COMPARISON OF C4.5 AND WEIGHTED C4.5 DECISION TREES FOR BREAST CANCER CLASSIFICATION

KHIN THUZAR WIN

M.C.Sc. **JANUARY 2020**

COMPARISON OF C4.5 AND WEIGHTED C4.5 DECISION TREES FOR BREAST CANCER CLASSIFICATION

By

KHIN THUZAR WIN

B.C.Sc.(Hons:)

A dissertation submitted in partial fulfilment of the requirement for the degree of

Master of Computer Science

M.C.Sc.

University of Computer Studies, Yangon
JANUARY 2020

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude and sincere appreciations to the following persons, who have contributed directly or indirectly towards the completion of this thesis and helped me make this dissertation possible.

I would like to express my gratitude and my sincere thanks to **Prof. Dr. Mie Mie Thet Thwin**, the Rector, the University of Computer Studies, Yangon, for allowing me to develop this thesis and giving me general guidance and workable environment during the period of study.

I would like to expect my special appreciation to **Dr. Thi Thi Soe Nyunt**, Professor and Head of Faculty of Computer Science, for her administrative support and encouragement in the thesis development.

My heartfelt thanks and respect go to my supervisor, **Dr. Aung Nway Oo,** Professor, Faculty of Computer Science, the University of Information Technology, for his invaluable recommendations regarding the thesis topic, giving me detailed guidance throughout the work of this thesis.

I would like to express my respectful gratitude to **Daw Htet Htet Aung**, Lecturer, the Department of Language, the University of Computer Studies, Yangon, for editing my thesis from the language point of view.

I would like to acknowledge my thanks to my teachers of the University of Computer Studies, Yangon and the University of Computer Studies (Hinthada), and all of my dear teachers from childhood to the present time.

In addition, I would like to thank the board of examiners for making precious comments and detailed corrections to my thesis and those who are pressing power to improve the end result.

Last but not least, I especially thank my parents and my family for raising me and inspiring me all the time. Finally, I am grateful to my colleagues and all my friends for their cooperation and help.

STATEMENT OF ORIGINALITY

I hereby certify that the work embodies in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

---------------------------------	------------------------------
Date	Khin Thuzar Win

ABSTRACT

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information. Classification is a data mining technique which addresses the problem of constructing a predictive model for a class attribute given the values of other attributes and some examples of records with known class. Decision tree is one of the most well-established classification methods.

This thesis presents a weighted C4.5 decision tree algorithm for breast cancer classification and compared with the classification results of traditional C4.5 algorithm. The weighted C4.5 algorithm is set to appropriate weights of preparation instances grounded on naïve Bayesian theorem before trying to construct a decision tree model. The aim of the proposed system is to examine the performance of weighted C4.5 decision tree algorithms. According to the experimental results, the accuracy of weighted C4.5 is 99.56% and traditional C4.5 is 94.27%. Therefore, the weighted C4.5 algorithm is better than traditional C4.5 algorithm on breast cancer dataset. This system is implemented by using Java language.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

People live in the information age – accumulation data is not difficult and warehousing – it is also inexpensive. Unfortunately, as the quantity of machine understandable information rises, the ability to understand and make practice of it does not save step with it development. An abundant amount of data can be automatically examined by means of tools provided by machine learning. Currently a topic of much interest in the machine learning and data mining communities, classification was studied widely in many fields. A system of data study applicable to extracting models that define significant data classes or predicting future data trends whose class label is unknown is classification. Classification can be employed in making intelligent decisions. Many classification methods have been suggested by researchers and are important for research and practical application in a variety of fields: including pattern recognition and artificial intelligence, statistics, vision analysis, medicine and son.

'Data mining' is an expression invented to define the act of moving through huge databases exploring alluring and new patterns. Data mining has become considerably important and a necessity today when data are bountiful and easily accessible. The automatic analysis of huge numbers of data is possible through the methods and instruments that the field of data mining provides. Data mining is one aspect of the course of Knowledge Discovery in Databases (KDD). Some searchers regard data mining as an equivocal and the term "Knowledge Mining" is preferable as it is very alike to gold mining. Data mining approaches are mostly grounded on inductive learning i.e., constructing a mode explicitly or implicitly by forming a generalization from enough preparation examples. The inductive approach forms a basic assumption that the prepared model is related to upcoming invisible examples. Specifically, any procedure of conjecture is considered an induction on conditions that conclusions are not logically drawn from premises. Data collection was traditionally agreed to be vital period in data scrutiny. An analyst would be able to select the variables to gather by means of the available domain knowledge. The amount of specified mutable was normally restricted and their principles could be recorded physically or using viva vex. If computer-aided analysis was to be used, the collected data had to be entered into numerical computer compendium or an electronic work sheet. Because the process of data gathering was expensive, analysts had to learn to make decisions on available information. Decision trees are regarded as well-known methods for representing classifiers. A decision tree is a classifier viewed as the repetitive subdivision of the instance space.

The decision tree is composed of nodes forming a ‘rooted tree’ i.e., a ‘directed tree’ with a node known as ‘root’ with no incoming edges. There is exactly one incoming edge in all other nodes. An internal node is a node with outgoing edges. All other nodes are known as leaf nodes. In a decision tree, it is each internal node subdivides the instance space into two or more sub-spaces by an assured discrete function of the input attribute’s values. Simply and greacheck frequently, each examination takes a particular attribute such that the attribute’s values subdivide the instance space. Concerning digital characteristics the condition deals with a variety. Each leaf is allocated to one class which indicates greacheck suitable goal value. On the other hand, the leaf may grip a probability vector that indicates the likelihood of the goal attribute having a definite value. Instances, from the source of a tree to a frond, are navigated and organized, following the result of the checks along the route. There have been many decision tree algorithms like C4.5, ID3 [9], CART [12] etc.

Classification is the process of finding a set of models (or functions) that describe and distinguish data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. Classification has been successfully applied to wide range application areas such as medical diagnosis, weather prediction, credit approval, customer segmentation, fraud detection among the different proposals. Classification is clearly useful in many decision problems where for a data item, a decision is to be made (which depend on the class to which data item belongs) [13].

Classification is a form of data analysis to extract models describing important data classes or to predict future data trends whose class label is unknown. In this study, weighted C4.5 algorithm is used for efficient classification. Breast cancer data set is used for checking of proposed method and then the results of normal C4.5 algorithm are compared with it.

1.1 Objectives of the Thesis

The main objectives of the thesis are as follows:

(i) To understand and use of C4.5 algorithm with weight values.

(ii) To use the Naïve Bayes probability for calculation of weight values.

(iii) To apply the weighted decision tree approach for breast cancer classification.

(iv) To compare the result of traditional decision tree and weighted decision tree for Breast Cancer Classification.

(v) To study the different decision tree algorithms and compare them in terms of their accuracy of Breast Cancer classification.

1.2 Organization of the Thesis

This thesis consists of five chapters.

Chapter 1 describes the introduction, objectives and organization of thesis are presented.

Chapter 2 explains data mining concepts and functionalities, classification and prediction, classification methods and classification algorithms.

Chapter 3 presents introduction to decision tree, the hierarchical nature of decision trees, appropriate problems for decision learning, decision tree induction, calculation of algorithms, filling in missing values and evaluating the accuracy of a classifier.

Chapter 4 describes system design, classifier accuracy measure, main page of the system and experimental results.

Finally, Chapter 5 presents the conclusion, advantages of the system, limitation and further extension of the system.

CHAPTER 2
BACKGROUND THEORY

Data Mining, also popularly known as Knowledge Discovery in Databases refers to extracting or “mining” knowledge from large amounts of data. Data mining techniques are used to operate on large volume of data to discover hidden patterns and relationships helpful in decision making.

2.1 Data Mining

Data mining can also be defined as knowledge mining from databases, knowledge extraction, data/pattern analysis, data archaeology and dredging. This is an interdisciplinary field that draws ideas from several areas of research including databases, machine learning and statistics. Data mining refers to the core process of a broader process of automatic information extraction called knowledge discovery in databases. A knowledge location in database is the important extraction of implicit, formerly unrevealed and potentially useful knowledge from data in large databases.

The iterative processes of knowledge discovery of data are data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation and knowledge presentation. Data cleaning is removing noise and inconsistent data from the database. In data addition,varisous data basis may be united. It is possible that numerous databases are combined in data blending. In data selection, data that are applicable to the analysis task are recovered from the database. In data transformation, data are converted or synthesized into forms which are suitable for mining by implementing summary or aggregation operations. Data mining is a crucial operation where intelligent methods are utilized for the purpose of extracting data patterns. Pattern evaluation discovers the truly interesting patterns that embody knowledge grounded on some interesting means. In the knowledge presentation, the uncovered knowledge is presented by using visualization and knowledge representation techniques to the user. Knowledge discovery and data mining have been increasingly well-known because the great quantity of stored data came out as computer storage became cheaper [8].

2.2 Data Mining Functionalities

Data mining responsibilities can be categorized into two groups: descriptive and predictive data mining. Descriptive data mining deliver information to know what is happening inside the data without a prearranged idea. Predictive data mining allows the user to acquiesce records with undefined pitch values, and the system guesses the undefined values built on porous patterns exposed form the database.

The descriptive function deals with the general properties of data in the database. The list of descriptive functions –

(i) Class/Concept Description

(ii) Mining of Frequent Patterns

(iii) Mining of Associations

(iv) Mining of Correlations

(v) Mining of Clusters

(i) Class/Concept Description

Class/Concept refers to the data associated with the classes or concepts. For example, in a company, the classes of items for sales include computer and printers, and concepts of customers include big spenders and budget spenders. These descriptions can be derived by the following two ways –

- **Data Characterization** – It refers to summarizing data of class under study that is called as target class.
- **Data Discrimination** – It refers to the classification of a class with some predefined group or class.

(ii) Mining of Frequent Patterns

Frequent patterns are those patterns that occur frequently in transactional data. The lists of kind of frequent patterns are-

- **Frequent Item Set** – It refers to a set of items that frequently appear together, e.g. milk and bread.

- **Frequent Subsequence** – A sequence of patterns that occur frequently such as purchasing a camera is followed by memory card.
- **Frequent Substructure** – A substructure refers to different structural forms, such as graphs, trees, or lattices, which may be combined with item sets or subsequences.

(i) Mining of Association

Associations are used in retail sales to identify data patterns that are frequently purchased together. This process refers to the process of determining association rules and uncovering the relationship among data.

(ii) Mining of Correlations

Correlation is a kind of additional analysis that performs to uncover interesting statistical correlations between associated-attribute-value pairs or between two item sets to analyze positive, negative pair or no effect on each other.

(iii) Mining of Clusters

Cluster refers to a group of similar kind of objects. Cluster analysis analyses group of objects that are very similar to each other in same cluster but are highly different from the objects in other clusters [6].

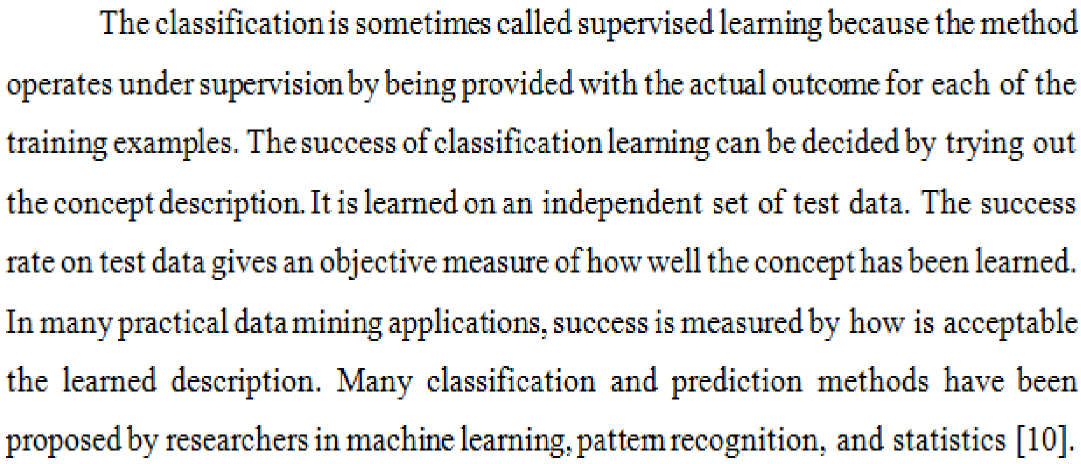
2.3 Classification and Prediction

Classification is the process of finding a model that describes the data classes or concepts. The purpose of classification is to use this model to predict the class of objects whose class label is unknown. This derived model is based on the analysis of sets of training data i.e. the data object whose class label is well known. The derived model can be presented as Classification (IF-THEN) Rules, Decision Trees, Mathematical Formulae and Neural Networks.

Prediction is used to predict missing data or unavailable numerical data values rather than class labels. Regression Analysis is generally used for prediction. Prediction can also be used for identification of distribution trends based on available data. Outlier Analysis describes the data objects that do not comply with the general behavior or model of the data available. Evolution Analysis describes the description and model regularities or trends for objects whose behavior changes over time [10].

