural information processing systems*, 483-490.

Fu, Y., Zhu, X., & Li, B. (2013). A survey on instance selection for active learning. *Knowledge and information systems, 35*(2), 249-283.

Furnkranz, J., & Petrak, J. (2001). An evaluation of landmarking variants. *Working Notes of the ECML/PKDD 2000 Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning*, 57-68.

Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering, 2*(4), 42-47.

Giraud-Carrier, C. (2008). Metalearning-a tutorial. *Proceedings of the 7th international conference on machine learning and applications*, 1-45.

Gomes, T. A. F., Prudancio, R., Soares, C., Rossi, A. L. D., & Carvalho, A. (2012). Combining meta-learning and search techniques to select parameters for support vector machines. *Neurocomputing, 75*(1), 3-13.

Guo, X., Yin, Y., Dong, C., Yang, G., & Zhou, G. (2008). On the class imbalance problem. *Natural Computation, 2008. ICNC'08. Fourth International Conference on, 4*, 192-201.

Hall, M. (1999a). *Correlation-based feature selection for machine learning.* Unpublished PhD, The University of Waikato.

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

Hall, M. A. (1999b). *Correlation-based feature selection for machine learning.* The University of Waikato.

Hall, M. A., & Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. *Knowledge and Data Engineering, IEEE Transactions on, 15*(6), 1437-1447.

Hecht-Nielsen, R. (1989). *Theory of the backpropagation neural network.* Paper presented at the Neural Networks, 1989. IJCNN., International Joint Conference on.

Hilario, M., Nguyen, P., Do, H., Woznica, A., & Kalousis, A. (2009). Ontology-based meta-mining of knowledge discovery workflows. In *Meta-learning in computational intelligence* (pp. 273-315): Springer.

Hilario, M., Nguyen, P., Do, H., Woznica, A., & Kalousis, A. (2011). Ontology-based meta-mining of knowledge discovery workflows. In *Meta-learning in computational intelligence* (pp. 273-315): Springer.

Holub, A., Perona, P., & Burl, M. C. (2008). Entropy-based active learning for object recognition. *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, 1-8.

Huang, S.-J., Jin, R., & Zhou, Z.-H. (2010). Active learning by querying informative and representative examples. *Advances in neural information processing systems*, 892-900.

Hutter, F., & Hamadi, Y. (2005). Parameter adjustment based on performance prediction: Towards an instance-aware problem solver. *In: Technical Report: MSR-TR-2005125, Microsoft Research*, 1-59.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis, 6*(5), 429-449.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. *Machine Learning: Proceedings of the Eleventh International Conference*, 121-129.

Jordan, A. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems, 14*, 841.

Kabakchieva, D. (2013). Predicting student performance by using data mining methods for classification. *Cybernetics and Information Technologies, 13*(1), 61-72.

Kalousis, A., & Hilario, M. (2001). *Feature selection for meta-learning*: Springer.

Kalousis, A., & Theoharis, T. (1999). Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection. *Intelligent Data Analysis, 3*(5), 319-337.

Kane, E. (1983). *Doing your own research: Basic descriptive research in the social sciences and humanities*: Marion Boyars.

Kearns, M. J., & Valiant, L. G. (1988). *Learning Boolean formulae or finite automata is as hard as factoring*: Harvard University, Center for Research in Computing Technology, Aiken Computation Laboratory.

Keet, M., awrynowicz, A., daeamato, C., & Hilario, M. (2013). Modeling issues & choices in the data mining optimization ontology.

Kim, Y., Street, W. N., & Menczer, F. (2003). Feature selection in data mining. *Data mining: opportunities and challenges, 3*(9), 80-105.

King, D., Feng, C., & Sutherland, A. (1995). Statlog: comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence an International Journal, 9*(3), 289-333.

Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. *Proceedings of the ninth international workshop on Machine learning*, 249-256.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence, 97*(1), 273-324.

Koller, D., & Sahami, M. (1996a). Toward optimal feature selection. *Proceedings of the Thirteenth International Conference on Machine Learning*, 284–292.

Koller, D., & Sahami, M. (1996b). Toward optimal feature selection.

Kononenko, I. (1994). Estimating attributes: analysis and extensions of RELIEF. *Machine Learning: ECML-94*, 171-182.

Kothari, C. R. (2004). *Research methodology: methods and techniques*: New Age International.

Kothari, C. R. (2011). *Research methodology: methods and techniques*: New Age International.

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering, 30*(1), 25-36.

Ladha, L., & Deepa, T. (2011). Feature Selection Methods and Algorithims *International Journal on Computer Science & Engineering, 3*(5), 129-134.

Lakatos, I. (1980). *The methodology of scientific research programmes: Volume 1: Philosophical papers* (Vol. 1): Cambridge university press.

Lemke, C., Budka, M., & Gabrys, B. (2013). Metalearning: a survey of trends and technologies. *Artificial Intelligence Review*, 1-14.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In *Machine learning: ECML-98* (pp. 4-15): Springer.

Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, 3-12.

Lindenbaum, M., Markovitch, S., & Rusakov, D. (2004). Selective sampling for nearest neighbor classifiers. *Machine learning, 54*(2), 125-152.

Liu, H., & Motoda, H. (1998). *Feature selection for knowledge discovery and data mining*: Springer Science & Business Media.

Liu, Y., & Schumann, M. (2005). Data mining feature selection for credit scoring models. *Journal of the Operational Research Society, 56*(9), 1099-1108.

Lomax, S., & Vadera, S. (2013). A survey of cost-sensitive decision tree induction algorithms. *ACM Computing Surveys (CSUR), 45*(2), 16.

Mamitsuka, N. A. H. (1998). Query learning strategies using boosting and bagging. *Machine Learning: Proceedings of the Fifteenth International Conference (ICML'98)*, 1.

McCarthy, K., Zabar, B., & Weiss, G. (2005a). *Does cost-sensitive learning beat sampling for classifying rare classes?* Paper presented at the Proceedings of the 1st international workshop on Utility-based data mining.

McCarthy, K., Zabar, B., & Weiss, G. (2005b). Does cost-sensitive learning beat sampling for classifying rare classes? *Proceedings of the 1st international workshop on Utility-based data mining*, 69-77.

Mease, D., Wyner, A. J., & Buja, A. (2007). Boosted classification trees and class probability/quantile estimation. *The Journal of Machine Learning Research, 8*, 409-439.

Melo , H., Hannois , G., Rodrigues , A., & Natal , J. (2011). Active learning on the ward: outcomes from a comparative trial with traditional methods. *Medical education, 45*(3), 273-279.

Melville, P., & Mooney, R. J. (2004). Diverse ensembles for active learning. *Proceedings of the twenty-first international conference on Machine learning*, 74.

Molina, L. C., Belanche, L., & Nebot, Ã. n. (2002). Feature selection algorithms: A survey and experimental evaluation. *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, 306-313.

Nakhaeizadeh, G., & Schnabl, A. (1997). Development of Multi-Criteria Metrics for Evaluation of Data Mining Algorithms. *KDD*, 37-42.

Nasa, C. (2012). Evaluation of different classification techniques for web data. *International Journal of Computer Applications, 52*(9).

Nguyen, H. T., & Smeulders, A. (2004). Active learning using pre-clustering. *Proceedings of the twenty-first international conference on Machine learning*, 79.

Norton, S. W. (1989). Generating Better Decision Trees. *IJCAI, 89*, 800-805.

Núnez, M. (1991). The use of background knowledge in decision tree induction. *Machine learning, 6*(3), 231-250.

Pajkossy, K. (2013). *Studying feature selection methods applied to classification tasks in natural language processing.* Unpublished PhD thesis, Eatvas Lorand University.

Patil, T. R., & Sherekar, S. S. (2013). Performance analysis of Naive Bayes and J48 classification algorithm for data classification. *International Journal of Computer Science and Applications, 6*(2), 256-261.

Patton, M. Q. (1980). Qualitative evaluation methods. 381.

Pfahringer, B., Bensusan, H., & Giraud-Carrier, C. (2000). Tell me who can learn you and i can tell you who you are: Landmarking various learning algorithms. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML2000)*, 743-750.

Prudancio, R., De Souto, M. C. P., & Ludermir, T. B. (2011). Selecting machine learning algorithms using the ranking meta-learning approach. In *Meta-Learning in Computational Intelligence* (pp. 225-243): Springer.

Prudancio, R., & Ludermir, T. (2007). Active selection of training examples for meta-learning. *Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on*, 126-131.

Prudancio, R., & Ludermir, T. (2008). Selective generation of training examples in active meta-learning. *Int. J. Hybrid Intell. Syst., 5*(2), 59-70.

Prudancio, R., Soares, C., & Ludermir, T. B. (2011). Combining meta-learning and active selection of datasetoids for algorithm selection. In *Hybrid Artificial Intelligent Systems* (pp. 164-171): Springer.

Qin, Z., Zhang, C., Wang, T., & Zhang, S. (2011). Cost sensitive classification in data mining. In *Advanced Data Mining and Applications* (pp. 1-11): Springer.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning, 1*(1), 81-106.

Rendell, L., & Cho, H. (1990). Empirical learning as a function of concept character. *Machine Learning, 5*(3), 267-298.

Rendell, L., & Ragavan, H. (1993). Improving the design of induction methods by analyzing algorithm functionality and data-based concept complexity. *IJCAI*, 952-959.

Rice, J. R. (1975). The algorithm selection problem. 75-152.

Roy, N., & McCallum, A. (2001). Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, 441–448.

Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine learning, 37*(3), 297-336.

Settles, B. (2011). From theories to queries: Active learning in practice. *Active Learning and Experimental Design W*, 1-18.

Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. *Proceedings of the fifth annual workshop on Computational learning theory*, 287-294.

Silverman, D. (2013). *Doing qualitative research: A practical handbook*: SAGE Publications Limited.

Skalak, D. B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. *Proceedings of the eleventh international conference on machine learning*, 293-301.

Sun, Q. (2014). *Meta-Learning and the Full Model Selection Problem*. Unpublished PhD thesis, University of Waikato.

Sun, Y., Kamel, M. S., Wong, A. K. C., & Wang, Y. (2007). Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition, 40*(12), 3358-3378.

Sun, Y., Wong, A. K. C., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence, 23*(04), 687-719.

Tan, M., & Schlimmer, J. (1989). Cost-sensitive concept learning of sensor use in approach and recognition. *Proceedings of the sixth international workshop on Machine learning*, 392-395.

Ting. (1998). Inducing cost-sensitive trees via instance weighting. In *Principles of Data Mining and Knowledge Discovery* (Vol. 1510, pp. 139-147): Springer Berlin Heidelberg.

Todorovski, L., Brazdil, P., & Soares, C. (2000). Report on the experiments with feature selection in meta-level learning. *Proceedings of the PKDD-00 workshop on data mining, decision support, meta-learning and ILP: forum for practical problem presentation and prospective solutions*, 27-39.

Tong, S., & Koller, D. (2002). Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research, 2*, 45-66.

Turney, P. (1995). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of artificial intelligence research*, 369-409.

Turney, P. (2000). Types of cost in inductive concept learning. *ICML-2000 Workshop on Cost-Sensitive Learning*, 15-21.

Urner, R., Wulff, S., & Ben-David, S. (2013). Plal: Cluster-based active learning. *Conference on Learning Theory*, 376-397.

Vadera, S. (2010). Inducing cost-sensitive non-linear decision trees. 1-22.

Vanschoren, J. (2010). *Understanding machine learning performance with experiment databases.* Unpublished Ph.D, Katholieke Universiteit Leuven.

Vilalta, R., & Drissi, Y. (2002). A perspective view and survey of meta-learning. *Artificial Intelligence Review, 18*(2), 77-95.

Wang, G., Yu, H., & Yang, D. C. (2002). Decision table reduction based on conditional information entropy. *CHINESE JOURNAL OF COMPUTERS-CHINESE EDITION-, 25*(7), 759-766.

Wang, T. (2013). *Efficient techniques for cost-sensitive learning with multiple cost considerations.* Unpublished PhD thesis, University of Technology, Sydney.

Weiss, G. M. (2004). Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter, 6*(1), 7-19.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on, 1*(1), 67-82.

Xu, Z., Akella, R., & Zhang, Y. (2007). *Incorporating diversity and density in active learning for relevance feedback*: Springer.

Xu, Z., Yu, K., Tresp, V., Xu, X., & Wang, J. (2003). *Representative sampling for text classification using support vector machines*: Springer.

Yang, J., & Honavar, V. (1998). Feature subset selection using a genetic algorithm. In *Feature extraction, construction and selection* (pp. 117-136): Springer.

Yeh, I. C., King-Jang, Y., & Tao-Ming, T. (2009). Knowledge discovery on RFM model using Bernoulli sequence. *Expert Syst. Appl., 36*(3), 5866-5871.

Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. *ICML, 3*, 856-863.

Zadrozny, B., Langford, J., & Abe, N. (2003). Cost-sensitive learning by cost-proportionate example weighting. *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, 435-442.

Zhu, J., Wang, H., Yao, T., & Tsou, B. K. (2008a). Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, 1137-1144.

Zhu, J., Wang, H., Yao, T., & Tsou, B. K. (2008b). *Active learning with sampling by uncertainty and density for word sense disambiguation and text classification.* Paper presented at the Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1.

Zhu, X., Lafferty, J., & Ghahramani, Z. (2003). Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, 58-65.