2.4 Classification Methods

The classification methods are classified in two groups such as supervised methods and unsupervised methods. The supervised methods used are Bayesian classifier, Decision Tree, Artificial Network and Support Vector Machine, whereas the unsupervised methods are clustering such as an adaptation of the K-means clustering method [8].



The classification is sometimes called supervised learning because the method operates under supervision by being provided with the actual outcome for each of the training examples. The success of classification learning can be decided by trying out the concept description. It is learned on an independent set of test data. The success rate on test data gives an objective measure of how well the concept has been learned. In many practical data mining applications, success is measured by how is acceptable the learned description. Many classification and prediction methods have been proposed by researchers in machine learning, pattern recognition, and statistics [10].

2.5 Classification Algorithms

Several cataloging and estimate algorithms have been planned by investigators in instrument education, professional systems, number, and neurobiology. Classification algorithm is applied to the preparation data set. The outputs of the classifier are stored for larger usage; this stage is known as Learn model. Then check data (with known classes) is checked to the classifier (Apply Model or Check Model). If the output of the classifier is the same as the known class, then the checking accuracy is good, which means the classifier algorithm and preparation sample have the good performance. If the system got the poor accuracy in the checking phase, it has bad classifier or bad preparation data set. There are preparation and checking processes in the classification. Preparation processes learn models using one of the learning algorithms (classification algorithm) from preparation data set. The output model is then checked with checking data set to deduct the checking samples [6]. There are several classification algorithms and most widely used algorithms are

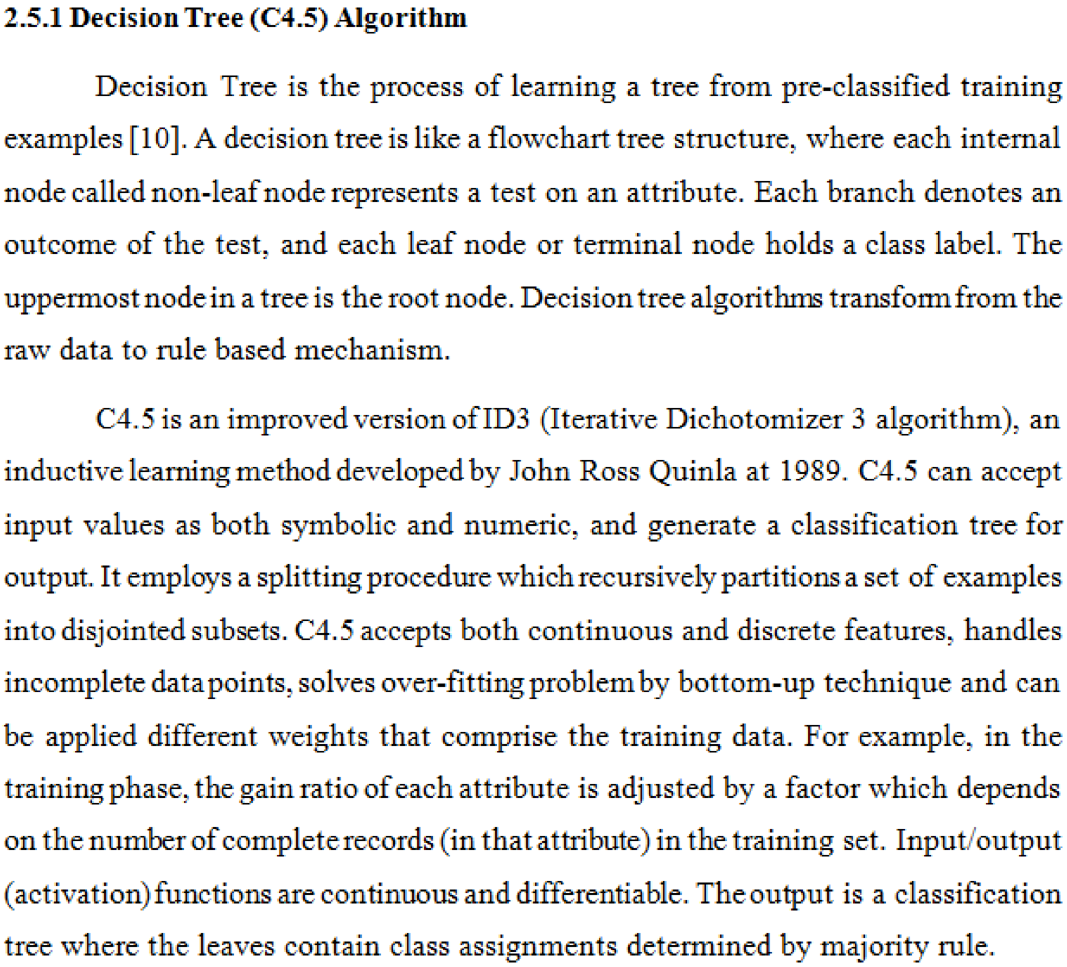
(i) Decision Tree (C4.5)

(ii) Naïve Bayesian Classification

(iii) Neural Network

(iv) Support Vector Machine Algorithm

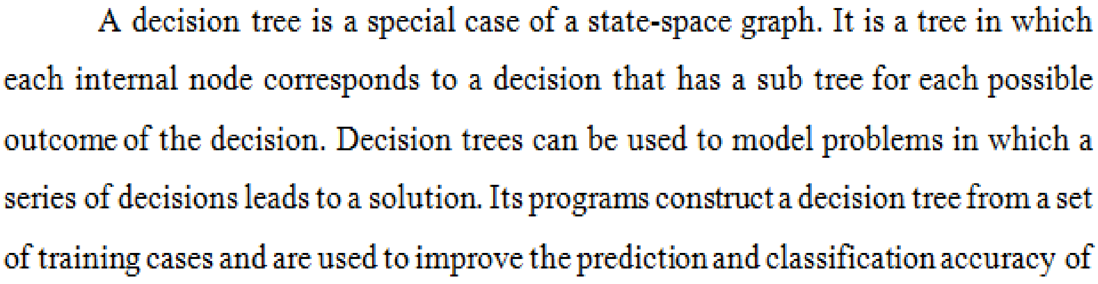
(v) Genetic Algorithms



2.5.1 Decision Tree (C4.5) Algorithm

Decision Tree is the process of learning a tree from pre-classified training examples [10]. A decision tree is like a flowchart tree structure, where each internal node called non-leaf node represents a test on an attribute. Each branch denotes an outcome of the test, and each leaf node or terminal node holds a class label. The uppermost node in a tree is the root node. Decision tree algorithms transform from the raw data to rule based mechanism.

C4.5 is an improved version of ID3 (Iterative Dichotomizer 3 algorithm), an inductive learning method developed by John Ross Quinla at 1989. C4.5 can accept input values as both symbolic and numeric, and generate a classification tree for output. It employs a splitting procedure which recursively partitions a set of examples into disjointed subsets. C4.5 accepts both continuous and discrete features, handles incomplete data points, solves over-fitting problem by bottom-up technique and can be applied different weights that comprise the training data. For example, in the training phase, the gain ratio of each attribute is adjusted by a factor which depends on the number of complete records (in that attribute) in the training set. Input/output (activation) functions are continuous and differentiable. The output is a classification tree where the leaves contain class assignments determined by majority rule.



A decision tree is a special case of a state-space graph. It is a tree in which each internal node corresponds to a decision that has a sub tree for each possible outcome of the decision. Decision trees can be used to model problems in which a series of decisions leads to a solution. Its programs construct a decision tree from a set of training cases and are used to improve the prediction and classification accuracy of the algorithm. It is extensively practical in various areas since it is stout to data balances or supplies by comparing to other data mining techniques.

2.5.2 Naive Bayesian Classification

Naive is a statistical classifier that can predict class membership likelihoods such as the probability a given example belongs to a particular class [3]. Naive Bayes classifier is a probabilistic classifier that produces probability estimates grounded on the Bayes theorem rather than predictions. For each class value, they approximate the probability that a given instance belongs to that class by using a minor amount of preparation data to approximation. It assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. Bayesian classifiers have also exhibited high accuracy and speed when applied to large database.

The Naive Bayes classifier technique is grounded on Bayes' theorem and is particularly appropriate when the dimensionality of the feature space is high. For example, a vector $x=(x_1,x_2,\ldots.,x_n)$ of *n* features is associated with each observation and Naive Bayes learns the class conditional probabilities $p(x_j|y_j)$ of each categorical variable j, j=1,2,….,n, assumed the class maker y_i. A new observation with feature vector x is classified by using the Bayes' rule to compute the posterior probability of each class y_i given the vector of attributes. The basic assumption of Naive Bayes' classifier is that the variables are conditionally independent given the class label.

2.5.3 Neural Network (NN)

A neural network (NN) can be described as reasoning model grounded on the human head [20]. A NN contains of a amount of interrelated processors called neurons. Firstly, a neuron obtains input signals from its effort relatives, calculates an output signal and transmits this signal through its output relatives. An input signal can be raw data or the outputs from other neurons. The output signal can be either a final solution to the problem or an input to other neurons. A NN is set through repeated adjustments of these weights. A neural network model, the branch of artificial intelligence is commonly referred to as Artificial Neural Networks (ANNs). ANN constructs the system to execute task, instead definite tasks.

Neural Networks are accomplished of predicting novel explanations from current annotations. The neurons within the system work composed, in parallel, to crop an production function. Since the computation is executed by the cooperative neurons, a neural network can motionless crop the production function even if some of the individual neurons are faulty (the network is strong and doing lenient). Neural

Networks (NN) are important data mining tool used for classification and clustering [21]. It is an attempt to build machine that can mimic brain activities and be able to learn. Basic NN consists of three layers such as input, output and hidden layer. Each layer can have number of nodes and nodes are connected from input layer to hidden layer and hidden layer's nodes are connected to the nodes from output layer. Those connections represent weights between nodes. Back Propagation Neural Network (BPNN), one of the most popular NN algorithms needs a very large number of training samples and need a lot of time to gradually approach good values of the weights.

2.5.4 Support Vector Machine (SVM)

The concept of decision planes to define decision boundaries is Support Vector Machine (SVM) that supports both regression and classification. A decision plane is the one that splits between a set of objects having different class membership [11]. SVM performs classification task by creating hyper plane in a multidimensional space that splits cases of different class labels. It uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyper plane.

SVM was first proposed by Vapnik at 1995 as learning systems for binary classification [20]. It is trained using an algorithm from optimization theory and statistical learning theory to derive a separating hyper plane in a high dimensional feature space. SVMs are based on a nonlinear mapping of the problem data into a higher dimension feature space. However, the learning algorithm may be inefficient and SVMs may be difficult to implement as a large number of 17 parameters is required. In addition, small training samples can result in over fitting, with poor generalization ability. The original model proposed by Vapnik was a linear classifier, but other types were later proposed in order to improve the accuracy of the original model. The main difference of the new models matched to the initial model is the function used to map the data into a higher dimensional space. New functions were proposed, namely: polynomial, Radial Basis Function (RBF) and sigmoid. All these functions transform the original data into a higher dimensional space and then the linear classifier is used subsequently.

2.5.5 Genetic Algorithms

Genetic Algorithms challenge to include the planning of usual development [1]. Genetic algorithms are used to discover classification rules for data that can be used for predictions. The genetic algorithms are adaptive techniques that can be successfully used to solve complex search and optimization problems. They are grounded on the principles of genetics and Darwin's ordinary choice theory ("the one that is best endowed, survives"). In data mining, genetic algorithms have been effectively used to determine classification rules, to search for appropriate cluster centers and to select the attributes of interest in forecasting the value of a target characteristic and so on. By using some hybrid algorithms, classification of instances was performed such as Genetic Algorithms and Particle Swarm are optimized, respectively by Naive Bayes and k-Nearest Neighbors. Genetic algorithms were effectively applied to solve classification problems such as classification, heart disease classification and the classification of emotions on the human face.

The fitness functions of the genetic algorithms used for mining classification rules may contain metrics concerning predictive accuracy, rule comprehensibility as well as rule interestingness [19]. Diverse studies suggest genetic algorithms with fitness functions that take into consideration in different ways. Genetic algorithms are a form of optimization algorithm, import they are expended to search the highest or lowest of a function [21]. These algorithms are remote additional efficient and powerful than arbitrary and complete search algorithms. In data mining, the advantage of Genetic algorithm becomes more obvious when the find space of a task is huge. Genetic algorithm is a find technique expended in calculating to search correct or estimated solution to optimization and find efforts.

CHAPTER 3
BREAST CANCER CLASSIFICATION USING C4.5 AND WEIGHTED C4.5 DECISION TREES

C4.5 is an algorithm expended to make a decision tree settled by Ross Quinlan. C4.5 is an allowance of Quinlan’s earlier ID3 algorithm. The weighted C4.5 algorithm is assigned appropriate weights of preparation instances which improve the classification accuracy. The decision tree produced by C4.5 can be expended for classification.

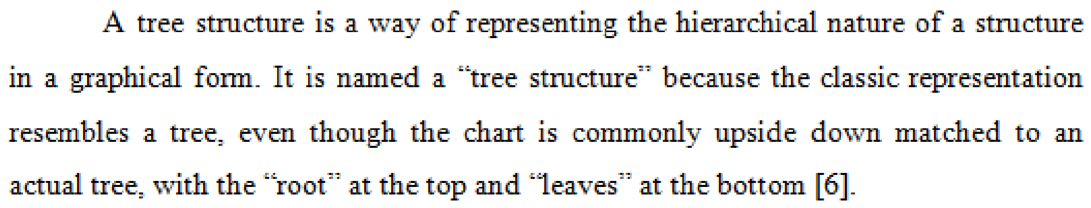
3.1 Introduction to Decision Tree

A decision tree is a catalog uttered as a repeatable divider of the occasion space. The decision tree comprises of bumps that procedure a root tree, sense it is a focused tree with a bumps called “source” that has no received edges. All other bumps have accurately one external point. A bump with leaving brinks is called an inner or check bumps. All extra bumps are called fronds (also known as incurable or decision bumps). In a decision tree, each internal bump separates the request area into two or more sub-places rendering to an assured discrete function of the record features morals. In the humblest and greacheck frequent event, each check studies a only attribute, such that the instance area is subdivided according to the attribute’s value. In the case of digital attributes, the condition donates to a series. Each foliage is allocated to one class expressive the greacheck suitable goal value. Alternatively, the leaf may grip a likelihood vector representative to the likelihood of the aim attribute having a sure value. Instances are categorized by routing them from the source of the tree miserable to a leaf, according to the result of the examinations along the track.

In instance of digital attributes, decision trees can be construed interpreted as a gathering of overexcited flat surface. Naturally, decision-makers favor fewer difficult decision trees, as these are extra understandable. The tree difficulty has on its correctness and is obviously measured by the discontinuing conditions used and the cropping method employed. Decision tree induction is carefully associated to rule investing. Each track from the source of a decision tree to single of its sprigs can transform into a rule basically touching the checks along the track. It forms the precursor portion and takes the foliage's class prediction as a category value. The subsequent instruction fixed can then basic to expand its directness to a human operator, and possibly its accuracy.

Decision tree algorithms have been expended for classification in many spaces, such as medicine, developed and creation, business evaluation, stargazing, and molecular natural science [14].

3.2 The Hierarchical Nature of Decision Trees



A tree structure is a way of representing the hierarchical nature of a structure in a graphical form. It is named a "tree structure" because the classic representation resembles a tree, even though the chart is commonly upside down matched to an actual tree, with the "root" at the top and "leaves" at the bottom [6].

3.3 Appropriate Problems for Decision Learning

Although a variability of decision tree learning methods have been recognized with fairly conflicting abilities and necessity, decision tree learning is commonly maximum matched to problems with following appearances:

- Occasions are denoted by attribute-value couples. Occasions are defined by a stable usual of characteristics and their morals. The easiest condition for decision tree learning is when each attributes proceeds on a small number of disjoint possible values. However, postponements to the essential algorithm permit managing real-valued characteristics as fine.
- The goal function has distinct production morals. Decision tree methods aim to learning functions with two or more possible output values. It has more advances in extensions by allowing the learning goal functions with real-valued output.
- Disjunctive descriptions may be needed. According to the above noted, decision trees obviously denote disjunctive terms.
- Although preparation data may include errors, decision tree learning methods are strong to mistakes, both mistakes in classifications of the preparation examples and mistakes in the characteristic values that explain these examples.

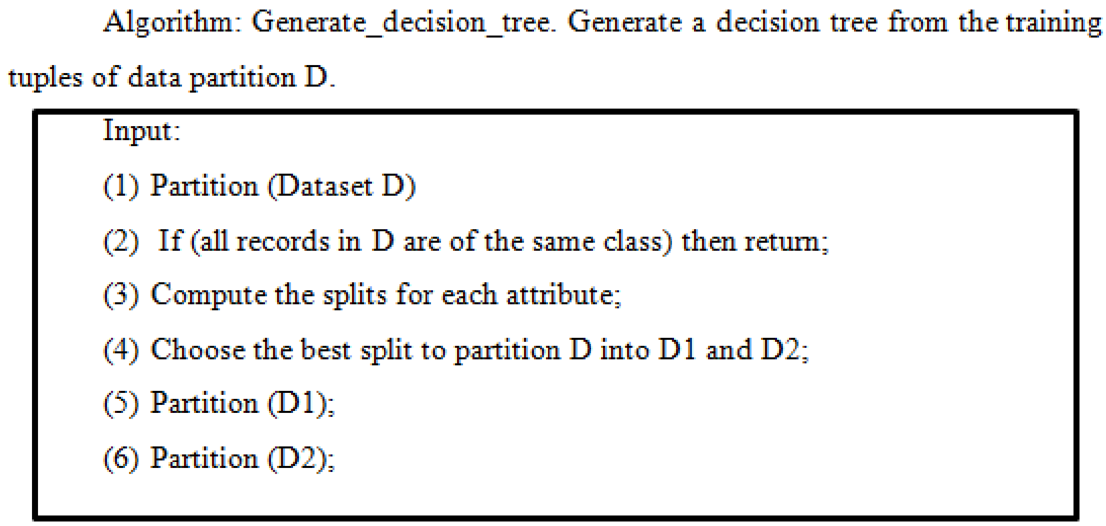
- The preparation data may include disappeared values while decision tree methods can be expended even when some preparation examples have indefinite values.

Decision tree methods can be expended even when some preparation examples have indefinite values. Many practical troubles have been established to suitable these characteristics. Decision tree learning has therefore been useful to difficulties such as equipment faults their source and advance applications by their probability of avoidance on expenses. Such problems, in which the task is to categorize examples into one of a separate set of possible groups, are often referred to as classification difficulties.

3.4 Decision Tree Induction

J. Ross Quinlan is a investigator in machine learning established a decision tree algorithm identified as ID3 [9] (Iterative Dichotomiser). Quinlan later presented C4.5 [8] (a successor of ID3), which converted a benchmark to which lacheck supervised learning algorithms are often compared.

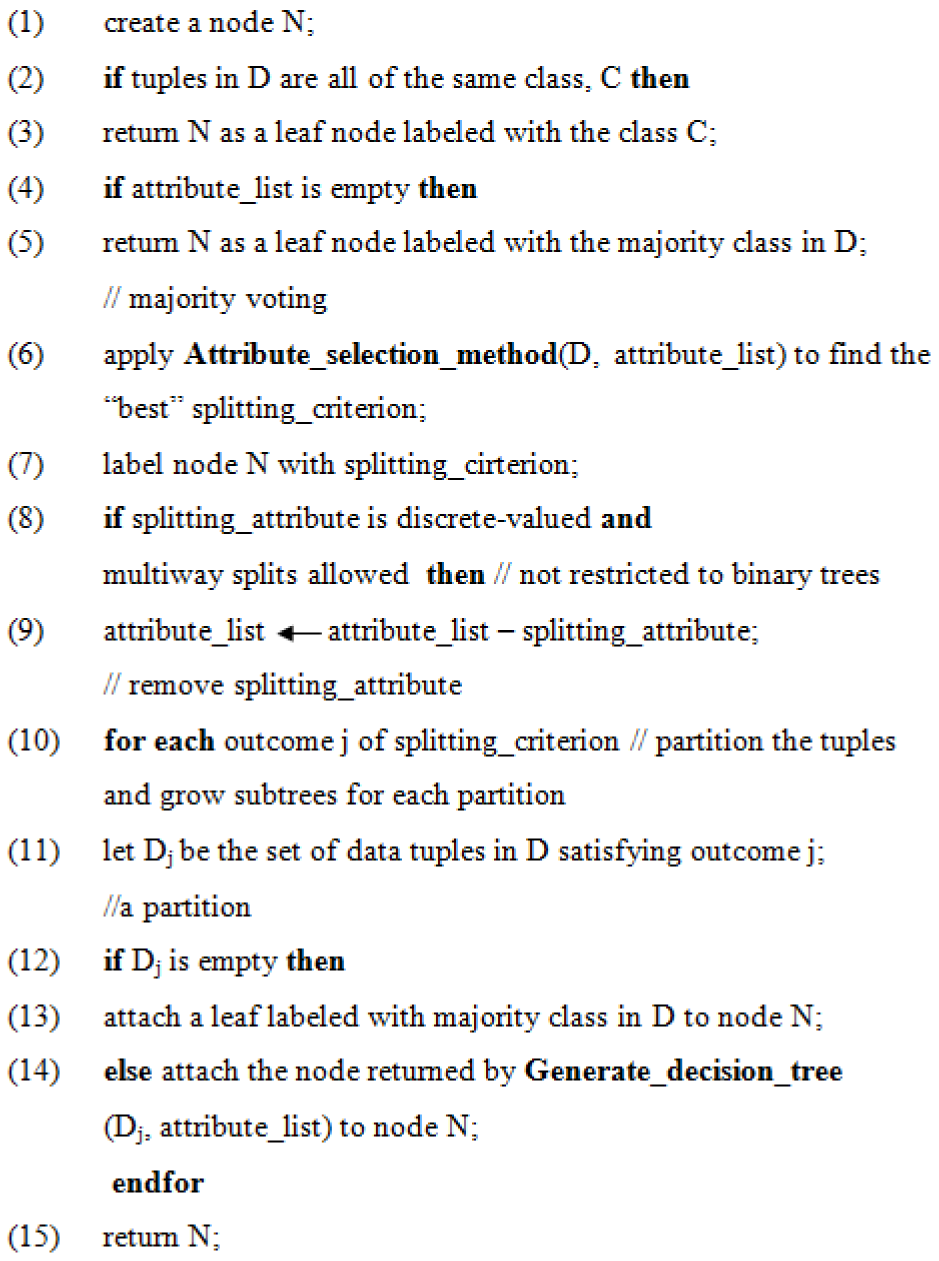
C4.5 adopt a desirous (i.e., non-backtracking) method in which decision trees are built in a high-low repeat divide-and-conquer way. Most algorithms for decision tree induction also follow such a high-low approach, which starts with a preparation arranged of tuples and their related class labels. The preparation set is repeat partitioned into lesser subsets as the tree is being constructed. A fundamental decision tree algorithm [5] is concise as follow:



Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition D.

```
Input:
(1) Partition (Dataset D)
(2)  If (all records in D are of the same class) then return;
(3) Compute the splits for each attribute;
(4) Choose the best split to partition D into D1 and D2;
(5) Partition (D1);
(6) Partition (D2);
```

Output: A decision tree.
Method:



```
create a node N;
if tuples in D are all of the same class, C then
    return N as a leaf node labeled with the class C;
if attribute_list is empty then
    return N as a leaf node labeled with the majority class in D;
    // majority voting
apply Attribute_selection_method(D, attribute_list) to find the
    "best" splitting_criterion;
label node N with splitting_cirterion;
if splitting_attribute is discrete-valued and
    multiway splits allowed then // not restricted to binary trees
    attribute_list ← attribute_list – splitting_attribute;
    // remove splitting_attribute
for each outcome j of splitting_criterion // partition the tuples
    and grow subtrees for each partition
    let D_j be the set of data tuples in D satisfying outcome j;
    //a partition
    if D_j is empty then
        attach a leaf labeled with majority class in D to node N;
    else attach the node returned by Generate_decision_tree
        (D_j, attribute_list) to node N;
endfor
return N;
```

3.4.1 C4.5 Decision Tree Algorithm

The C4.5 algorithm is the modified version of ID3 algorithm which chooses cracking characteristics from a dataset with the highest information gain. Input to C4.5 involves of a collection of preparation cases D, each having a tuple of values for a stable set of attributes A= {A_1, A_2, …, A_k} and a class attribute. An attribute A_a is showed as uninterrupted or disconnected according to whether its values are digital or normal.

The fundamental algorithm for decision tree induction is grasping algorithm that concepts decision trees in a high-low repeat split-and-conquer manner. A decision tree can be expended to categorize an example by initial at the tree and affecting through it until a leaf node which provides the classification of the instance. Decision trees are powerful and popular tools for classification and prediction [7].

Decision tree generation involves of two phases. Information gain (or recheck entropy reduction) is chosen as check attributes for the current node.

The expected information (entropy) needed to classify a tuple in D:

$$\mathrm{Info(D)} = -\sum_{\mathrm{i=1}}^{\mathrm{m}} \mathrm{p_i} log_2(\mathrm{p_i}) \qquad 2.1$$

Let p_i is the probability that an arbitrary tuple in D belongs to class C_i and m is the quantity in class label. A log function to the base 2 is used, because the information is encoded in bits. The information is grounded on the proportions of tuples of each class [16].