Appendix A Cost-Sensitive Meta-knowledge Decision Trees

This appendix presents the decision trees created as part of cost sensitive meta-knowledge development for both cost sensitive recommender and feature selection recommender

A1. Feature Selection Meta-knowledge Decision Tree

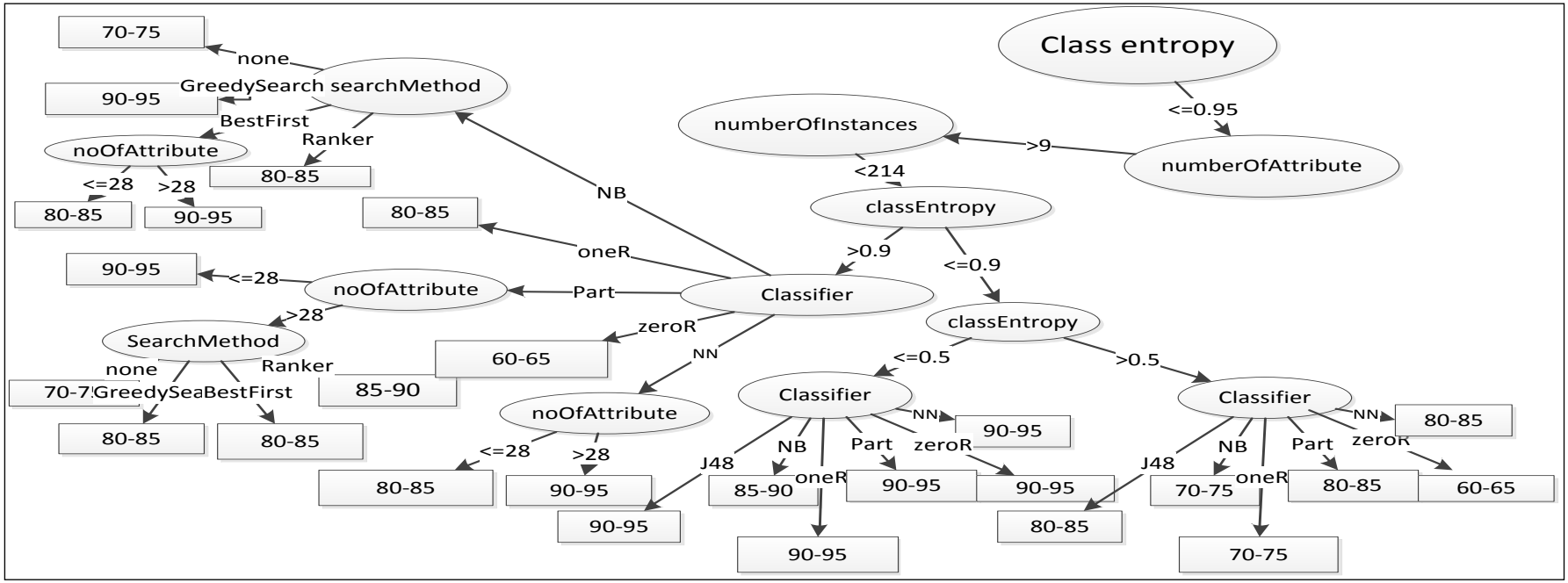

Figure A2-1-1: Decision Tree for Feature Selection with Accuracy

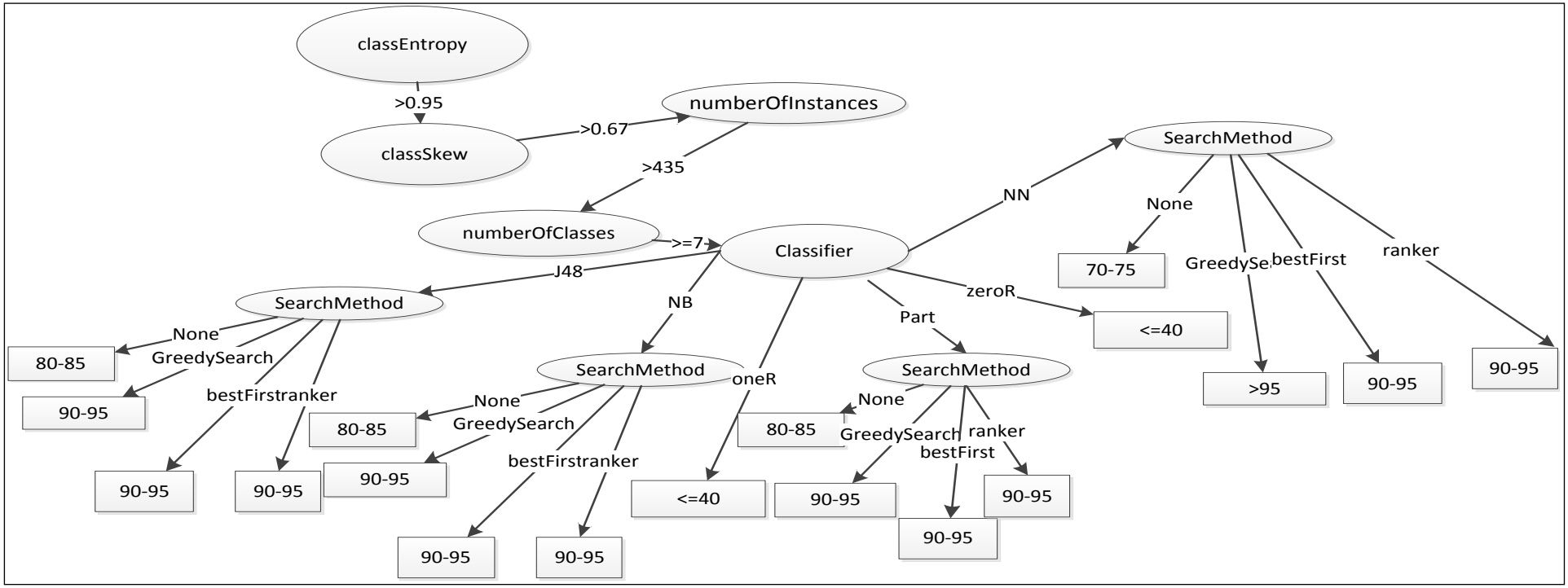

Figure A1-2: Decision Tree for Feature Selection with Accuracy

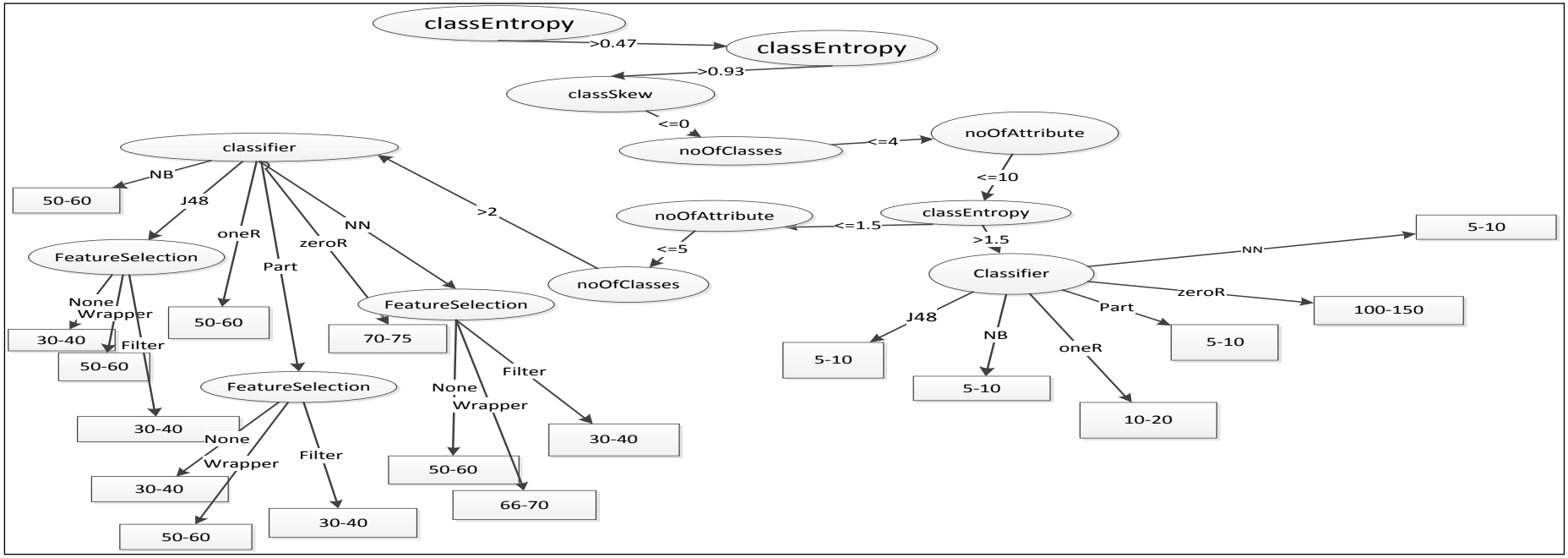

Figure A1-3: Decision Tree for Feature Selection with Misclassification Cost

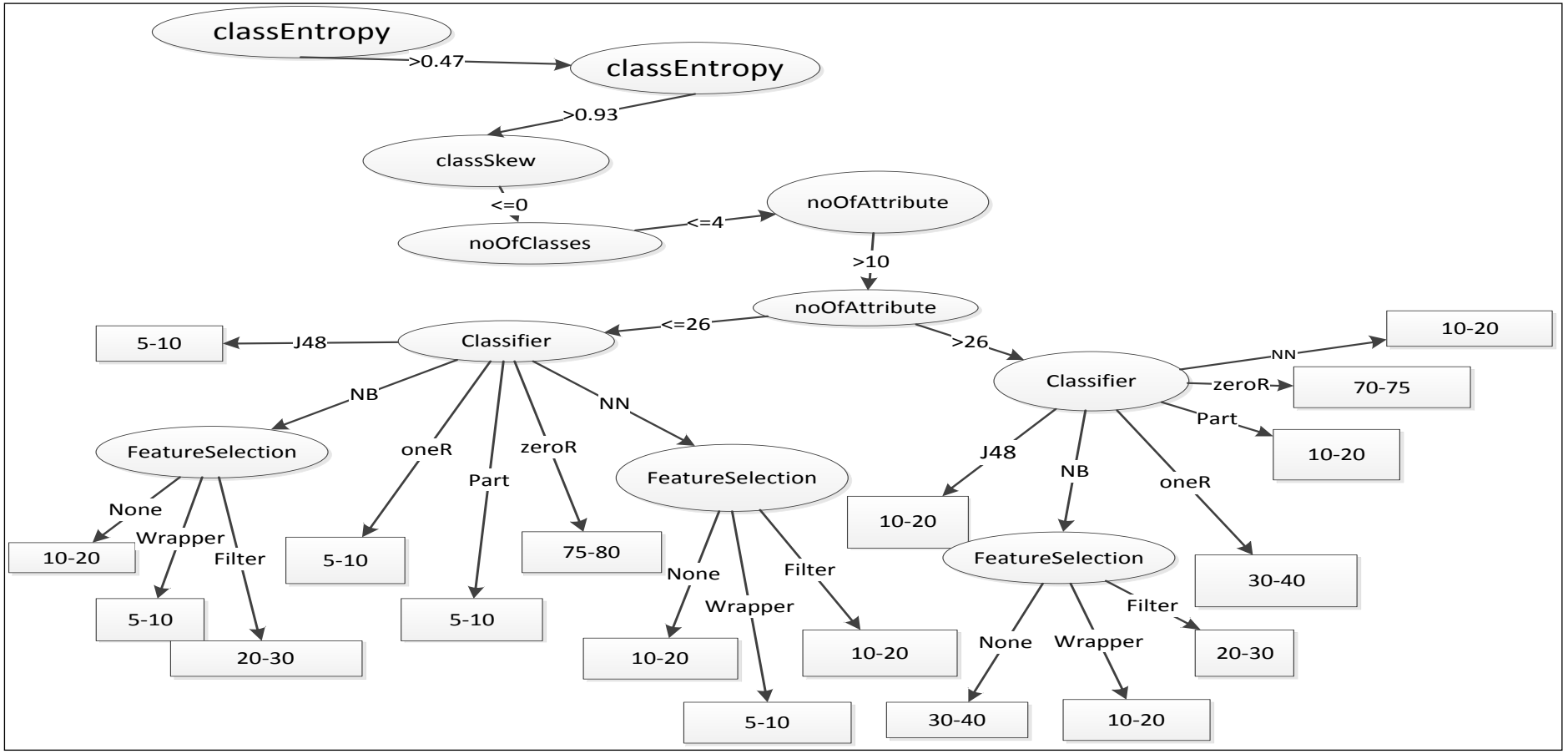

Figure A1-4: Decision Tree for Feature Selection with Misclassification Cost

A2. Cost-Sensitive Meta-knowledge Decision Trees

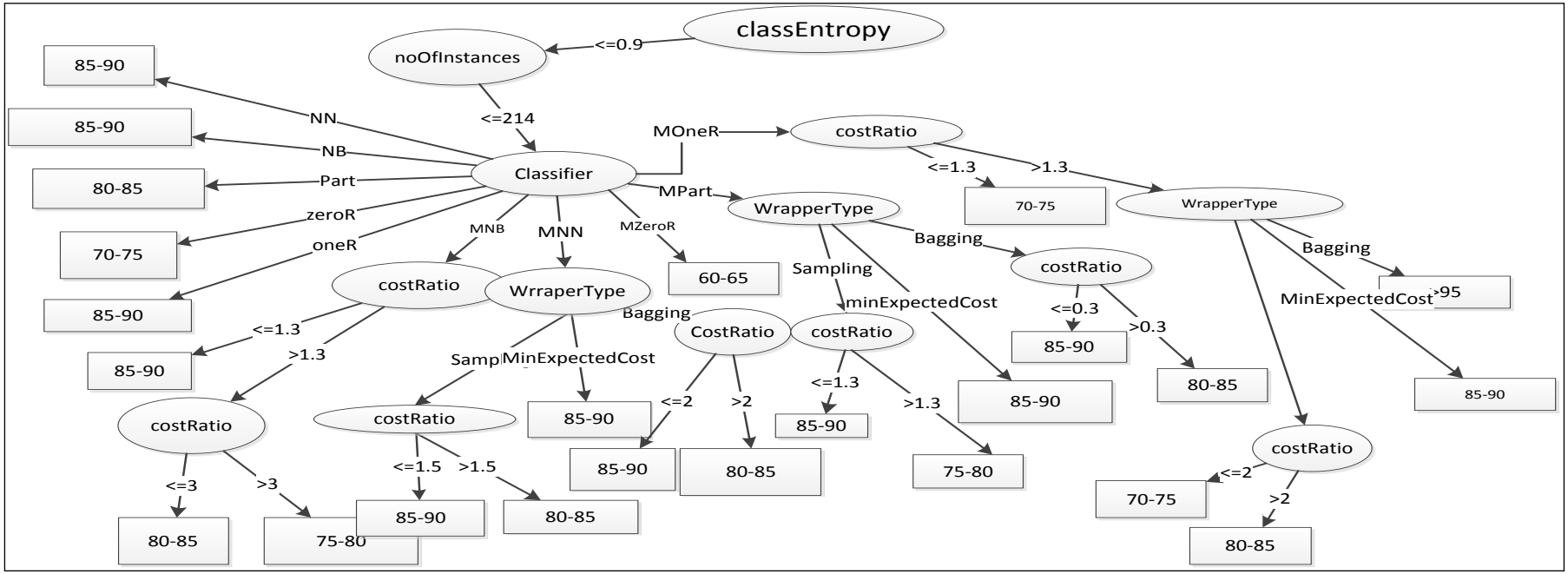

Figure A2-1: Decision Tree for Cost Sensitive Methods with Accuracy

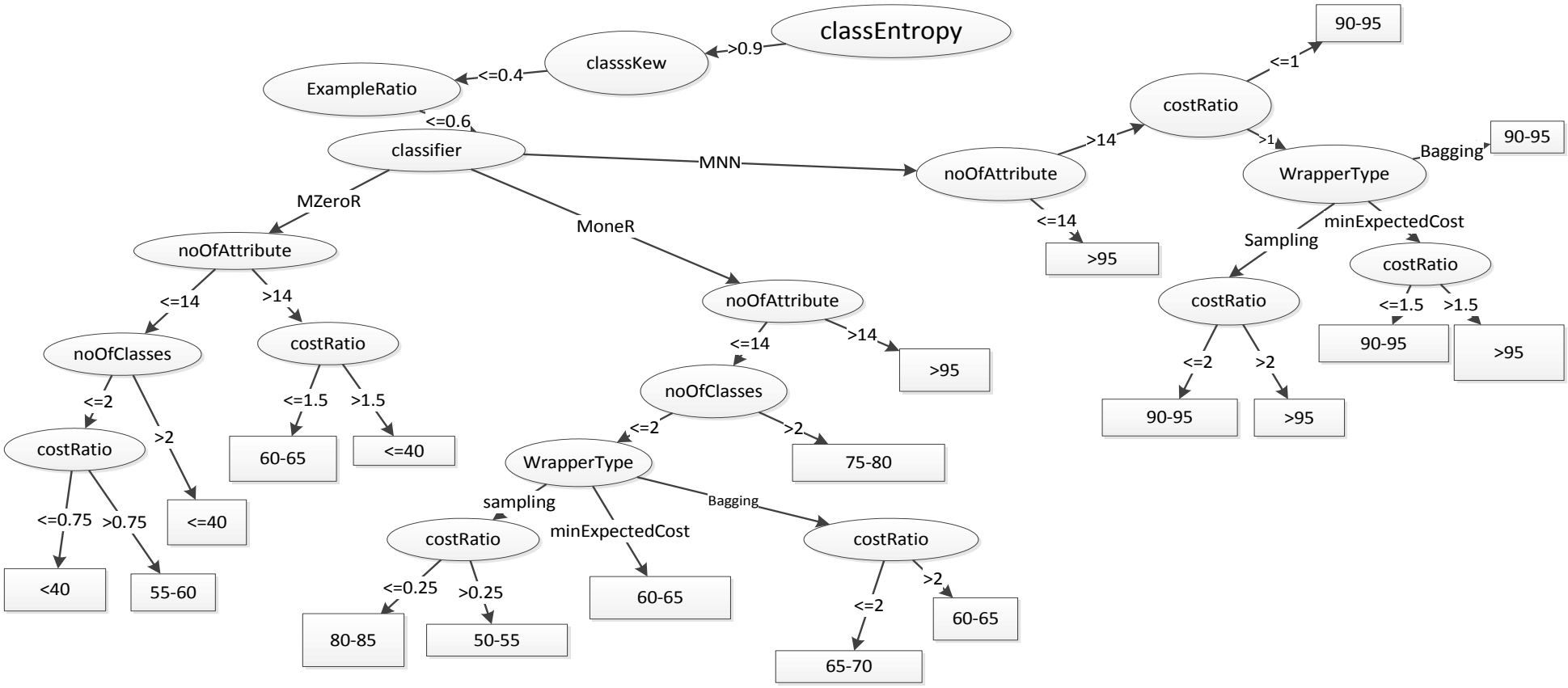

Figure A2-2: Decision Tree for Cost Sensitive Methods with Accuracy

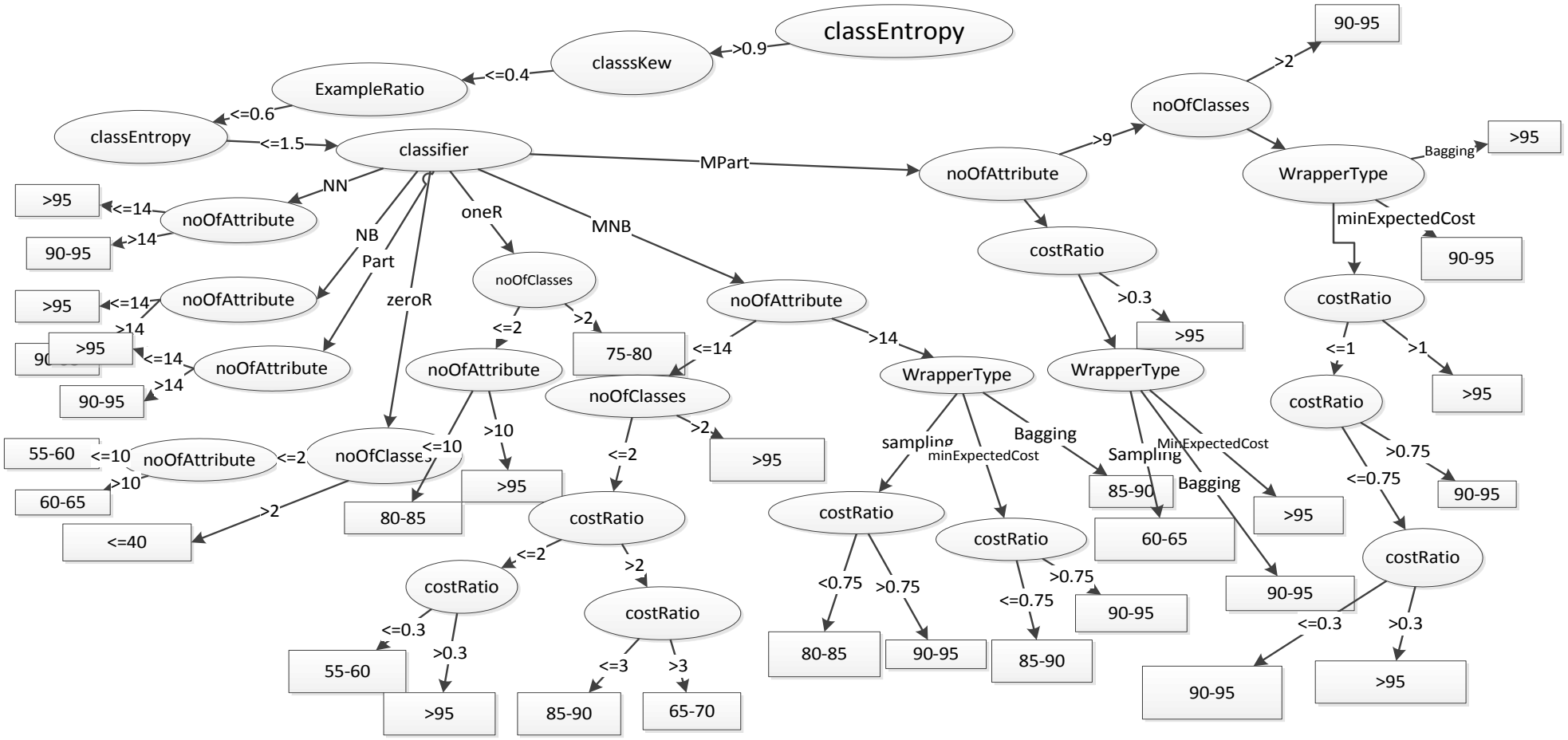

Figure A2-3: Decision Tree for Cost Sensitive Methods with Accuracy

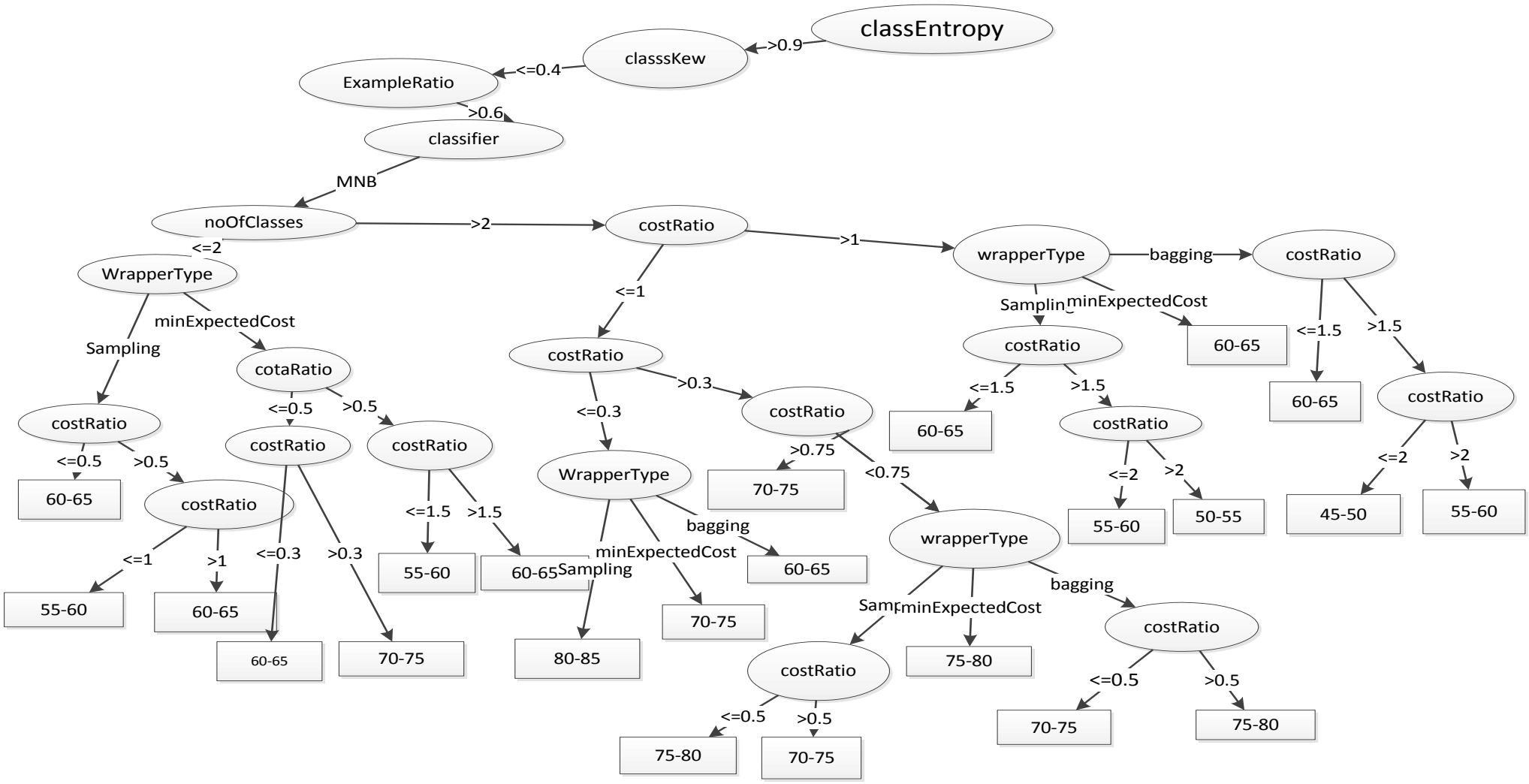

Figure A2-4: Decision Tree for Cost Sensitive Methods with Accuracy

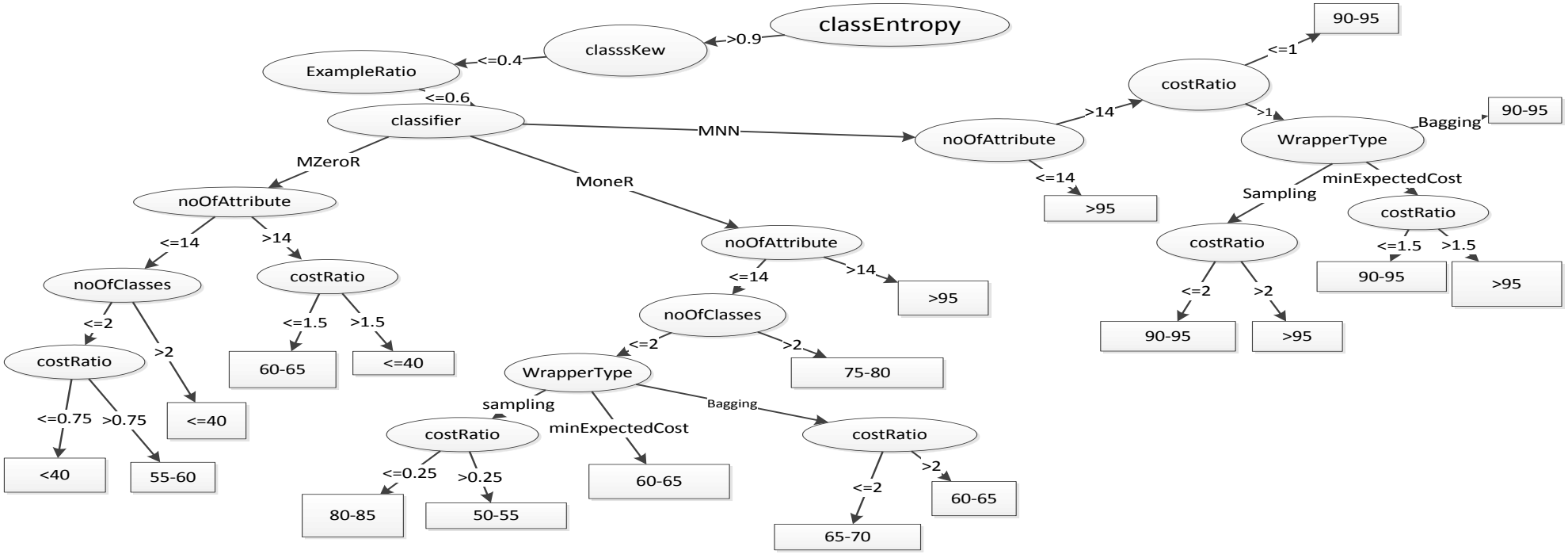

Figure A2-5: Decision Tree for Cost Sensitive Methods with Accuracy

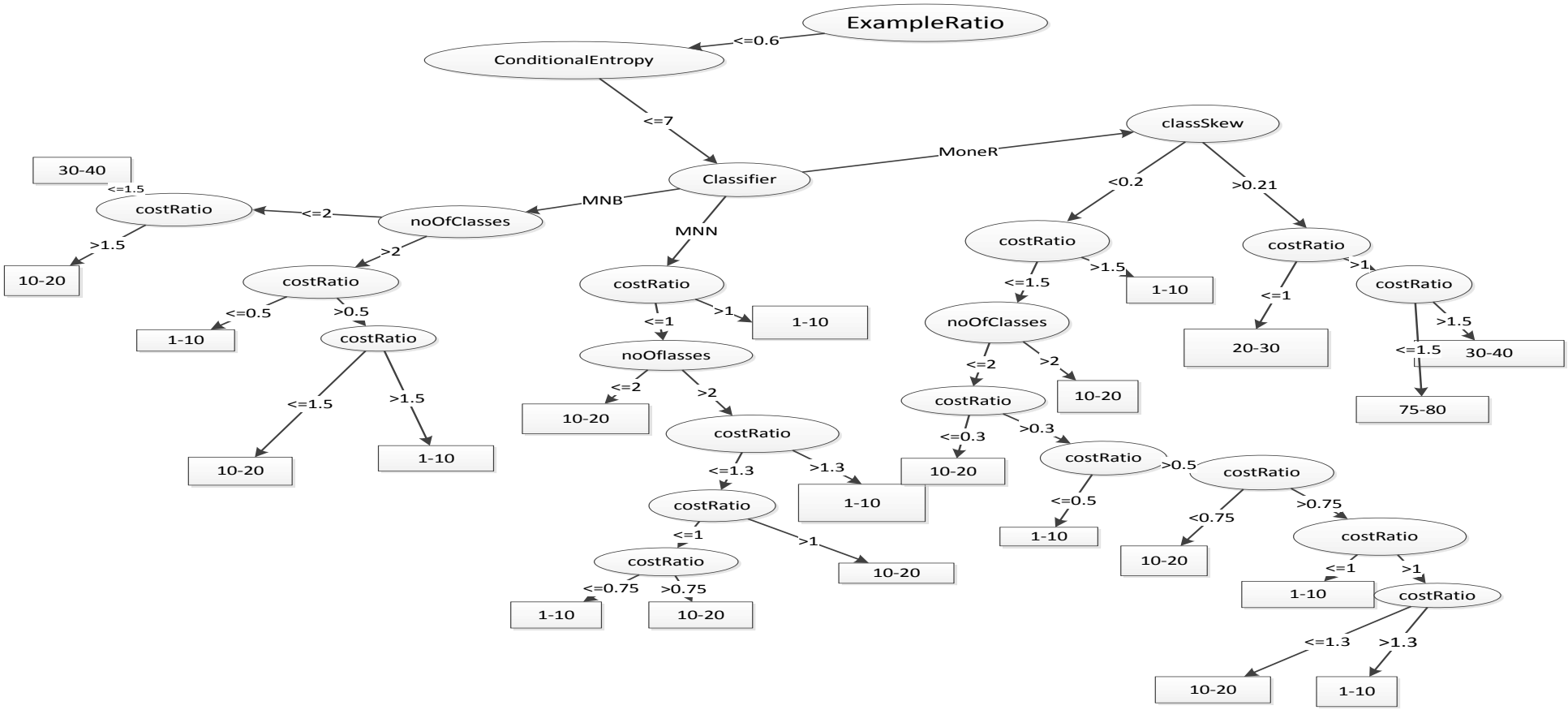

Figure A2-6: Decision Tree for Cost Sensitive Methods with Misclassification Cost

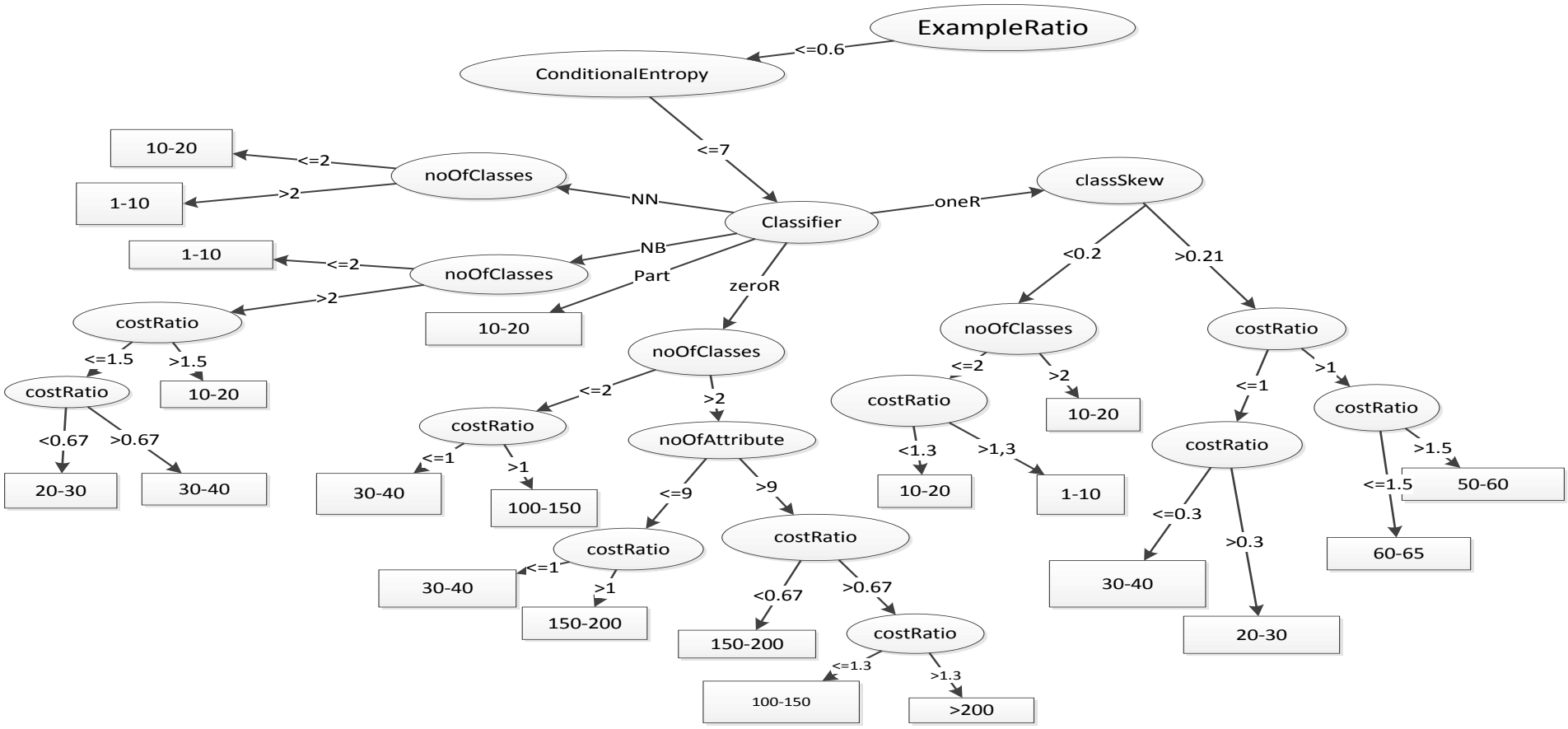

Figure A2-7: Decision Tree for Cost Sensitive Methods with Misclassification Cost

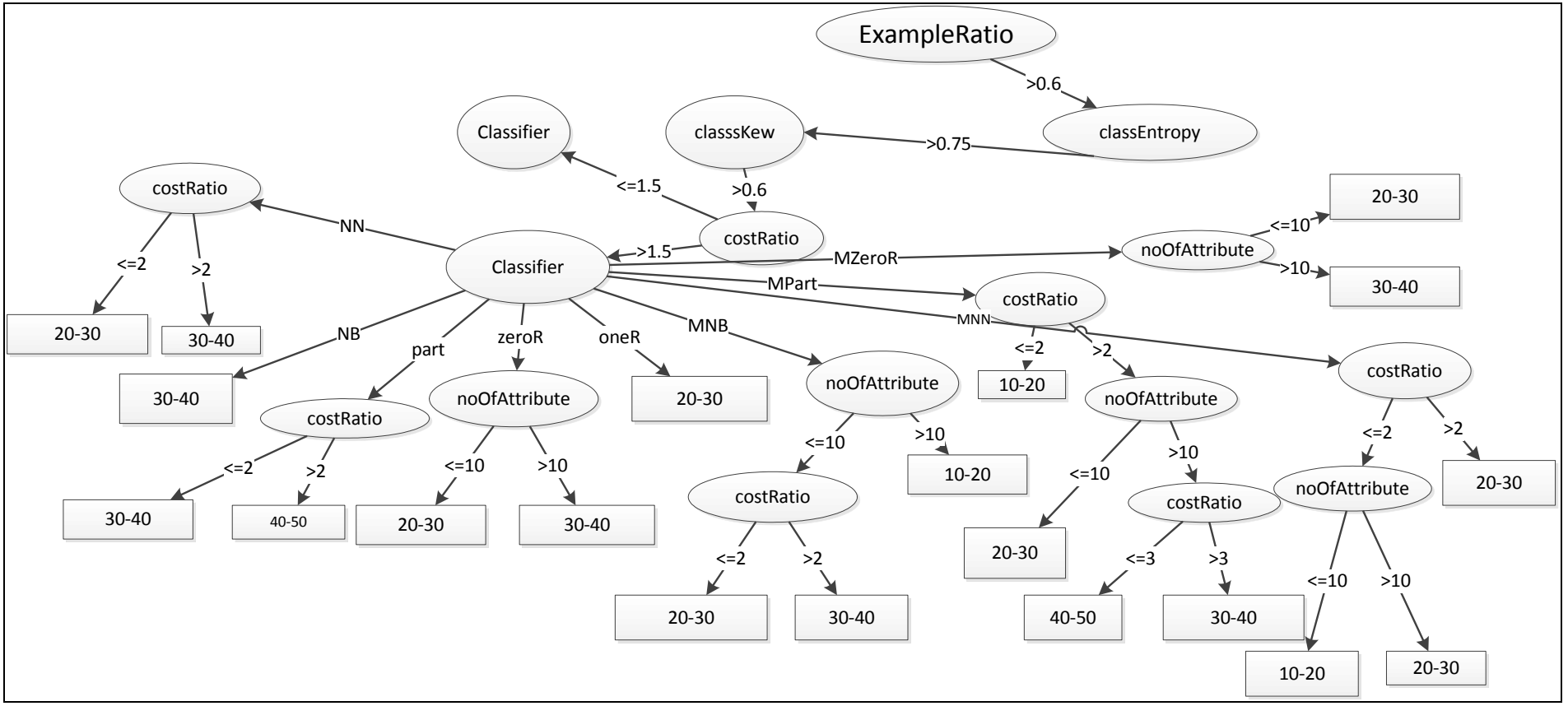

Figure A2-8: Decision Tree for Cost Sensitive Methods with Misclassification Cost

Appendix B Meta-Learning Cost-Senstive and Active Learning design

This appendix presents the details design for both meta-learning system and active learning based on clustering approach, the design includes class diagrams which includes the collection of classes that contribute in the system, their relations and each system scenario that shows how each part of the system is developed and used.

B1. Cost- Sensitive Meta-knowledge System Design

The first step in structuring the system is to model the system domain in terms of a collection of classes and the relationships between them. These classes are conceptual classes, which includes modeling the main elements that contributes in building the system and the relation between them, the relation between those elements represent connection that are important in building the system, the aim of this section is to develop class diagrams that show the main elements that lead to full system development, this section is divided into two parts:

- Feature selection meta-knowledge that recommends feature selection method.
- Cost sensitive meta-knowledge development which recommends cost sensitive method.

- **Feature Selection Meta-Knowledge Development**

This part of the system takes a data set and recommends feature selection method with the learner that will produce a best choice highest accuracy and minimum cost.

Following is the main classes that contribute in reaching this aim with each class attributes and associations, class diagram of Feature selection recommender shown in Figure B1-1.

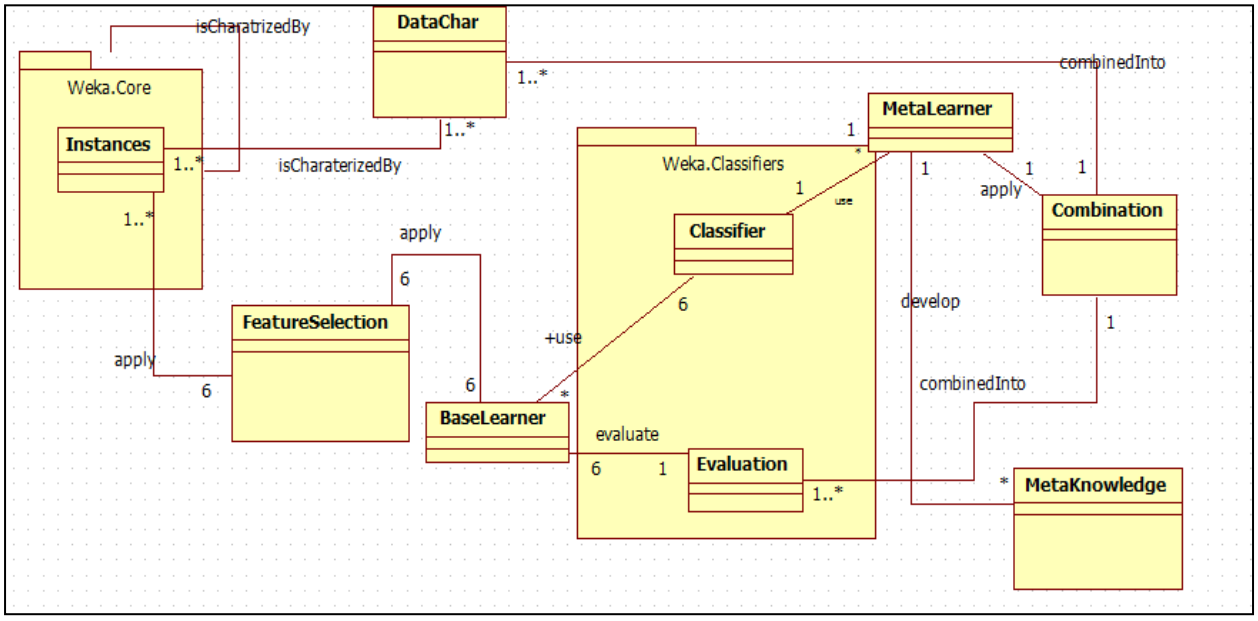

Figure B1-1: Feature Selection Meta-Knowledge Development Class Diagram

Instances class

Instances refer to the input, or the example that will be learned , each dataset is charaterized by the values of a set of predetermenined attributes, An ***Instances*** class is dealing with instances that is in our case a dataset that will be examined, this class is available in ***Weka.Core*** package and connects with ***Instance*** class where number of ***Instance*** object will create ***Instances*** object, and it connects with ***Attribute*** class that records each instance characteristics as shown in Fig B1-2 .