Information needed (after using attribute A to split D into v partitions) to classify D:

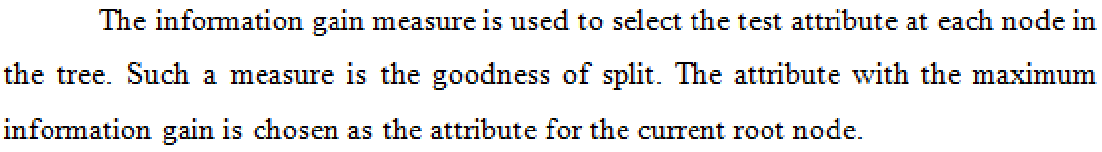
$$Info_A(D) = \sum_{j=1}^{v} \frac{|Dj|}{|D|} * Info(D_j) \qquad 2.2$$

where $\mathrm{Info_A(D)}$ is the predictable information of each attribute in data D and v is types of data in that attribute. Information gained by diverging on attribute A. The term of $\frac{|Dj|}{|D|}$ performances as the weight of the j$^{\mathrm{th}}$ separation. The lesser the predictable information needed, the larger the purity of separations.

$$Gain(A) = Info(D) - Info_A(D) \qquad 2.3$$

In other words, *Gain(A)* is the predictable decrease in entropy manufactured by expressive the value of attribute A. The algorithm calculates the information gain of each attribute.

The attribute with the maximum information gain is selected as the examination attribute for the current node. A node is manufactured and categorized with the attribute, divisions are manufactured for each value of the characteristic, and the examples are split accordingly.



The information gain measure is used to select the test attribute at each node in the tree. Such a measure is the goodness of split. The attribute with the maximum information gain is chosen as the attribute for the current root node.

An arithmetical property, which is called information gain, is expended. Gain instruments how well a specified attribute splits preparation samples into targeted classes. The one with the maximum information (information being the best useful for classification) is chosen. In order to describe gain, an impression from material theory called entropy is first borrowed. Entropy measure the quantity of impression in an a characteristic [15].

The notion of maximum information gain is used in the C4.5 algorithm to decide which attribute to select. If an attribute has a diverse value for apiece record, then this attribute will hold the highest information gain and the preparation set will be subdivided according to this attributes. Such separation in the preparation data is useless and C4.5 expends the Gain Ratio to avoid this. Quinlan indicated that the gain ratio criterion is robust and typically gives a conformity better choice of check attribute [2].

Assume a set of preparation instances D and attribute A, with value (A_1,A_2,…,A_m) used for the root if the decision tree, separations C into subsets (A_1,A_2,…,A_m) where D_i includes those objects in D that have value A of A_i. SplitInfo for each attribute is computed in equation 2.4.

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} * log_2(\frac{|D_j|}{|D|}) \qquad 2.4$$

$SplitInfo_A(D)$ is the information due to the split of D on the basis of the value of the categorical attribute A.

GainRatio for each attribute may be computed by equation 2.5. The attribute that yields the largest GainRatio is chosen for the decision node.

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)} \qquad 2.5$$

3.4.2 Weighted C4.5 Decision Tree Algorithm

Weighted decision tree learning algorithm was developed by assigning appropriate weights to preparation instances, which improve the classification accuracy. The weights of the preparation instances are calculated using maximum posteriori hypothesis of Naïve Bayesian theorem. Weight of each preparation instance is calculated with the maximum value of the class conditional probabilities.

Weighted C4.5algorithm calculates the highest information gain by using these weights and builds the decision tree model for classification. Given a preparation dataset, the weighted C4.5 algorithm initializes the weights of each preparation instance, *Wi* by highest posterior probability for that preparation instance. The algorithm uses the weight value calculated from Naïve Bayes probabilistic model to initialize the weights of each preparation instance.

The naïve Bayesian classifier is founded on Bayes' theorem. Expect that there are m classes, C_1, C_2, …, C_m. The classifier forecasts an invisible example maximum next probability hardened on X. In other words, X is appointed to class C_i if and only if

$$P(Ci|X) > P(Cj|X) \quad for\ 1 \le j \le m, j \ne i.$$

By Bayes' theorem

$$\boldsymbol{P(C_i\,|\,\mathbf{X}) = \frac{P(\mathbf{X}\,|\,C_i)P(C_i)}{P(\mathbf{X})}} \tag{2.6}$$

where, $P(C_i|X)$ is highest posteriori hypothesis of C_i conditional probability on X. C_i is class. P(X) is constant for all classes.

As P(X) is persistent for all classes, only $P(X|C_i)P(C_i)$ need to be enlarged. Given a set preparation data, $P(C_i)$ can be predictable by including how often each class happens in the preparation data. To decrease the calculable cost in predicating $P(X|C_i)$ for all probable X, the classifier creates a naïve supposition that the attributes expended in relating X are provisionally liberated of each other specified the class of X. Thus, specified the attribute values $(x_1,x_2,\ldots,x_n)$ that express X, the researches have

$$W_i = argmax\ P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i) \tag{2.7}$$

The naïve Bayesian classifier is sample to expend and proficient to acquire. It needs only one scan of the preparation data. In spite of the point that the free assumption is often despoiled in repetition, naïve Bayes often conchecks well with more erudite classifiers. In other words, the forecast class brand is the class C_i for which $P(X|C_i)$ $P(C_i)$ is the greacheck [5].

The expected information required to classify a tuple in dataset D is calculated by applying equation (2.1). In this case, p_i is the relative frequency of class i in D, *where*, p_i is the probability that an arbitrary tuple in D belongs to class C_i and m is the quantity in class label. The log function to the base 2 is used, because the information is encoded in bits. The information is based on the proportions of tuples of each class. The sum is computed over m classes.

To determine the information required to classify D, all the possible subsets that can be formed using known values of attribute A are examined. When thinking a split, a weighted sum of the impurity of each resulting separation is calculated. And then $Info_A(D)$ is calculated by applying equation (2.2). In this time, the value of equation (2.2) is defined as follows:

$|D_j|$ = the set of tuple with weight value in D that have outcome aj of A,

$|D|$ = total weight value tuple

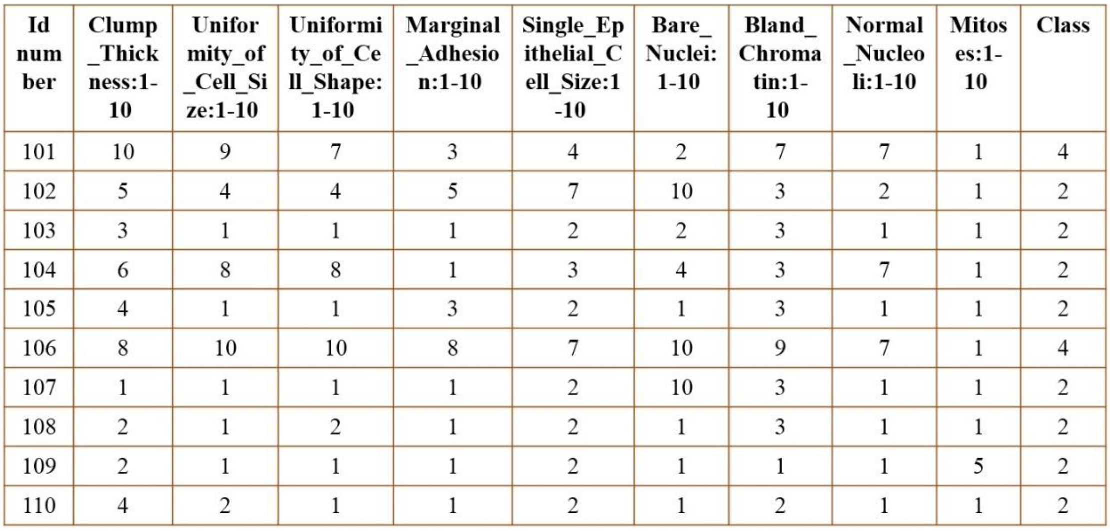
Information gain means as the dissimilarity between the unique information necessity (i.e., grounded on just the proportion of classed) and the novel necessity (i.e., got after separating on A) by using equation (2.3) and gain ratio to overcome the problem by using equation (2.4) and equation (2.5). The decision tree is constructed grounded on the weights of preparation data which results from naïve Bayes probabilities.

$Info_A(D)$, Gain(A) ,$SplitInfo_A(D)$ and GainRatio are calculated to assign weight value. The attribute with the greacheck information Gain Ratio is preferred as the check attribute at ach node in the tree. Such a scale is the golly of divided. The characteristic with the maximum information gain ratio is selected as the attribute for the existing root node.

3.5 Calculation of Algorithms

Breast Cancer dataset from UCI machine learning repository [25] is used to show as an example for implementing the algorithms: C4.5 and weighted C4.5. Breast Cancer dataset contains 683 tuples with 10 attributes and 2 classes. As an implementation example, 10 tuples of breast cancer dataset are used to train the sample decision tree model. 8 tuples of 10 tuples belong to class label 2 (Benign) and 2 tuples belong to class label 4(Malignant). A root node is created for 10 tuples. Table 3.1 represents 10 sample total records of the Breast Cancer Dataset.

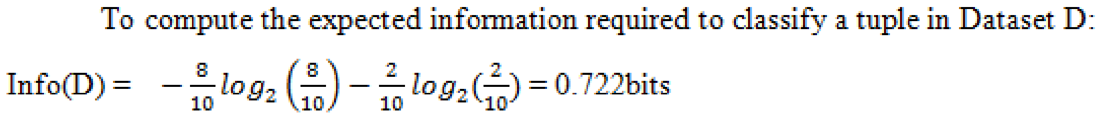
Table 3.1 Sample Data Records



Id number	Clump_Thickness:1-10	Uniformity_of_Cell_Size:1-10	Uniformity_of_Cell_Shape: 1-10	Marginal_Adhesion:1-10	Single_Epithelial_Cell_Size:1-10	Bare_Nuclei: 1-10	Bland_Chromatin:1-10	Normal_Nucleoli:1-10	Mitoses:1-10	Class
101	10	9	7	3	4	2	7	7	1	4
102	5	4	4	5	7	10	3	2	1	2
103	3	1	1	1	2	2	3	1	1	2
104	6	8	8	1	3	4	3	7	1	2
105	4	1	1	3	2	1	3	1	1	2
106	8	10	10	8	7	10	9	7	1	4
107	1	1	1	1	2	10	3	1	1	2
108	2	1	2	1	2	1	3	1	1	2
109	2	1	1	1	2	1	1	1	5	2
110	4	2	1	1	2	1	2	1	1	2

3.5.1 Calculation of C4.5 Algorithm

The C4.5 algorithm also chooses the attribute “Bland Chromatin” as the splitting criterion for the root node because this attribute with the maximum Gain Ratio is selected to carry on the expansion of the classification. The expected information required to classify a tuple in preparation set is calculated using Equation 2.1:



To compute the expected information required to classify a tuple in Dataset D:

$$\text{Info(D)} = \; -\frac{8}{10}log_2\left(\frac{8}{10}\right) - \frac{2}{10}log_2(\frac{2}{10}) = 0.722\text{bits}$$

Next, the expected information necessity for each attribute is computed by using Equation 2.2:

The expected evidence required to catalog a tuple in D if the tuples are separated according to Clump_Thickness is

$$\begin{aligned}\text{Info}_{\text{Clump\_Thickness}}(\text{D}) = {} & \frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)+\frac{2}{10}\times\left(-\frac{2}{2}log_2\frac{2}{2}-\frac{0}{2}log_2\frac{0}{2}\right)\\ & +\frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)+\frac{2}{10}\times\left(-\frac{2}{2}log_2\frac{2}{2}-\frac{0}{2}log_2\frac{0}{2}\right)\\ & +\frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)+\frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)\\ & +\frac{1}{10}\times\left(-\frac{0}{1}log_2\frac{0}{1}-\frac{1}{1}log_2\frac{1}{1}\right)+\frac{1}{10}\times\left(-\frac{0}{1}log_2\frac{0}{1}-\frac{1}{1}log_2\frac{1}{1}\right)\\ & = 0 \text{ bit}\end{aligned}$$

Next, The gain in information from such a separating would be handled by using Equation 2.3:

Gain (Clump_Thickness) = $0.722 - 0$ = 0.722bits

Next, the predictable information necessity for each attribute is computed by using Equation 2.4:

Calculation of gain ratio for the attribute Clump_Thickness

$$\begin{aligned}\text{SplitInfo}_{\text{Clump\_Thickness}}(\text{D}) = {} & -\frac{1}{10}log_2\left(\frac{1}{10}\right)-\frac{2}{10}log_2\left(\frac{2}{10}\right)-\frac{1}{10}log_2\left(\frac{1}{10}\right)\\ & -\frac{2}{10}log_2\left(\frac{2}{10}\right)\ -\frac{1}{10}log_2(\frac{1}{10})-\frac{1}{10}log_2(\frac{1}{10})\\ & -\frac{1}{10}log_2(\frac{1}{10})-\frac{1}{10}log_2(\frac{1}{10})\\ & = 2.922\text{bits}\end{aligned}$$

Next, the gain ration is computed by using Equation 2.5:

Therefore, GainRatio(Clump_Thickness)= $\frac{0.722}{2.922}$ = 0.247 bits

The predictable information required to organize a tuple in D if the tuples are separated rendering to Uniformity_of_Cell_Size is

$$\text{Info}_{\text{Uniformity\_of\_Cell\_Size}}(\text{D}) = \frac{5}{10}\times\left(-\frac{5}{5}log_2\frac{5}{5}-\frac{0}{5}log_2\frac{0}{5}\right)+\frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)+\frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)+\frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)+\frac{1}{10}\times\left(-\frac{0}{1}log_2\frac{0}{1}-\frac{1}{1}log_2\frac{1}{1}\right)\ +\frac{1}{10}\times\left(-\frac{0}{1}log_2\frac{0}{1}-\frac{1}{1}log_2\frac{1}{1}\right)$$

$$= 0 \text{ bit}$$

The gain in material from such a separating would be

Gain (Uniformity_of_Cell_Size) = $0.722 - 0$ = 0.722bits

Calculation of gain ratio for the attribute Uniformity_of_Cell_Size

$$\text{SplitInfo}_{\text{Uniformity\_of\_Cell\_Size}}(\text{D}) = -\frac{5}{10}log_2\left(\frac{5}{10}\right)-\frac{1}{10}log_2\left(\frac{1}{10}\right)-\frac{1}{10}log_2\left(\frac{1}{10}\right)$$

$$-\frac{1}{10}log_2\left(\frac{1}{10}\right)-\frac{1}{10}log_2(\frac{1}{10})-\frac{1}{10}log_2(\frac{1}{10})$$

= 2.161bits

Therefore, GainRatio(Uniformity_of_Cell_Size)= $\frac{0.722}{2.161}$ = 0.334 bits

The predictable information required to organize a tuple in D if the tuples are separated rendering to **Uniformity_of_Cell_Shape** is

Info $_{\text{Uniformity\_of\_Cell\_Shape}}$ (D) = $\frac{5}{10}\times\left(-\frac{5}{5}log_2\frac{5}{5}-\frac{0}{5}log_2\frac{0}{5}\right)+\frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)+\frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)+\frac{1}{10}\times\left(-\frac{0}{1}log_2\frac{0}{1}-\frac{1}{1}log_2\frac{1}{1}\right)+\frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)+\frac{1}{10}\times\left(-\frac{0}{1}log_2\frac{0}{1}-\frac{1}{1}log_2\frac{1}{1}\right)$

= 0 bit

The gain in material from such a separating would be

Gain (Uniformity_of_Cell_Shape) = $0.722-0$ = 0.722bits

Calculation of gain ratio for the attribute Uniformity_of_Cell_Shape

SplitInfo $_{\text{Uniformity\_of\_Cell\_Shape}}$ (D) = $-\frac{5}{10}log_2\left(\frac{5}{10}\right)-\frac{1}{10}log_2\left(\frac{1}{10}\right)-\frac{1}{10}log_2\left(\frac{1}{10}\right)$

$$-\frac{1}{10}log_2\left(\frac{1}{10}\right)-\frac{1}{10}log_2(\frac{1}{10})-\frac{1}{10}log_2(\frac{1}{10})$$

= 2.161bits

Therefore, GainRatio(Uniformity_of_Cell_Shape)= $\frac{0.722}{2.161}$ = 0.334 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Marginal_Adhesion** is

Info $_{\text{Marginal\_Adhesion}}$ (D) = $\frac{6}{10}\times\left(-\frac{6}{6}log_2\frac{6}{6}-\frac{0}{6}log_2\frac{0}{6}\right)+\frac{2}{10}\times\left(-\frac{1}{2}log_2\frac{1}{2}-\frac{1}{2}log_2\frac{1}{2}\right)$

$$+\frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)+\frac{1}{10}\times\left(-\frac{0}{1}log_2\frac{0}{1}-\frac{1}{1}log_2\frac{1}{1}\right)$$

= 0.2 bit

The gain in material from such a separating would be

Gain (Marginal_Adhesion) = $0.722-0.2$ = 0.522bits

Calculation of gain ratio for the attribute Marginal_Adhesion

SplitInfo $_{\text{Marginal\_Adhesion}}$ (D) = $-\frac{6}{10}log_2\left(\frac{6}{10}\right)-\frac{2}{10}log_2\left(\frac{2}{10}\right)$

$$-\frac{1}{10}log_2\left(\frac{1}{10}\right)-\frac{1}{10}log_2\left(\frac{1}{10}\right)$$

= 1.571bits

Therefore, GainRatio(Marginal_Adhesion)= $\frac{0.522}{1.571}$ = 0.332 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Single_Epithelial_Cell_Size** is

$$\text{Info}_{\text{Single\_Epithelial\_Cell\_Size}}(D) = \frac{6}{10} \times \left(-\frac{6}{6} log_2 \frac{6}{6} - \frac{0}{6} log_2 \frac{0}{6}\right)$$

$$+\frac{1}{10} \times \left(-\frac{1}{1} log_2 \frac{1}{1} - \frac{0}{1} log_2 \frac{0}{1}\right)$$

$$+\frac{1}{10} \times \left(-\frac{0}{1} log_2 \frac{0}{1} - \frac{1}{1} log_2 \frac{1}{1}\right)$$

$$+\frac{2}{10} \times \left(-\frac{1}{2} log_2 \frac{1}{2} - \frac{1}{2} log_2 \frac{1}{2}\right)$$

= 0.2 bit

The gain in information from such a separating would be

Gain (Single_Epithelial_Cell_Size) = $0.722 - 0.2$ = 0.522bits

Calculation of gain ratio for the attribute Single_Epithelial_Cell_Size

$$\text{SplitInfo}_{\text{Single\_Epithelial\_Cell\_Size}}(D) = -\frac{6}{10} log_2 \left(\frac{6}{10}\right) - \frac{1}{10} log_2 \left(\frac{1}{10}\right)$$

$$-\frac{1}{10} log_2 \left(\frac{1}{10}\right) - \frac{2}{10} log_2 \left(\frac{2}{10}\right)$$

= 1.571bits

Therefore, GainRatio(Single_Epithelial_Cell_Size)= $\frac{0.522}{1.571}$ = 0.332 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Bare_Nuclei** is

$$\text{Info}_{\text{Bare\_Nuclei}}(D) = \frac{4}{10} \times \left(-\frac{4}{4} log_2 \frac{4}{4} - \frac{0}{4} log_2 \frac{0}{4}\right) + \frac{2}{10} \times \left(-\frac{1}{2} log_2 \frac{1}{2} - \frac{1}{2} log_2 \frac{1}{2}\right)$$

$$+\frac{1}{10} \times \left(-\frac{1}{1} log_2 \frac{1}{1} - \frac{0}{1} log_2 \frac{0}{1}\right) + \frac{3}{10} \times \left(-\frac{2}{3} log_2 \frac{2}{3} - \frac{1}{3} log_2 \frac{1}{3}\right)$$

= 0.475 bit

The gain in material from such a separating would be

Gain (Bare_Nuclei) = $0.722 - 0.475$ = 0.247bits

Calculation of gain ratio for the attribute Bare_Nuclei

$$\text{SplitInfo}_{\text{Bare\_Nuclei}}(D) = -\frac{4}{10} log_2 \left(\frac{4}{10}\right) - \frac{2}{10} log_2 \left(\frac{2}{10}\right) - \frac{1}{10} log_2 \left(\frac{1}{10}\right) - \frac{3}{10} log_2 \left(\frac{3}{10}\right)$$

= 1.846bits

Therefore, GainRatio(Bare_Nuclei)= $\frac{0.247}{1.846}$ = 0.134 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Bland_Chromatin** is

$$\text{Info}_{\text{Bland\_Chromatin}}(D) = \frac{1}{10} \times \left(-\frac{1}{1} log_2 \frac{1}{1} - \frac{0}{1} log_2 \frac{0}{1}\right) + \frac{1}{10} \times \left(-\frac{1}{1} log_2 \frac{1}{1} - \frac{0}{1} log_2 \frac{0}{1}\right)$$

$$+\frac{6}{10}\times\left(-\frac{6}{6}log_2\frac{6}{6}-\frac{0}{6}log_2\frac{0}{6}\right)+\frac{1}{10}\times\left(-\frac{0}{1}log_2\frac{0}{1}-\frac{1}{1}log_2\frac{1}{1}\right)$$

$$+\frac{1}{10}\times\left(-\frac{0}{1}log_2\frac{0}{1}-\frac{1}{1}log_2\frac{1}{1}\right)$$

= 0 bit

The gain in material from such a separating would be

Gain (Bland_Chromatin) = $0.722 - 0$= 0.722bits

Calculation of gain ratio for the attribute Bland_Chromatin

SplitInfo $_{\text{Bland\_Chromatin}}$ (D) = $-\frac{1}{10}log_2\left(\frac{1}{10}\right)-\frac{1}{10}log_2\left(\frac{1}{10}\right)-\frac{6}{10}log_2\left(\frac{6}{10}\right)$

$$-\frac{1}{10}log_2\left(\frac{1}{10}\right)-\frac{1}{10}log_2\left(\frac{1}{10}\right)$$

= 1.771bits

Therefore, GainRatio (Bland_Chromatin)= $\frac{0.722}{1.771}$ = 0.408 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Normal_Nucleoli** is

Info $_{\text{Normal\_Nucleoli}}$ (D) = $\frac{6}{10}\times\left(-\frac{6}{6}log_2\frac{6}{6}-\frac{0}{6}log_2\frac{0}{6}\right)+\frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)$

$$+\frac{3}{10}\times\left(-\frac{1}{3}log_2\frac{1}{3}-\frac{2}{3}log_2\frac{2}{3}\right)$$

= 0.276 bit

The gain in material from such a separating would be

Gain (Normal_Nucleoli) = $0.722 - 0.276$= 0.447 bits

Calculation of gain ratio for the attribute Normal_Nucleoli

SplitInfo $_{\text{Normal\_Nucleoli}}$ (D) = $-\frac{6}{10}log_2\left(\frac{6}{10}\right)-\frac{1}{10}log_2\left(\frac{1}{10}\right)-\frac{3}{10}log_2\left(\frac{3}{10}\right)$

= 1.296bits

Therefore, GainRatio(Normal_Nucleoli)= $\frac{0.447}{1.296}$ = 0.347 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Mitoses** is

Info $_{\text{Mitoses}}$ (D) = $\frac{9}{10}\times\left(-\frac{7}{9}log_2\frac{7}{9}-\frac{2}{9}log_2\frac{2}{9}\right)+\frac{1}{10}\times\left(-\frac{1}{1}log_2\frac{1}{1}-\frac{0}{1}log_2\frac{0}{1}\right)$

= 0.688 bit

The gain in material from such a separating would be

Gain (Mitoses) = $0.722 - 0.688$= 0.034 bits

Calculation of gain ratio for the attribute Mitoses

SplitInfo $_{\text{Mitoses}}$ (D) = $-\frac{9}{10}log_2\left(\frac{9}{10}\right)-\frac{1}{10}log_2\left(\frac{1}{10}\right)$

= 0.469bits

Therefore, GainRatio(Mitoses)= $\frac{0.034}{0.469}$ = 0.073 bits

As a result, the attribute “Bland Chromatin” with the maximum Gain Ratio is chosen to carry on the expansion of the classification. After computing the above steps, the system finds the “Bland Chromatin” has the greacheck information gain including the characteristics; it is particular as the root node. A node is made and considered with Class and braches are gown for each of the characteristic’s value. The decision tree generated by C4.5 for 10 tuples of Breast Cancer dataset is shown in Figure 3.1.

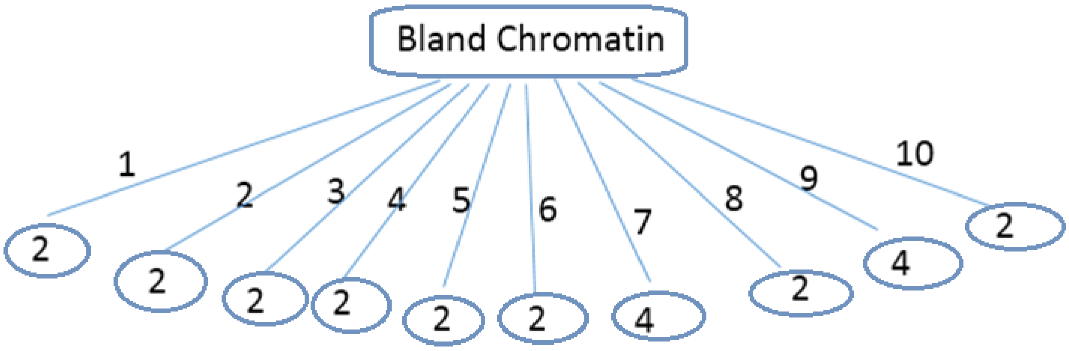


Figure 3.1 Decision Tree Generated by C4.5 for 10 tuples of Breast Cancer

3.5.2 Calculation of Weighted C4.5 Algorithm

Given a preparation dataset, the weighted C4.5 algorithm initializes the weights of each preparation instance, W_i by Naïve Bayes theorem. Estimating the prior probability for each class is computed by how often each class occurs in the Breast Cancer sample dataset Table 3.2.