DataChar class

This class is responsible for developing different dataset characterization which includes some simple, mathematical and statistical data characters for ***Instances*** class

objects, following is a class diagram showing the classes, relations that connect each class elements in ***DataChar*** class (Figure B1-2)

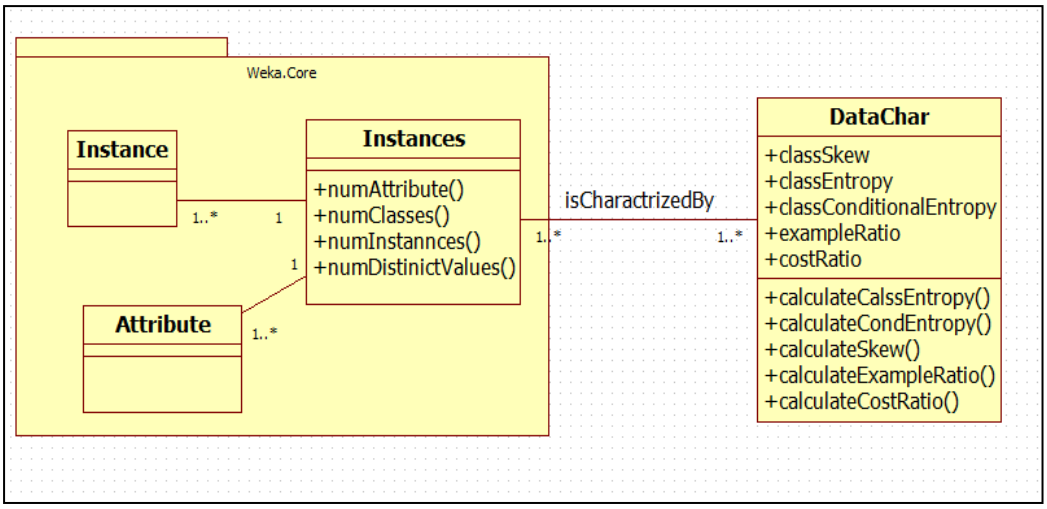

Figure B1-2: DataChar Class Diagram

Following is a detailed desription for each class contributes in this process

FeatureSelection class

This class is resposible for applying diffrent feature selection methods and search stategies that is used to remove the features that are irrelevent or reduncdent after evluating thier effects on the learning process using feature selection wrapper methods, or by finding thier correlation with each others using featue selection filter methods, following show feature selections methods which are defind in ***FeatureSelection*** class that uses ***AttributeSelection*** weka class, ***FeatureSelection*** class uses five diffrent ***AttributeSelection*** methods and four search satratigies, each ***Featureselection*** object is resposible to record a feature selection name, type from ***AttributeSelection*** class and search strategy used from ***AttributeSelectedClassifier*** class, see Figure B1-3.

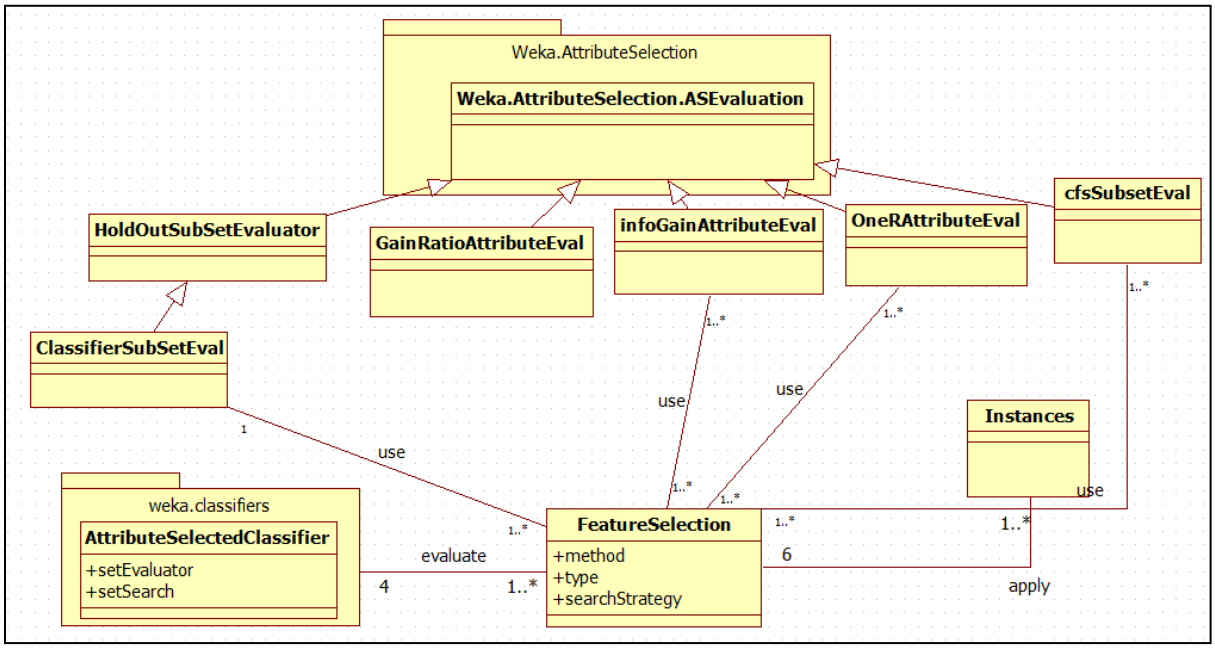

Figure B1-3: FeatureSelection Class Diagram

BaseLearner class

Baselearner class is responsible for recording information about the classifiers applied to the given dataset in base learning stage, six diffrent classifiers are applied to the given dataset using ***Classifiers*** class from ***Weka.Classifier*** package using ***buildClassifier*** method, the ***nameOfClassifier*** is recorded in ***BaseLearner*** class, then its accuracy and cost are calculated in ***BaseLearner*** class using values returned from **Evaluation** class as shown in the diagram Figure B1-4, a full detailes of every tasks and how could be achieved will be desribed below in dynamic diagrams development.

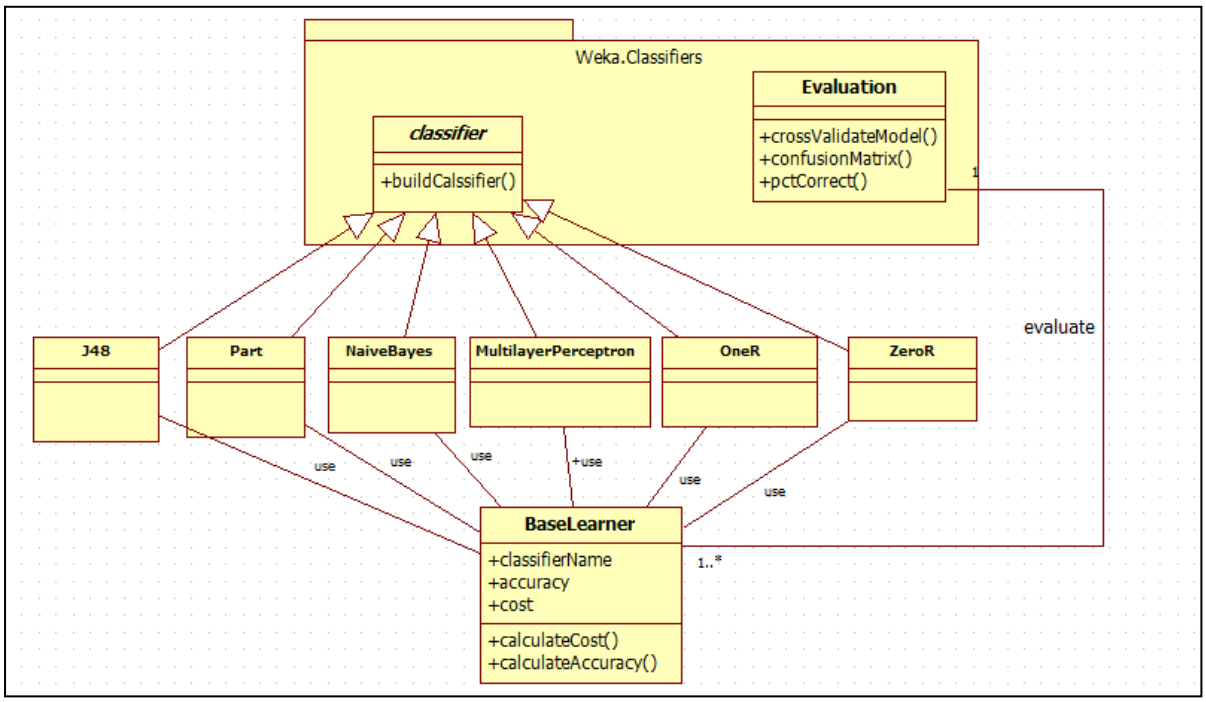

Figure B1-4: BaseLearner Class Diagram

Combination class

This class is resposible for recording all information from all previous processes including data charaters, the result of applying feature selections methods, and the result of applying diffrent base learner classifiers with thier performance, all those information are linked togother to create combination object, that will be linked to metaLearner object via apply, see Figure B1-5.

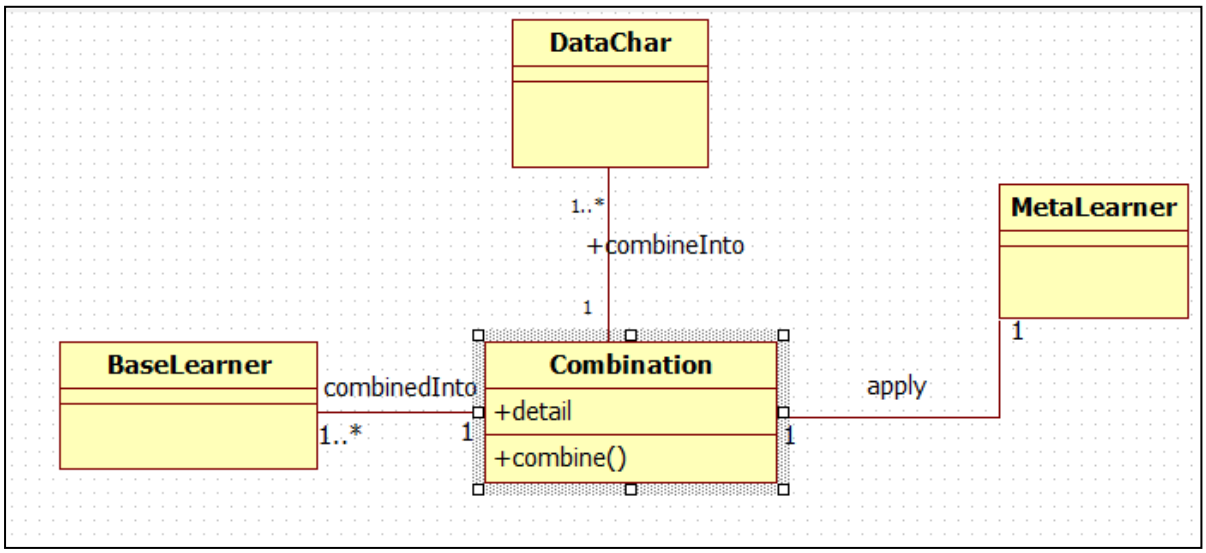

Figure B1-5 : Combination Class Diagram

MetaLearner class

Metalearner class is responsible for recoding information about the classifier applied in meta learning level, ***J48*** is used as meta learner to develop a decision tree that will be used in creating rules used in meta knowedlge development, ***MetaLearner*** class stores information about the decision tree results from applying J48 on the combination of all previous processes, the result of applying a J48 can be returned using ***result*** method and the combination can be returned using ***getCombination*** method that returns combination object from ***Combination*** class see Figure B1-6.

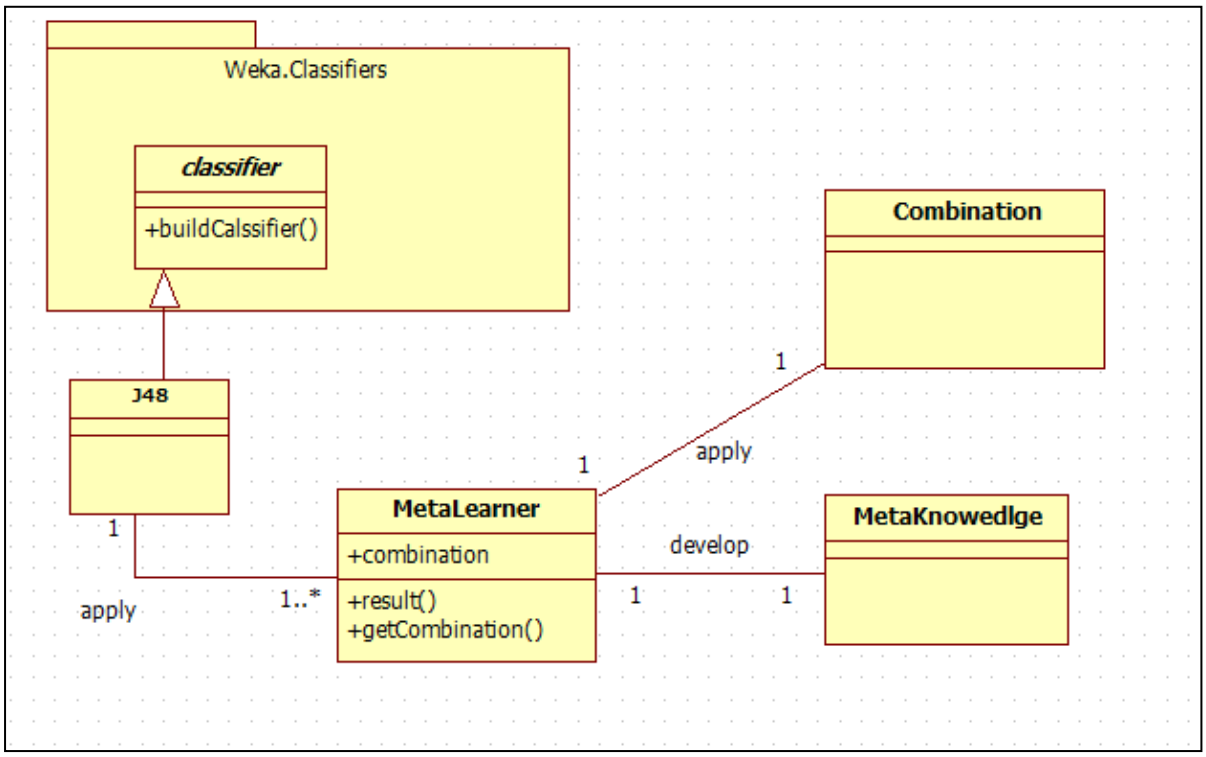

Figure B1-6: Combination Class Diagram

Metaknowledge class

This class is responsible for developing meta knowledge as rules from the desision tree result from ***MetaLearner*** class see Figure B1-7.