Now the prior probability for each class and conditional probabilities for each attribute value are computed by using Equation 2.6:

Assigning weight value By Using Naïve Bayes Probability

P(class=2) = $\frac{8}{10}$

P(class=4) = $\frac{2}{10}$

Conditional probabilities for each attribute value:

P(ClumpThickness=“1”,class=“2”)= $\frac{1}{8}$

P(ClumpThickness=“1”, class=“4”)= $\frac{0}{2}$

P(ClumpThickness=“2”, class=“2”)= $\frac{2}{8}$

P(ClumpThickness=“2”, class=“4”)= $\frac{0}{2}$

P(ClumpThickness=“3”, class=“2”)= $\frac{1}{8}$

P(ClumpThickness=“3”, class=“4”)= $\frac{0}{2}$

P(ClumpThickness=“4”, class=“2”)= $\frac{2}{8}$

P(ClumpThickness=“4”, class=“4”)= $\frac{0}{2}$

P(ClumpThickness=“5”, class=“2”)= $\frac{1}{8}$

P(ClumpThickness=“5”, class=“4”)= $\frac{0}{2}$

P(ClumpThickness=“6”, class=“2”)= $\frac{1}{8}$

P(ClumpThickness=“6”, class=“4”)= $\frac{0}{2}$

P(ClumpThickness=“8”, class=“2”)= $\frac{0}{8}$

P(ClumpThickness=“8”, class=“4”)= $\frac{1}{2}$

P(ClumpThickness=“10”, class=“2”)= $\frac{0}{8}$

P(ClumpThickness=“10”, class=“4”)= $\frac{1}{2}$

P(Uniformity of Cell Size=“1”,class=“2”)= $\frac{5}{8}$

P(Uniformity of Cell Size =“1”, class=“4”)= $\frac{0}{2}$

P(Uniformity of Cell Size=“2”,class=“2”)= $\frac{1}{8}$

P(Uniformity of Cell Size =“2”, class=“4”)= $\frac{0}{2}$

P(Uniformity of Cell Size=“4”,class=“2”)= $\frac{1}{8}$

P(Uniformity of Cell Size =“4”, class=“4”)= $\frac{0}{2}$

P(Uniformity of Cell Size=“8”,class=“2”)= $\frac{1}{8}$

P(Uniformity of Cell Size =“8”, class=“4”)= $\frac{0}{2}$

P(Uniformity of Cell Size=“9”,class=“2”)= $\frac{0}{8}$

P(Uniformity of Cell Size =“9”, class=“4”)= $\frac{1}{2}$

P(Uniformity of Cell Size=“10”,class=“2”)= $\frac{0}{8}$

P(Uniformity of Cell Size =“10”, class=“4”)= $\frac{1}{2}$

For Id Number= “101”

P(class= “2”)= $\frac{8}{10} \times \frac{0}{8} \times \frac{0}{8} \times \frac{0}{8} \times \frac{1}{8} \times \frac{0}{8} \times \frac{1}{8} \times \frac{0}{8} \times \frac{1}{8} \times \frac{7}{8} = 0$

P(class= “4”)= $\frac{2}{10} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{2}{2} \times \frac{2}{2} = 0.0015625$

For Id Number= “102”

P(class= “2”)= $\frac{8}{10} \times \frac{1}{8} \times \frac{1}{8} \times \frac{1}{8} \times \frac{1}{8} \times \frac{1}{8} \times \frac{2}{8} \times \frac{6}{8} \times \frac{1}{8} \times \frac{7}{8} = 5.00679 \times 10^{-7}$

P(class= “4”)= $\frac{2}{10} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{2}{2} = 0$

For Id Number= “103”

P(class= “2”)= $\frac{8}{10} \times \frac{1}{8} \times \frac{5}{8} \times \frac{5}{8} \times \frac{6}{8} \times \frac{6}{8} \times \frac{1}{8} \times \frac{6}{8} \times \frac{6}{8} \times \frac{7}{8} = 0.0013518$

P(class= “4”)= $\frac{2}{10} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{1}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{2}{2} = 0$

For Id Number= “104”

P(class= “2”)= $\frac{8}{10} \times \frac{1}{8} \times \frac{1}{8} \times \frac{1}{8} \times \frac{6}{8} \times \frac{1}{8} \times \frac{1}{8} \times \frac{6}{8} \times \frac{1}{8} \times \frac{7}{8} = 1.50203 \times 10^{-6}$

P(class= “4”)= $\frac{2}{10} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{2}{2} \times \frac{2}{2} = 0$

For Id Number= “105”

P(class= “2”)= $\frac{8}{10} \times \frac{2}{8} \times \frac{5}{8} \times \frac{5}{8} \times \frac{1}{8} \times \frac{1}{8} \times \frac{4}{8} \times \frac{6}{8} \times \frac{6}{8} \times \frac{7}{8} = 0.0018024$

P(class= “4”)= $\frac{2}{10} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{1}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{2}{2} = 0$

For Id Number= “106”

P(class= “2”)= $\frac{8}{10} \times \frac{0}{8} \times \frac{0}{8} \times \frac{0}{8} \times \frac{0}{8} \times \frac{1}{8} \times \frac{2}{8} \times \frac{0}{8} \times \frac{1}{8} \times \frac{7}{8} = 0$

P(class= “4”)= $\frac{2}{10} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{2}{2} \times \frac{2}{2} = 0.0015625$

For Id Number= “107”

P(class= “2”)= $\frac{8}{10} \times \frac{1}{8} \times \frac{5}{8} \times \frac{5}{8} \times \frac{6}{8} \times \frac{6}{8} \times \frac{2}{8} \times \frac{6}{8} \times \frac{6}{8} \times \frac{7}{8} = 0.0027037$

P(class= “4”)= $\frac{2}{10} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{1}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{2}{2}$ = 0

For Id Number= “108”

P(class= “2”)= $\frac{8}{10} \times \frac{2}{8} \times \frac{5}{8} \times \frac{1}{8} \times \frac{6}{8} \times \frac{6}{8} \times \frac{4}{8} \times \frac{6}{8} \times \frac{6}{8} \times \frac{7}{8}$ = 0.0021629

P(class= “4”)= $\frac{2}{10} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2}$ = 0

For Id Number= “109”

P(class= “2”)= $\frac{8}{10} \times \frac{2}{8} \times \frac{5}{8} \times \frac{5}{8} \times \frac{6}{8} \times \frac{6}{8} \times \frac{4}{8} \times \frac{1}{8} \times \frac{6}{8} \times \frac{1}{8}$ = 2.57492×10^{-4}

P(class= “4”)= $\frac{2}{10} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2}$ = 0

For Id Number= “110”

P(class= “2”)= $\frac{8}{10} \times \frac{2}{8} \times \frac{1}{8} \times \frac{5}{8} \times \frac{6}{8} \times \frac{6}{8} \times \frac{4}{8} \times \frac{1}{8} \times \frac{6}{8} \times \frac{7}{8}$ = 3.60488×10^{-4}

P(class= “4”)= $\frac{2}{10} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{0}{2} \times \frac{2}{2}$ = 0

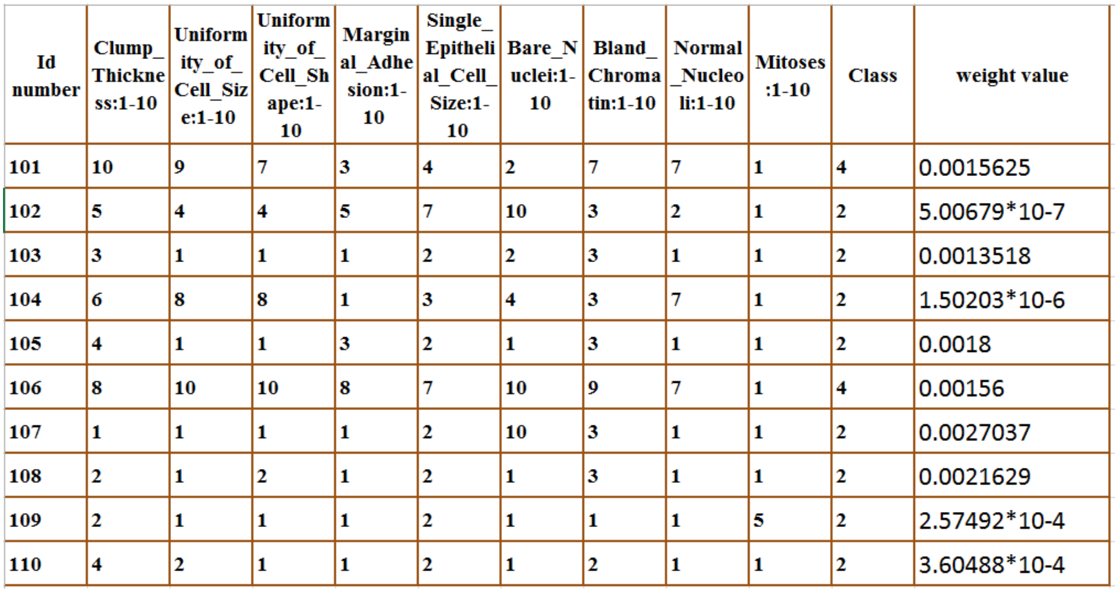
Now calculate the posterior probabilities for preparation instances and assign the weights of each attribute instance with highest posterior probability using sample dataset in Table 3.1. Table 3.2 describes the assigned weights of preparation instances in preparation dataset.

Table 3.2 Assigned weights in training sample data

Id Number	Class 2	Class 4	weight value
101	0	0.0015625	0.0015625
102	5.00679*10-7	0	5.00679*10-7
103	0.0013518	0	0.0013518
104	1.50203*10-6	0	1.50203*10-6
105	0.0018	0	0.0018
106	0	0.00156	0.00156
107	0.0027037	0	0.0027037
108	0.0021629	0	0.0021629
109	2.57492*10-4	0	2.57492*10-4
110	3.60488*10-4	0	3.60488*10-4

Next, to calculate the information gain for each attribute using weights, the predictable material required to categorize a tuple in preparation usual is computed using Equation 2.1. Table 3.3 represents 10 sample records of the Breast Cancer Dataset.

Table 3.3 Sample records of the Breast Cancer Dataset



Id number	Clump_Thickness:1-10	Uniformity_of_Cell_Size:1-10	Uniformity_of_Cell_Shape:1-10	Marginal_Adhesion:1-10	Single_Epithelial_Cell_Size:1-10	Bare_Nuclei:1-10	Bland_Chromatin:1-10	Normal_Nucleoli:1-10	Mitoses:1-10	Class	weight value
101	10	9	7	3	4	2	7	7	1	4	0.0015625
102	5	4	4	5	7	10	3	2	1	2	5.00679*10-7
103	3	1	1	1	2	2	3	1	1	2	0.0013518
104	6	8	8	1	3	4	3	7	1	2	1.50203*10-6
105	4	1	1	3	2	1	3	1	1	2	0.0018
106	8	10	10	8	7	10	9	7	1	4	0.00156
107	1	1	1	1	2	10	3	1	1	2	0.0027037
108	2	1	2	1	2	1	3	1	1	2	0.0021629
109	2	1	1	1	2	1	1	1	5	2	2.57492*10-4
110	4	2	1	1	2	1	2	1	1	2	3.60488*10-4

To calculate the expected information required to classify a tuple in Dataset D:

$$\text{Info(D)} = \quad -\frac{8.640862\times10^{-3}}{0.011765862}log_2\left(\frac{8.640862\times10^{-3}}{0.011765862}\right)-\frac{0.003125}{0.011765862}log_2(\frac{0.003125}{0.011765862})$$

= 0.8350728bits

Next, the expected information necessity for each attribute is computed by using Equation 2.2:

The predictable material required to organize a tuple in D if the tuples are separated according to Clump_Thickness is

$\text{Info}_{\text{Clump\_Thickness}}(\text{D})$

$$=\frac{0.0027037}{0.011765862}\times\left(-\frac{0.0027037}{0.0027037}log_2\frac{0.0027037}{0.0027037}-\frac{0}{0.0027037}log_2\frac{0}{0.0027037}\right)+\frac{2.420392\times10^{-3}}{0.011765862}\times$$
$$\left(-\frac{2.420392\times10^{-3}}{2.420392\times10^{-3}}log_2\frac{2.420392\times10^{-3}}{2.420392\times10^{-3}}-\frac{0}{2.420392\times10^{-3}}log_2\frac{0}{2.420392\times10^{-3}}\right)+\frac{0.0013518}{0.011765862}\times$$
$$\left(-\frac{0.0013518}{0.0013518}log_2\frac{0.0013518}{0.0013518}-\frac{0}{0.0013518}log_2\frac{0}{0.0013518}\right)+\frac{2.162888\times10^{-3}}{0.011765862}\times$$
$$\left(-\frac{2.162888\times10^{-3}}{2.162888\times10^{-3}}log_2\frac{2.162888\times10^{-3}}{2.162888\times10^{-3}}-\frac{0}{2.162888\times10^{-3}}log_2\frac{0}{2.162888\times10^{-3}}\right)$$
$$+\frac{5.00679\times10^{-7}}{0.011765862}\times\left(-\frac{5.00679\times10^{-7}}{5.00679\times10^{-7}}log_2\frac{5.00679\times10^{-7}}{5.00679\times10^{-7}}-\frac{0}{15.00679\times10^{-7}}log_2\frac{0}{5.00679\times10^{-7}}\right)$$

$$+\frac{1.50203\times10^{-6}}{0.011765862}\times\left(-\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}}log_2\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}}-\frac{0}{1.50203\times10^{-6}}log_2\frac{0}{1.50203\times10^{-6}}\right)$$
$$+\frac{0.0015625}{0.011765862}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.00156251}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$
$$+\frac{0.0015625}{0.011765862}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.00156251}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$
= 0 bit

Next, the predictable material necessity for each characteristic is computed by using Equation 2.3:

The gain in material from such a separating would be

Gain (Clump_Thickness) = $0.8350728 - 0$ = 0.8350728bits

Next, the predictable material necessity for each attribute is processed by using Equation 2.4:

Calculation of gain ratio for the attribute Clump_Thickness

SplitInfo$_{\text{Clump\_Thickness}}$(D)= $-\frac{0.0027037}{0.011765862}log_2\left(\frac{0.0027037}{0.011765862}\right)$
$$-\frac{2.420392\times10^{-3}}{0.011765862}log_2\left(\frac{.420392\times10^{-3}}{0.011765862}\right)-\frac{0.001351}{0.011765862}log_2\left(\frac{0.001351}{0.011765862}\right)$$
$$-\frac{2.162888\times10^{-3}}{0.011765862}log_2\left(\frac{2.162888\times10^{-3}}{0.011765862}\right)-\frac{5.00679\times10^{-7}}{0.011765862}log_2\left(\frac{5.00679\times10^{-7}}{0.011765862}\right)$$
$$-\frac{1.50203\times10^{-6}}{0.011765862}log_2\left(\frac{1.50203\times10^{-6}}{0.011765862}\right)-\frac{0.0015625}{0.011765862}log_2\left(\frac{0.0015625}{0.011765862}\right)$$
$$-\frac{0.0015625}{0.011765862}log_2(\frac{0.0015625}{0.011765862})$$
= 2.7977228bits

Next, the predictable material necessity for each attribute is computed by using Equation 2.5:

Therefore, GainRatio(Clump_Thickness)= $\frac{0.8350728}{2.7977228}$ = 0.0298484 bits

The predictable material required to organize a tuple in D if the tuples are separated according to Uniformity_of_Cell_Size is

Info $_{\text{Uniformity\_of\_Cell\_Size}}$ (D)

$$=\frac{0.0082783}{0.011765862}\times\left(-\frac{0.0082783}{0.0082783}log_2\frac{0.0082783}{0.0082783}-\frac{0}{0.0082783}log_2\frac{0}{0.0082783}\right)+\frac{3.604888\times10^{-4}}{0.011765862}\times$$
$$\left(-\frac{3.604888\times10^{-4}}{3.604888\times10^{-4}}log_2\frac{3.604888\times10^{-4}}{3.604888\times10^{-4}}-\frac{0}{3.604888\times10^{-4}}log_2\frac{0}{13.604888\times10^{-4}}\right)$$
$$+\frac{5.00679\times10^{-7}}{0.011765862}\times\left(-\frac{5.00679\times10^{-7}}{5.00679\times10^{-7}}log_2\frac{5.00679\times10^{-7}}{5.00679\times10^{-7}}-\frac{0}{5.00679\times10^{-7}}log_2\frac{0}{5.00679\times10^{-7}}\right)+$$
$$\frac{1.502037\times10^{-6}}{0.011765862}\times\left(-\frac{1.502037\times10^{-6}}{1.502037\times10^{-6}}log_2\frac{1.502037\times10^{-6}}{1.502037\times10^{-6}}-\frac{0}{1.502037\times10^{-6}}log_2\frac{0}{1.502037\times10^{-6}}\right)$$

$$+\frac{0.0015625}{0.011765862}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$
$$+\frac{0.0015625}{0.011765862}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$
= 0 bit

The gain in material from such a separating could be

Gain (Uniformity_of_Cell_Size) = $0.8350728 - 0$ = 0.8350728bits

Computation of gain ratio for the attribute Uniformity_of_Cell_Size

SplitInfo $_{\text{Uniformity\_of\_Cell\_Size}}$ (D)

$$=-\frac{0.0082783}{0.011765862}log_2\left(\frac{0.0082783}{0.011765862}\right)-\frac{3.604888\times10^{-4}}{0.011765862}log_2\left(\frac{3.604888\times10^{-4}}{0.011765862}\right)$$
$$-\frac{5.00679\times10^{-7}}{0.011765862}log_2\left(\frac{5.00679\times10^{-7}}{0.011765862}\right)-\frac{1.502037\times10^{-6}}{0.011765862}log_2\left(\frac{1.502037\times10^{-6}}{0.011765862}\right)$$
$$-\frac{0.0015625}{0.011765862}log_2(\frac{0.0015625}{0.011765862})-\frac{0.0015625}{0.011765862}log_2(\frac{0.0015625}{0.011765862})$$
= 1.9381352bits

Therefore, GainRatio(Uniformity_of_Cell_Size)= $\frac{0.8350728}{1.9381352}$ = 0.4308668bits

The predictable material required to organize a tuple in D if the tuples are separated separated according to **Uniformity_of_Cell_Shape** is

Info $_{\text{Uniformity\_of\_Cell\_Shape}}$ (D)

$$=\frac{0.0064759}{0.011765862}\times\left(-\frac{0.0064759}{0.0064759}log_2\frac{0.0064759}{0.0064759}-\frac{0}{0.0064759}log_2\frac{0}{0.0064759}\right)+\frac{0.0021629}{0.011765862}\times$$
$$\left(-\frac{0.0021629}{0.0021629}log_2\frac{0.0021629}{0.0021629}-\frac{0}{0.0021629}log_2\frac{0}{0.0021629}\right)+\frac{5.00679\times10^{-7}}{0.011765862}\times$$
$$\left(-\frac{5.00679\times10^{-7}}{5.00679\times10^{-7}}log_2\frac{5.00679\times10^{-7}}{5.00679\times10^{-7}}-\frac{0}{5.00679\times10^{-7}}log_2\frac{0}{5.00679\times10^{-7}}\right)+\frac{0.0015625}{0.011765862}\times$$
$$\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)+\frac{1.502037\times10^{-6}}{0.011765862}\times$$
$$\left(-\frac{1.502037\times10^{-6}}{1.502037\times10^{-6}}log_2\frac{1.502037\times10^{-6}}{1.502037\times10^{-6}}-\frac{0}{1.502037\times10^{-6}}log_2\frac{0}{1.502037\times10^{-6}}\right)+\frac{0.0015625}{0.011765862}\times$$
$$\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{10.0015625}\right)$$
= 0 bit

The gain in information from such a separating would be

Gain (Uniformity_of_Cell_Shape) = $0.8350728 - 0$ = 0.8350728bits

Calculation of gain ratio for the attribute Uniformity_of_Cell_Shape

SplitInfo $_{\text{Uniformity\_of\_Cell\_Shape}}$ (D)

$$=-\frac{0.0064759}{0.011765862}log_2\left(\frac{0.0064759}{0.011765862}\right)-\frac{0.0021629}{0.011765862}log_2\left(\frac{0.0021629}{0.011765862}\right)$$
$$-\frac{5.00679\times10^{-7}}{0.011765862}log_2\left(\frac{5.00679\times10^{-7}}{0.011765862}\right)-\frac{0.0015625}{0.011765862}log_2\left(\frac{0.0015625}{0.011765862}\right)$$

$$-\frac{1.502037\times10^{-6}}{0.011765862}log_2(\frac{1.502037\times10^{-6}1}{0.011765862})-\frac{0.0015625}{0.011765862}log_2(\frac{0.0015625}{0.011765862})$$

= 1.9581201bits

Therefore, GainRatio(Uniformity_of_Cell_Shape)= $\frac{0.8350728}{1.9581201}$ = 0.4264693 bits

The predictable material required to organize a tuple in D if the tuples are separated separated according to **Marginal_Adhesion** is

Info $_{\text{Marginal\_Adhesion}}$ (D)

$$=\frac{0.0068379}{0.011765862}\times\left(-\frac{0.0068379}{0.0068379}log_2\frac{0.0068379}{0.0068379}-\frac{0}{0.0068379}log_2\frac{0}{0.0068379}\right)$$

$$+\frac{0.00336494}{0.011765862}\times\left(-\frac{0.0018024}{0.00336494}log_2\frac{0.0018024}{0.00336494}-\frac{0.0015625}{0.00336494}log_2\frac{0.0015625}{0.00336494}\right)$$

$$+\frac{5.00679\times10^{-7}}{0.011765862}\times\left(-\frac{5.00679\times10^{-7}}{5.00679\times10^{-7}}log_2\frac{5.00679\times10^{-7}}{5.00679\times10^{-7}}-\frac{0}{5.00679\times10^{-7}}log_2\frac{0}{5.00679\times10^{-7}}\right)$$

$$+\frac{0.0015625}{0.011765862}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$

= 0.2672727 bit

The gain in material from such a separating could be

Gain (Marginal_Adhesion) = $0.8350728-0.2672727$ = 0.567805bits

Calculation of gain ratio for the attribute Marginal_Adhesion

SplitInfo $_{\text{Marginal\_Adhesion}}$ (D)

$$=-\frac{0.0068379}{0.011765862}log_2\left(\frac{0.0068379}{0.011765862}\right)-\frac{0.00336494}{0.011765862}log_2\left(\frac{0.00336494}{0.011765862}\right)$$

$$-\frac{5.00679\times10^{-7}}{0.011765862}log_2\left(\frac{5.00679\times10^{-7}}{0.011765862}\right)-\frac{0.0015625}{0.011765862}log_2\left(\frac{0.0015625}{0.011765862}\right)$$

= 1.344931bits

Therefore, GainRatio(Marginal_Adhesion)= $\frac{0.567805}{1.344931}$ = 0.4221815 bits

The predictable material required to organize a tuple in D if the tuples are separated separated according to **Single_Epithelial_Cell_Size** is

Info $_{\text{Single\_Epithelial\_Cell\_Size}}$ (D)

$$=\frac{0.0086388}{0.011765862}\times\left(-\frac{0.0086388}{0.0086388}log_2\frac{0.0086388}{0.0086388}-\frac{0}{0.0086388}log_2\frac{0}{0.0086388}\right)$$

$$+\frac{1.502037}{0.011765862}\times\left(-\frac{1.502037}{1.502037}log_2\frac{1.502037}{1.502037}-\frac{0}{1.502037}log_2\frac{0}{1.502037}\right)$$

$$+\frac{0.00156251}{0.011765862}\times\left(-\frac{0}{0.00156251}log_2\frac{0}{0.00156251}-\frac{0.00156251}{0.00156251}log_2\frac{0.00156251}{0.00156251}\right)$$

$$+\frac{1.56300067\times10^{-3}}{0.011765862}\times$$

$$\left(-\frac{5.0067\times10^{-7}}{1.56300067\times10^{-3}}log_2\frac{5.0067\times10^{-7}}{1.56300067\times10^{-3}}-\frac{0.0015625}{1.56300067\times10^{-3}}log_2\frac{0.0015625}{1.56300067\times10^{-3}}\right)$$

= 0.0011103 bit

The gain in material from such a separating could be

Gain (Single_Epithelial_Cell_Size) = $0.8350728 - 0.0011103$ = 0.8339678bits

Calculation of gain ratio for the attribute Single_Epithelial_Cell_Size

SplitInfo $_{\text{Single\_Epithelial\_Cell\_Size}}$ (D)

$$= -\frac{0.0086388}{0.011765862} log_2\left(\frac{0.0086388}{0.011765862}\right) - \frac{1.502037}{0.011765862} log_2\left(\frac{1.502037}{0.011765862}\right)$$

$$-\frac{0.00156251}{0.011765862} log_2\left(\frac{0.00156251}{0.011765862}\right) - \frac{1.56300067\times10^{-3}}{0.011765862} log_2\left(\frac{1.56300067\times10^{-3}}{0.011765862}\right)$$

= 1.5808228bits

Therefore, GainRatio(Single_Epithelial_Cell_Size)= $\frac{0.8339678}{1.5808228}$ = 0.5275530 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Bare_Nuclei** is

Info $_{\text{Bare\_Nuclei}}$ (D)

$$= \frac{0.0045833}{0.011765862} \times \left(-\frac{0.0045833}{0.0045833} log_2 \frac{0.0045833}{0.0045833} - \frac{0}{0.0045833} log_2 \frac{0}{0.0045833}\right)$$

$$+\frac{0.00291433}{0.011765862} \times \left(-\frac{0.0013518}{0.00291433} log_2 \frac{0.0013518}{0.00291433} - \frac{0.0015625}{0.00291433} log_2 \frac{0.0015625}{0.00291433}\right)$$

$$+\frac{1.50203\times10^{-6}}{0.011765862} \times \left(-\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}} log_2 \frac{1.50203\times10^{-6}}{1.50203\times10^{-6}} - \frac{0}{1.50203\times10^{-6}} log_2 \frac{0}{1.50203\times10^{-6}}\right)$$

$$+\frac{0.0042666}{0.011765862} \times \left(-\frac{0.0027041}{0.0042666} log_2 \frac{0.0027041}{0.0042666} - \frac{0.0015625}{0.0042666} log_2 \frac{0.0015625}{0.0042666}\right)$$