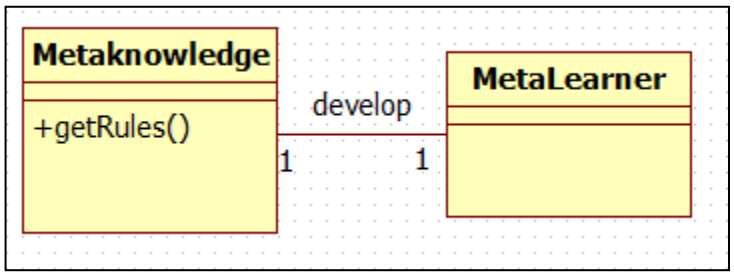

Figure B1-7: Met-knowledge Class Diagram

- **Cost sensitive Meta-Knowledge Development**

This part of the system takes a dataset and recommend a learner that will produce best cost sensitive method, this is done by using cost sensitive wrapper methods as individual phase to convert cost insensitive learner to cost sensitive learner, following is the main classes that contribute in reaching this aim with each class attributes and methods as shown in Figure B1-8.

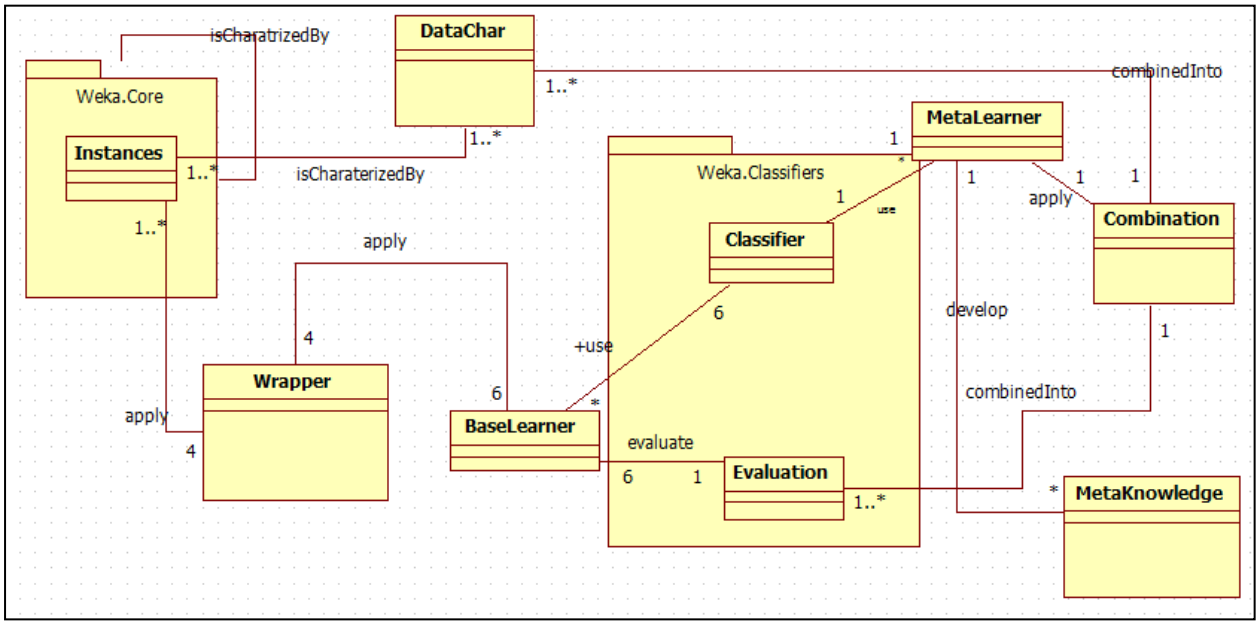

Figure B1-8: Cost Sensitive Meta-Knowledge Development

Wrapper class

This class is resposible for adding individual wrapper phase to convert cost insensitive learners to cost sensitive learner, this will be done using three diffrent wrapper approches sampling and minmum expected cost taken from weka ***costSensitiveClassifer*** class and one bagging from ***MetaCost*** class see Figure B1-9.

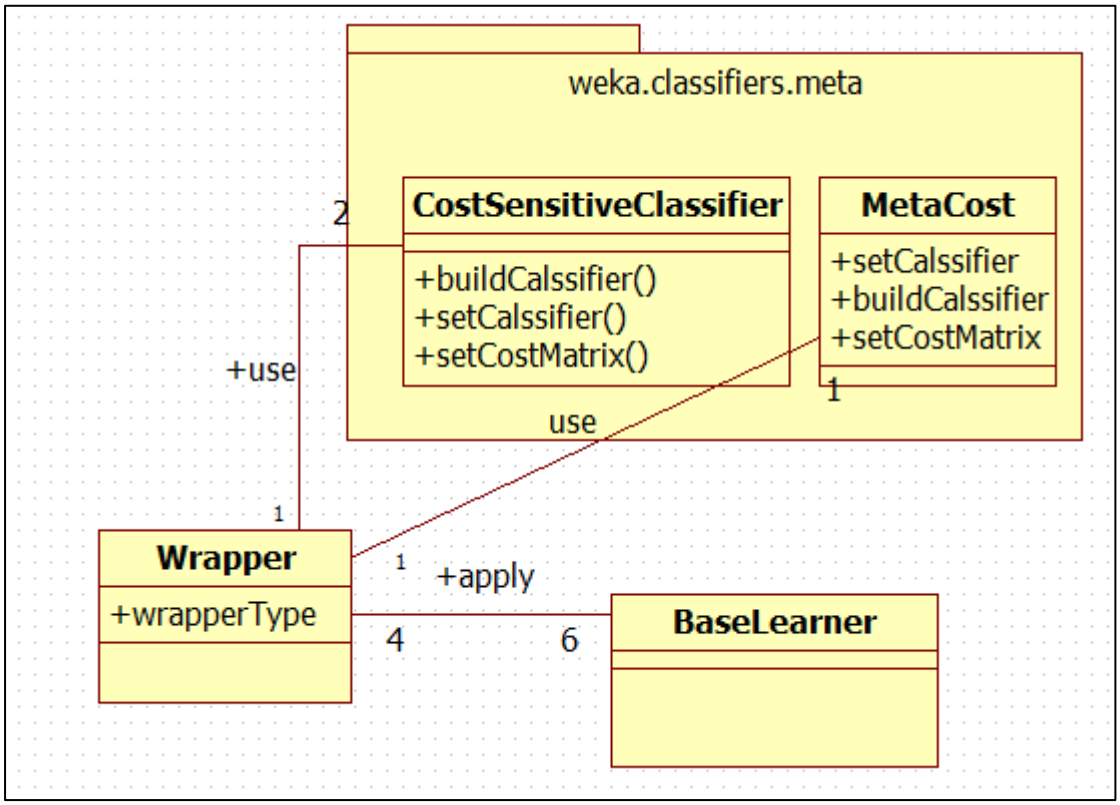

Figure B1-9: Wrapper Class Diagram

The Use Cases:

In this section, use cases that will be implemented will be shown, to decide what is required to implement the system; each task the system has to perform is investigated, breaking down what is required into separate steps within the core system. This section is divided into two sections: section one describes all steps in developing feature selection meta-knowledge development; section two describes all stapes in developing cost sensitive meta-knowledge development.

Feature Selection Meta-knowledge Development

The aim of this section is to show each individual task to develop a working scenario and how each object interact to demonstarte those working scenario.

- Feature selection meta-knowledge development scenario

In this scenario, large number of dataset are used to generate a meta-knowledge that recommend feature selection with classifier that maximize accuracy and minmize cost, each given dataset is characterized, then diffrent feature selection methods with diffrent stategies are applied on a given dataset in a base learning level, then all previous result is fed to meta learner in meta learning base to develop a set of rules that created the needed meta knowledge, this scenario with its use cases are shown in Figure B1-10.

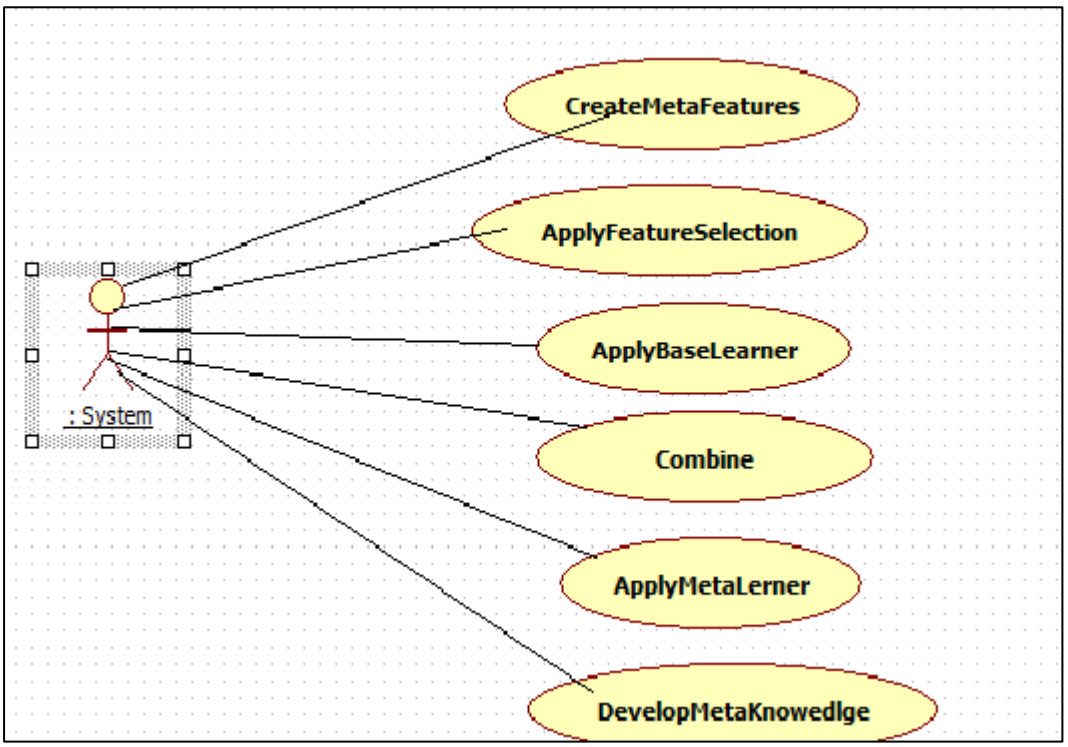

Figure B1-10: Feature Selection Meta-knowledge Development Use Case

- Predict feature selection plan and classifier with its performance for a given dataset scenario.

  The user inputs a dataset, the system generates meta-features for a given dataset, after meta-features are generated, the system predicts the performance of applying specific classifier and specific feature selection method on a given dataset. Figure B1-11 shows predict feature selection plan and performance for given dataset.

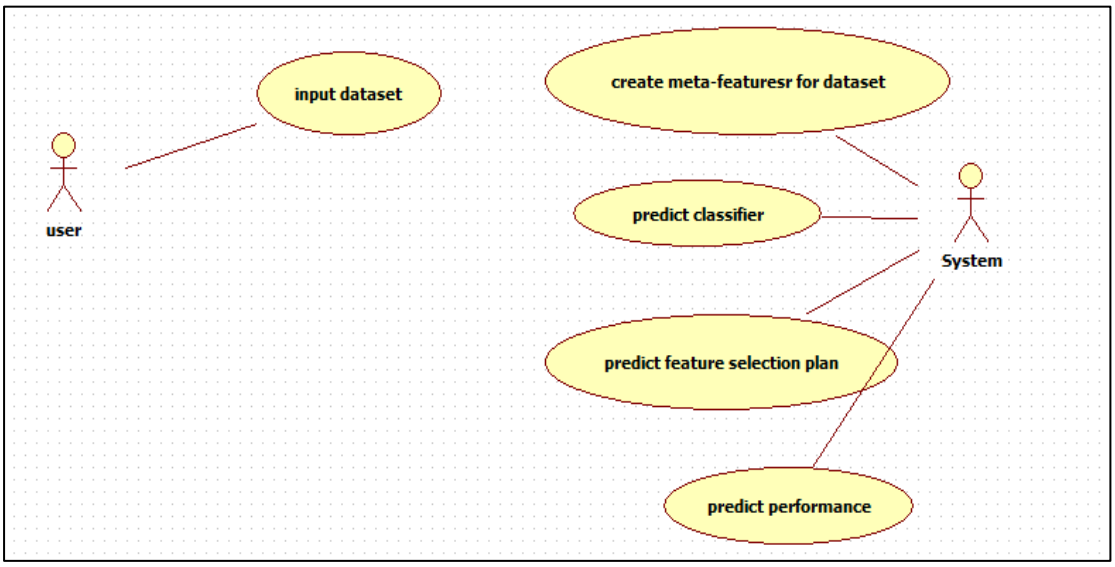

Figure B1-11: Predict feature selection with classifier with its performance for given dataset

- Predict feature selection plan and classifier performance for a given dataset and given cost file.

  The user will input a dataset, and a cost file, the system will generate meta-features for a given dataset, after meta-features are generated, the system will predict the performance of applying specific classifier on a give dataset after applying a specific feature selection method see Figure B1-12.

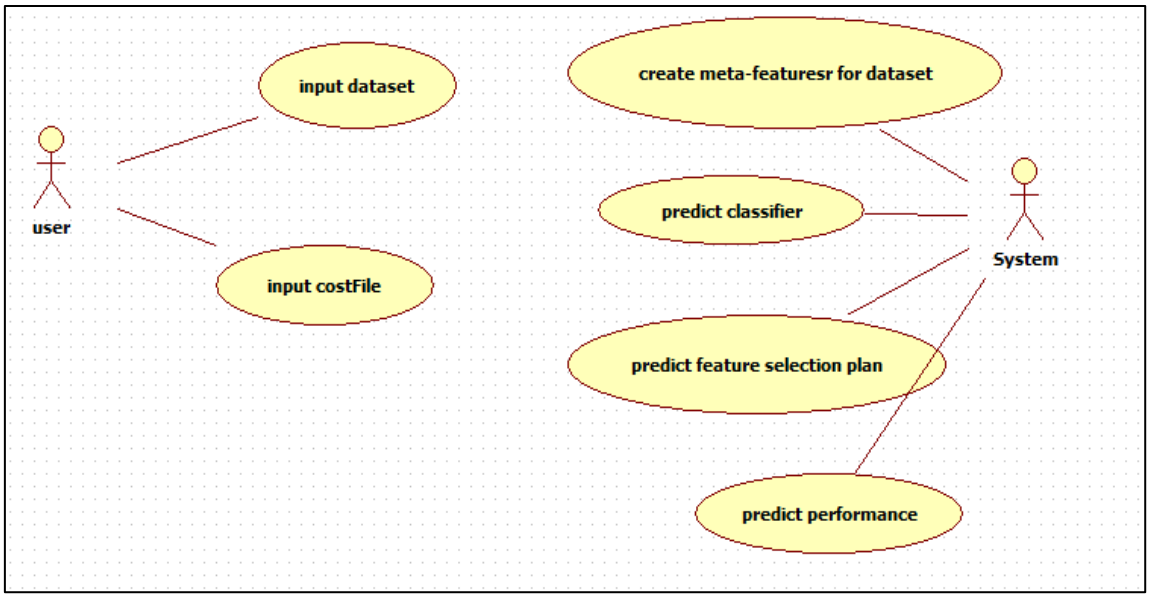

Figure B1-12: Predict feature selection and classifier with its performance for given dataset and given cost file

- Predict feature selection plan and classifier performance for a given dataset and given classifier

  The user inputs a dataset and the classifier name, the system generates metafeatures for a given dataset, after meta-features are generated, the system predicts the performance of applying given classifier on a given dataset with feature selection method.

  Figure B1-13 shows predict feature selection method and performance for given dataset and given classifier.

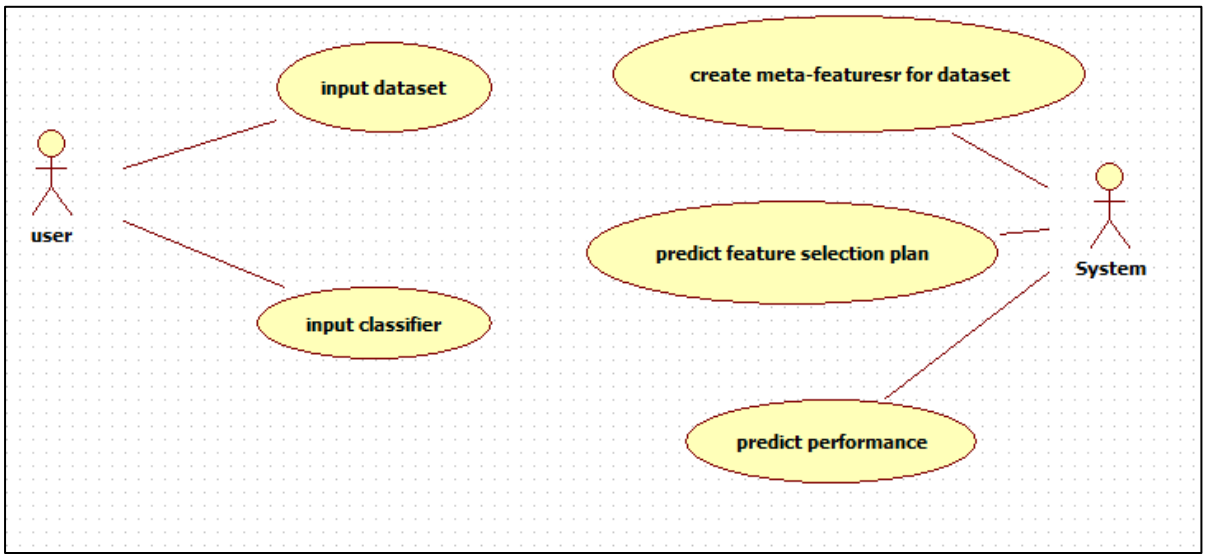

Figure B1-13: Predict feature selection and classifier performance for given dataset and given classifier

- Predict feature selection method and classifier performance for a given dataset, given classifier and given cost file.

  The user will input a dataset, chooses a classifier name, and input a cost file, the system will generate meta-features for a given dataset, after meta-features are generated, the system will predict the performance of applying specific classifier on a give dataset after applying a specific feature selection plan for a given cost file, see Figure B1-14.

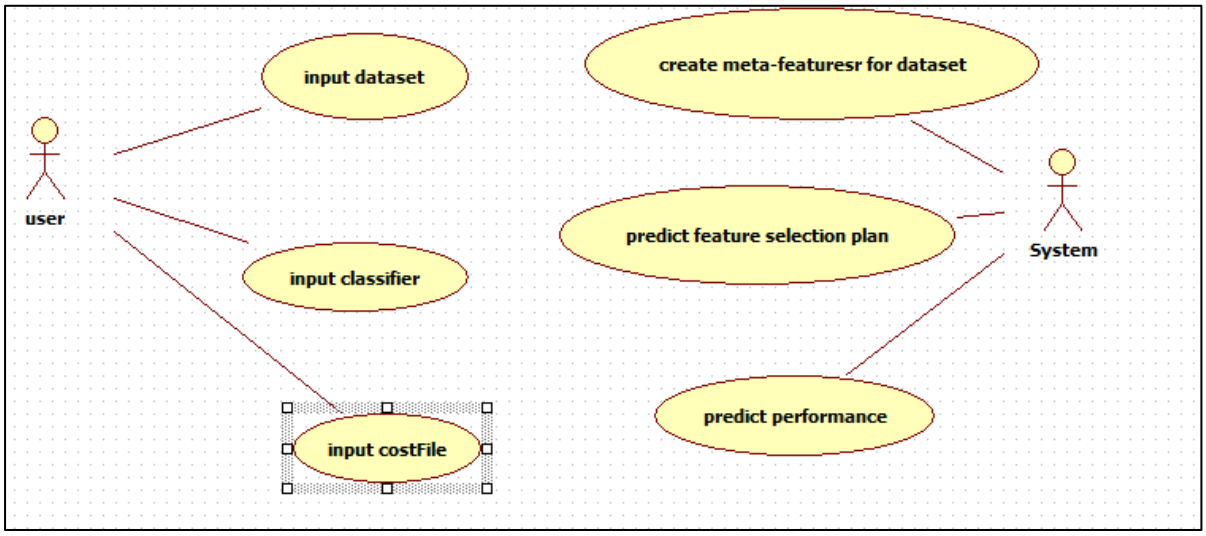

Figure 2B1-14: Predict feature selection and classifier performance a given dataset and given cost file, and given classifier

Cost Sensitive Meta-knowledge Development

The aim of this section is to show each individual task to develop a working scenario for cost senstive meta knowdelge development and how each object interacts to demonostrate those working scenario:

- Cost sensitive meta knowledge development scenario

In this scenario, large number of dataset are used to generate a meta-knowledge that recommend cost senstive with classifier that maximize accuracy and minmize cost, each given dataset is characterized, then diffrent cost senstive methods with diffrent wrapper methods are applied to convert cost insenstive base learners to cost senstive, then all previous result is fed to meta learner in meta learning base to develop a set of rules that used to create the needed meta knowledge, this scenario with its use cases are shown in Figure B1-15.

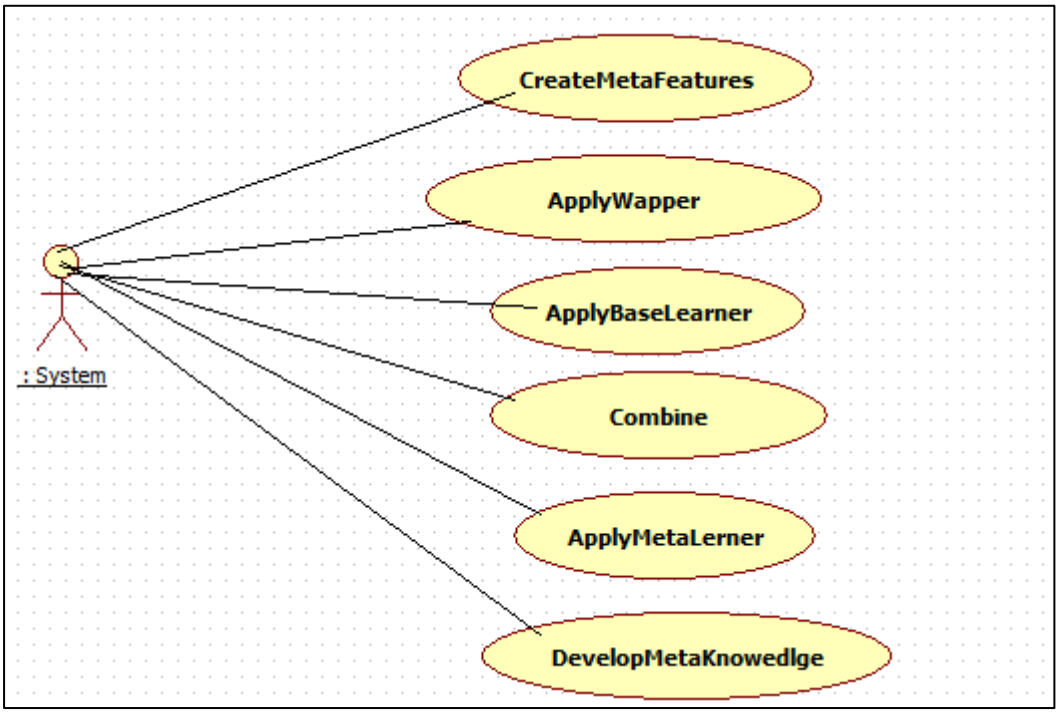

Figure B1-15: Cost Sensitive Meta-Knowledge Development Scenario

- Predict cost senstive method and classifier performance for a given dataset and given cost ratio scenario.

  The user inputs a dataset, the system generates meta-features for a given dataset, after meta-features are generated, the system predicts the performance of applying specific classifier and specific cost senstive plan on a given dataset with the selected cost ratio.

  Figure B1-16 shows predict cost sensitive plan and performance for given dataset and given costRatio.

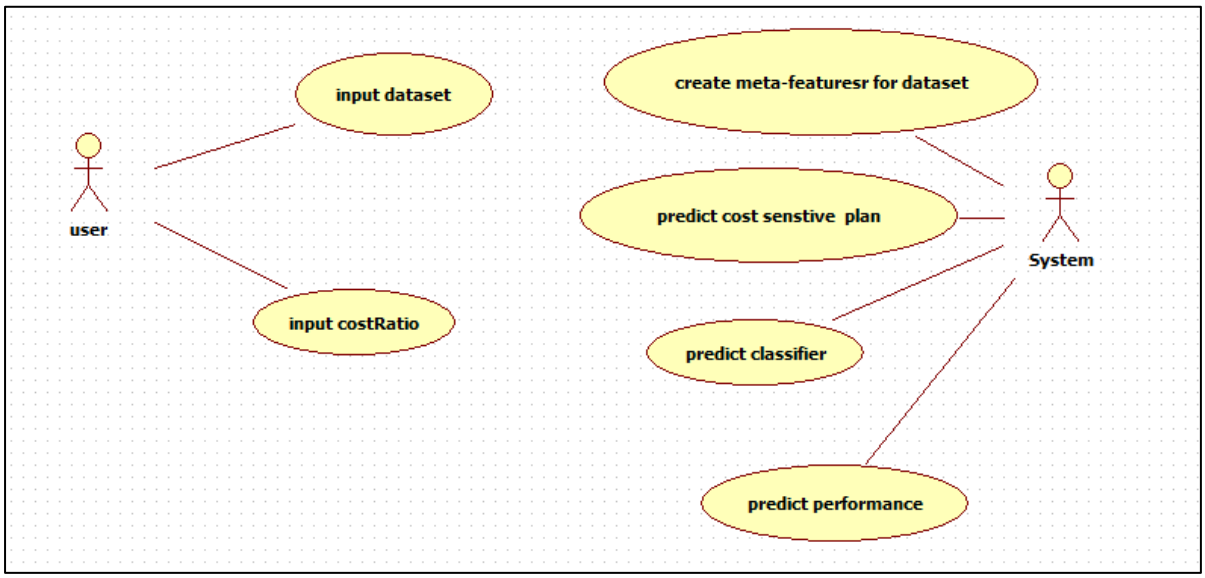

Figure B1-16: Predict cost sensitive plan and performance for given dataset and given costRatio

- Predict classifier performance for a given dataset, given cost ratio, given classifier, and given wrapper cost sensitive wrapper method.

  The user will input a dataset, the system will generate meta-features for a given dataset, after meta-features are generated, the system will predict the performance of applying specific classifier after adding given wrapper cost senstive method with given cost ratio, see Figure B1-17.

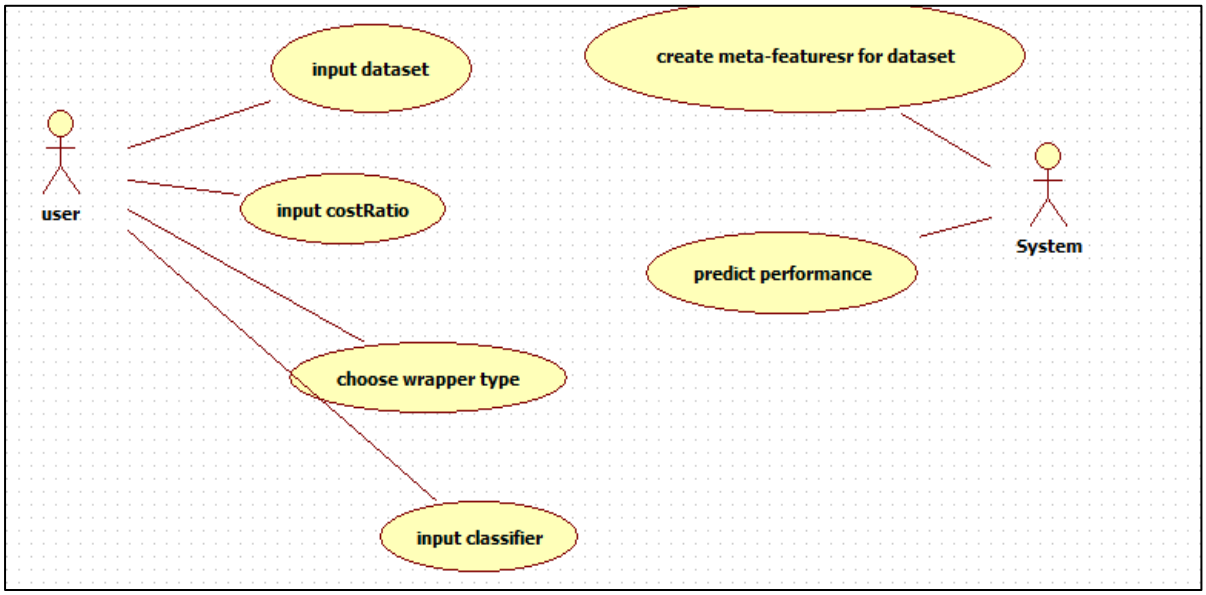

Figure B1-17: Predict classifier performance for a given dataset, given cost ratio, given classifier, and given wrapper cost sensitive wrapper method

- Predict cost senstive method and classifier performance for a given dataset and given cost file scenario.

  The user inputs a dataset, the system generates meta-features for a given dataset, after meta-features are generated, the system predicts the performance of applying specific classifier and specific cost senstive plan on a given dataset with the given cost file.

  Figure B1-18 shows predict cost sensitive plan and performance for given dataset and given cost file.

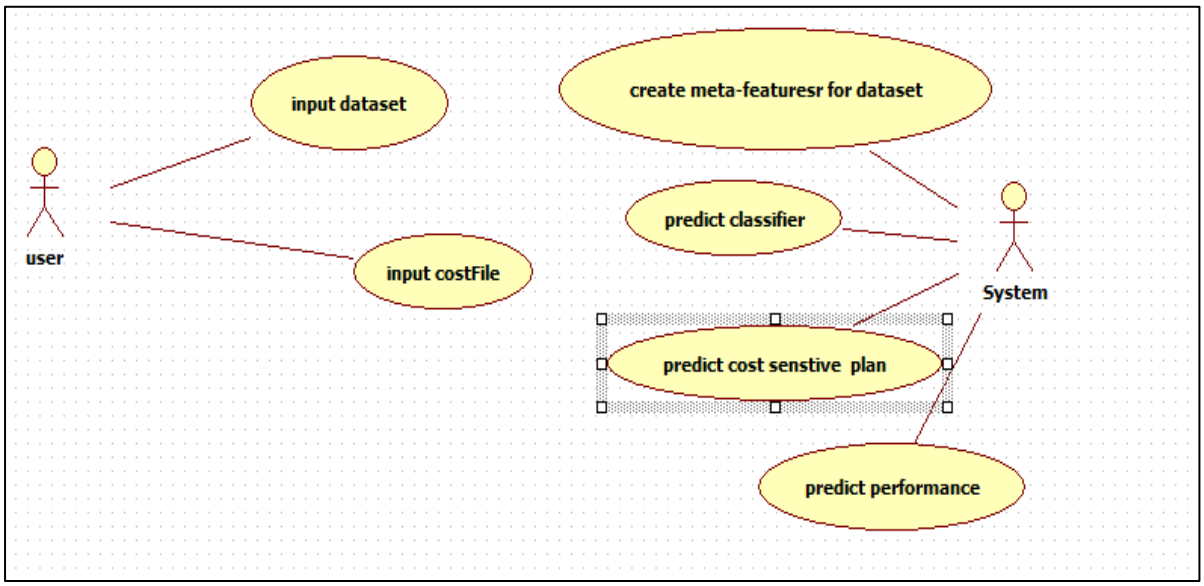

Figure B1-18: Predict cost sensitive plan and performance for given dataset and given cost file

B2. Active Learning Based on Clustering

The aim of this part of the system is seek more informative data that is selected using active learning based on clustering algorithim, following is class diagram for this part of the system that shows the main classes, thier links and mutiplicies.

Class Model

The first step in structuring this part of the system is to model the system domain in terms of a collection of classes and the relationships between them.

Following is the main classes that contribute in reaching this aim with each class attributes and associations, shown in B2-1.

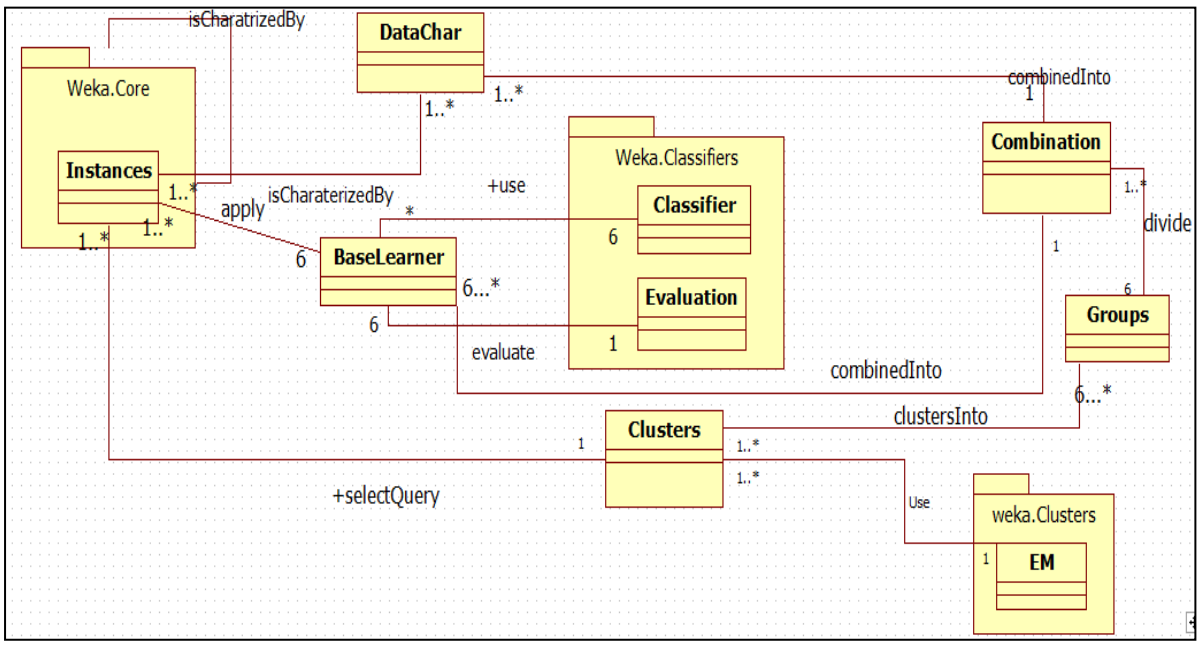

Figure B2-1: Active Learning based on Clustering Class Diagram

Appendix C System Implementation

As described before, this research develops a cost sensitive meta-learning system that aims to recommend the cost sensitive and insensitive methods that best suit a specific problem. The system has been implemented using java programming language. In particular, the JDK (Java development Kit (version 8) and eclipse (version 4.4) open source java environment language is used. For Data mining algorithms, there are many tools available for data mining and machine learning, but this thesis is built over the open source software suite WEKA (version 3.6) which stands for Waikato Environment for Knowledge Analysis. The main reason why I selected to use WEKA is its popularity, ease of use, and availability. WEKA is a popular tool used for data analysis, machine learning and predictive modelling that was developed by the University of Waikato in New Zealand using the programming language JAVA (Hall et al., 2009). WEKA is a big workbench for data analysis and machine learning which has a majority of the most needed algorithms that (are) ready to be used in the developed system. The following are some of the WEKA features that have been used in the developed cost sensitive system and ALBC algorithm:

Data pre-processing: WEKA supports a couple of popular text file formats such as CSV, and ARFF. This feature facilities a wide collection of supervised and unsupervised filters to convert each dataset to the format applicable in the algorithms which needs to be applied such as: ***Discretization***, ***AttributeSelection,NominalToBinary.***

Data clustering: Includes a wide range of algorithms for clustering such as ***SimplekMeans***, ***EM***, and ***CobWeb. EM (Expectation Maximization)*** is used in developing ALBC algorithm *which* assigns a probability distribution to each instance that indicates the probability of its membership to each cluster (Hall et al., 2009). The main benefit of using EM is its ability to decide on the number of the clusters by using a cross validation technique. Furthermore, it may give users the option to specify a priori the number of clusters to generate.

Classification: WEKA supports a huge collection of algorithms that have been implemented to perform classification on different types of datasets. These include Bayesian algorithms such as **Naïve Bayes**, mathematical functions such as ***Neural Network***, lazy classifiers

implementing nearest-neighbour such as **K-NN**, rule and tree-based classifiers such as **J48**, rule based algorithms such as ***OneR, Part*** and **ZeroR**.

Attribute selection: Methods to evaluate which attribute will perform best when predicting the classifier model. Those contain filter and wrapper methods such as ***CfsSubSetEval, GainRatioSubSetEval, InfoGainAttributeEval, OneAttributeEval, and WrapperSubSetEval***

And search strategies to find the minimum attribute subsets that contribute more in the predictive power such as ***BestFirst, Ranker,*** and ***GreedyStepWise.***

The user interface for the implemented application has been developed by using java web-based technology using Apache tomcat 7. Tomcat includes web applications which are the most popular HTTP server for running Java applications. The most benefit of using Apache tomcat is flexibility. For example, it is possible to run Apache on one physical server, and run the actual JSP and servlets on another machine.

The interface of the developed application for the system can provide a good interaction between the user and the system itself. The following options are available for cost sensitive meta-learning system:

- To choose either to work with cost sensitive learning or to work with cost in-sensitive learning. Cost sensitive learning includes ***cost sensitive without any wrapper method***, ***bagging, sampling,*** and ***minimum expected cost,*** while cost in-sensitive includes working with the classifiers without any cost sensitive methods.
- To choose to work with a specific classifier ***(J48, NN, Naïve Bayes, Part, ZeroR, and OneR).***
- To choose to work with a specific wrapper method (***None***, ***bagging, sampling,*** and ***minimum expected cost).***
- To upload the data file.
- To upload the cost file.
- To choose a specific cost ratio.
- To choose a system define classifier
- To choose a system define wrapper methods

Figure C1-1 shows a snapshot for cost sensitive meta-learning user interface

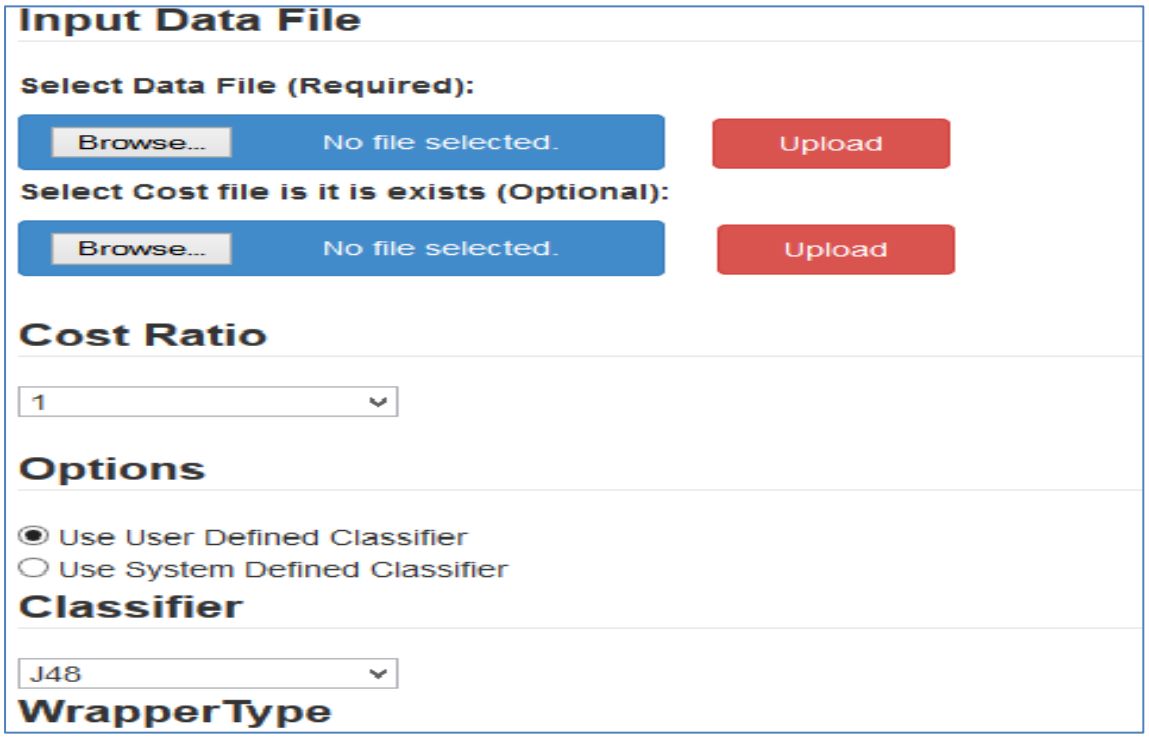

Figure C-1: Cost sensitive Meta learning user interface

Two CDs are provided with this thesis one contains cost sensitive meta-learning system provided with friendly user interface, and the other contains the application of developed ALBC algorithm.