= 0.5865113 bit

The gain in material from such a separating could be

Gain (Bare_Nuclei) = $0.8350728 - 0.5865113$ = 0.2485668bits

Calculation of gain ratio for the attribute Bare_Nuclei

SplitInfo $_{\text{Bare\_Nuclei}}$ (D)

$$=-\frac{0.0045833}{0.011765862} log_2\left(\frac{0.0045833}{0.011765862}\right) - \frac{0.00291433}{0.011765862} log_2\left(\frac{0.00291433}{0.011765862}\right)$$

$$-\frac{1.502037\times10^{-6}}{0.011765862} log_2\left(\frac{1.502037\times10^{-6}}{0.011765862}\right) - \frac{0.0042666}{0.011765862} log_2\left(\frac{0.0042666}{0.011765862}\right)$$

= 1.5456027bits

Therefore, GainRatio(Bare_Nuclei)= $\frac{0.2485668}{1.5456027}$ = 0.1608219 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Bland_Chromatin** is

Info $_{\text{Bland\_Chromatin}}$ (D)

$$=\frac{2.574920\times10^{-4}}{0.011765862}\times$$
$$\left(-\frac{2.574920\times10^{-4}}{2.574920\times10^{-4}}log_2\frac{2.574920\times10^{-4}}{2.574920\times10^{-4}}-\frac{0}{2.574920\times10^{-4}}log_2\frac{0}{2.574920\times10^{-4}}\right)+$$
$$\frac{3.604888\times10^{-4}}{0.011765862}\times\left(-\frac{3.604888\times10^{-4}}{3.604888\times10^{-4}}log_2\frac{3.604888\times10^{-4}}{3.604888\times10^{-4}}-\frac{0}{3.604888\times10^{-4}}log_2\frac{0}{3.604888\times10^{-4}}\right)$$
$$+\frac{0.00802288}{0.011765862}\times\left(-\frac{0.00802288}{0.00802288}log_2\frac{0.00802288}{0.00802288}-\frac{0}{0.00802288}log_2\frac{0}{0.00802288}\right)$$
$$+\frac{0.0015625}{0.011765862}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$
$$+\frac{0.0015625}{0.011765862}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$

= 0 bit

The gain in material from such a separating could be

Gain (Bland_Chromatin) = $0.8350728-0$= 0.8350728bits

Calculation of gain ratio for the attribute Bland_Chromatin

SplitInfo $_{\text{Bland\_Chromatin}}$ (D) =$-\frac{2.574920\times10^{-4}}{0.011765862}log_2\left(\frac{2.574920\times10^{-4}}{0.011765862}\right)$

$$-\frac{3.604888\times10^{-4}}{0.011765862}log_2\left(\frac{3.604888\times10^{-4}}{0.011765862}\right)-\frac{0.00802288}{0.011765862}log_2\left(\frac{0.00802288}{0.011765862}\right)$$
$$-\frac{0.0015625}{0.011765862}log_2\left(\frac{0.0015625}{0.011765862}\right)-\frac{0.0015625}{0.011765862}log_2\left(\frac{0.0015625}{0.011765862}\right)$$

= 1.9552543bits

Therefore, GainRatio (Bland_Chromatin)= $\frac{0.8350728}{1.9552543}$ = 0.4270944 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Normal_Nucleoli** is

Info $_{\text{Normal\_Nucleoli}}$ (D)

$$=\frac{0.0086388}{0.011765862}\times\left(-\frac{0.0086388}{0.0086388}log_2\frac{0.0086388}{0.0086388}-\frac{0}{0.0086388}log_2\frac{0}{0.0086388}\right)+\frac{5.006790\times10^{-7}}{0.011765862}\times$$
$$\left(-\frac{5.006790\times10^{-7}}{5.006790\times10^{-7}}log_2\frac{5.006790\times10^{-7}}{5.006790\times10^{-7}}-\frac{0}{5.006790\times10^{-7}}log_2\frac{0}{5.006790\times10^{-7}}\right)$$
$$+\frac{0.0031265}{0.011765862}\times\left(-\frac{1.502037\times10^{-6}}{0.0031265}log_2\frac{1.502037\times10^{-6}}{0.0031265}-\frac{0.003125}{0.0031265}log_2\frac{0.003125}{0.0031265}\right)$$

= 0.0015906 bit

The gain in material from such as separating would be

Gain (Normal_Nucleoli) = $0.8350728-0.0015906$= 0.8334875 bits

Calculation of gain ratio for the attribute Normal_Nucleoli

SplitInfo $_{\text{Normal\_Nucleoli}}$ (D)

$$=-\frac{0.0086388}{0.011765862}log_2\left(\frac{0.0086388}{0.011765862}\right)-\frac{5.006790\times10^{-7}}{0.011765862}log_2\left(\frac{5.006790\times10^{-7}}{0.011765862}\right)-\frac{0.0031265}{0.011765862}log_2\left(\frac{0.0031265}{0.011765862}\right)$$

= 0.835696 bits

Therefore, GainRatio(Normal_Nucleoli)= $\frac{0.8334875}{0.835696}$ = 0.9973572 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Mitoses** is

Info $_{\text{Mitoses}}$ (D)

$$=\frac{0.0115083}{0.011765862}\times\left(-\frac{0.00838336}{0.0115083}log_2\frac{0.00838336}{0.0115083}-\frac{0.003125}{0.0115083}log_2\frac{0.003125}{0.0115083}\right)$$

$$+\frac{2.5749206}{0.011765862}\times\left(-\frac{2.5749206}{2.5749206}log_2\frac{2.5749206}{2.5749206}-\frac{0}{2.5749206}log_2\frac{0}{2.5749206}\right)$$

= 0.6195906 bit

The gain in material from such a separating could be

Gain (Mitoses) = $0.8350728 - 0.6195906$= 0.2154875 bits

Calculation of gain ratio for the attribute Mitoses

SplitInfo $_{\text{Mitoses}}$ (D) = $-\frac{0.0115083}{0.011765862}log_2\left(\frac{0.0115083}{0.011765862}\right)-\frac{2.5749206}{0.011765862}log_2\left(\frac{2.5749206}{0.011765862}\right)$

= 0.6837673bits

Therefore, GainRatio(Mitoses)= $\frac{0.2154875}{0.6837673}$ = 0.3151475 bits

The gain value of Normal Nucleoli is maximum than other attributes, so root of decision tree will be Normal Nucleoli.

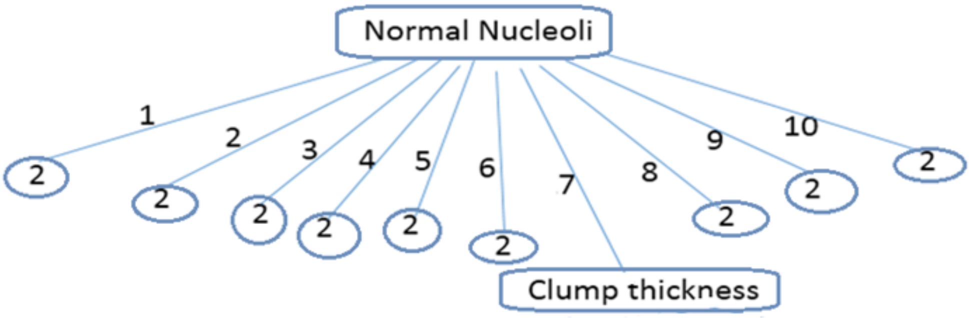


Figure 3.2 Root node of the tree

As a result, the attribute "Normal Nucleoli" with the maximum Gain Ratio is selected to carry on the expansion of the classification. For the next expressions, the attribute "Normal Nucleoli" is deleted from the resulting partition and the Entropy, Information Gain, SplitInfo and Gain Ratio are computed to select the splitting criterion for the resulting partition. This process will continue if all tuples in the resulting partition do not belong to one class. There is no left of attribute or tuple in the resulting partition, the process will be terminated.

Table 3.4 Subset of data Normal_Nucleoli=7

Id number	Clump_Thickness:1-10	Uniformity_of_Cell_Size:1-10	Uniformity_of_Cell_Shape:1-10	Marginal_Adhesion:1-10	Single_Epithelial_Cell_Size:1-10	Bare_Nuclei:1-10	Bland_Chromatin:1-10	Mitoses:1-10	Class	weight value
101	10	9	7	3	4	2	7	1	4	0.0015625
104	6	8	8	1	3	4	3	1	2	1.50203*10-6
106	8	10	10	8	7	10	9	1	4	0.00156

To calculate the expected information required to classify a tuple in Dataset D:

Info(D) = $-\frac{1.50203\times10^{-6}}{0.0031265}log_2\left(\frac{1.50203\times10^{-6}}{0.0031265}\right)-\frac{0.003125}{0.0031265}log_2\left(\frac{0.003125}{0.0031265}\right)$

= 0.0059888bits

The predictable material required to organize a tuple in D if the tuples are partitioned according to Clump_Thickness is

Info$_{\text{Clump\_Thickness}}$(D)

$=\frac{1.50203\times10^{-6}}{0.0031265}\times\left(-\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}}log_2\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}}-\frac{0}{1.50203\times10^{-6}}log_2\frac{0}{1.50203\times10^{-6}}\right)$

$+\frac{0.0015625}{0.0031265}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$

$+\frac{0.0015625}{0.0031265}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$

= 0 bit

The gain in material from such a separating could be

Gain (Clump_Thickness) = 0.0059888 − 0 = 0.0059888bits

Calculation of gain ratio for the attribute Clump_Thickness

SplitInfo$_{\text{Clump\_Thickness}}$(D)= $-\frac{1.50203\times10^{-6}}{0.0031265}log_2\left(\frac{1.50203\times10^{-3}}{0.0031265}\right)$

$-\frac{0.0015625}{0.00312650.011765862}log_2\left(\frac{0.0015625}{0.0031265}\right)-\frac{0.0015625}{0.0031265}log_2\left(\frac{0.0015625}{0.0031265}\right)$

= 0.5060950bits

Therefore, GainRatio(Clump_Thickness)= $\frac{0.0059888}{0.5060950}$ = 0.011833 bits

The predictable material required to organize a tuple in D if the tuples are partitioned according to Uniformity_of_Cell_Size is

Info $_{\text{Uniformity\_of\_Cell\_Size}}$ (D)

$$=\frac{1.50203\times10^{-6}}{0.0031265}\times\left(-\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}}log_2\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}}-\frac{0}{1.50203\times10^{-6}}log_2\frac{0}{1.50203\times10^{-6}}\right)$$

$$+\frac{0.0015625}{0.0031265}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$

$$+\frac{0.0015625}{0.0031265}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$

= 0 bit

The gain in material from such a separating could be

Gain (Uniformity_of_Cell_Size) = $0.0059888-0$ = 0.0059888bits

Calculation of gain ratio for the attribute Uniformity_of_Cell_Size

SplitInfo $_{\text{Uniformity\_of\_Cell\_Size}}$ (D)

$$=-\frac{1.50203\times10^{-6}}{0.0031265}log_2\left(\frac{1.50203\times10^{-3}}{0.0031265}\right)$$

$$-\frac{0.0015625}{0.00312650.011765862}log_2\left(\frac{0.0015625}{0.0031265}\right)-\frac{0.0015625}{0.0031265}log_2\left(\frac{0.0015625}{0.0031265}\right)$$

= 0.5060950bits

Therefore, GainRatio(Uniformity_of_Cell_Size)= $\frac{0.0059888}{0.5060950}$ = 0.011833 bits

The predictable material required to organize a tuple in D if the tuples are partitioned according to **Uniformity_of_Cell_Shape** is

Info $_{\text{Uniformity\_of\_Cell\_Shape}}$ (D)

$$=\frac{0.0015625}{0.0031265}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$

$$+\frac{1.50203\times10^{-6}}{0.0031265}\times\left(-\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}}log_2\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}}-\frac{0}{1.50203\times10^{-6}}log_2\frac{0}{1.50203\times10^{-6}}\right)$$

$$+\frac{0.0015625}{0.0031265}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$

= 0 bit

The gain in material from such a separating could be

Gain (Uniformity_of_Cell_Shape) = $0.0059888-0$ = 0.0059888bits

Calculation of gain ratio for the attribute Uniformity_of_Cell_Shape

SplitInfo $_{\text{Uniformity\_of\_Cell\_Shape}}$ (D)

$$=-\frac{0.0015625}{0.00312650.011765862}log_2\left(\frac{0.0015625}{0.0031265}\right)-\frac{1.50203\times10^{-6}}{0.0031265}log_2\left(\frac{1.50203\times10^{-3}}{0.0031265}\right)$$

$$-\frac{0.0015625}{0.0031265}log_2\left(\frac{0.0015625}{0.0031265}\right)$$

= 0.5060950bits

Therefore, GainRatio(Uniformity_of_Cell_Shape)= $\frac{0.0059888}{0.5060950}$ = 0.011833 bits

The predictable material required to organize a tuple in D if the tuples are partitioned according to **Marginal_Adhesion** is

Info $_{\text{Marginal\_Adhesion}}$ (D)

$$=\frac{1.50203\times10^{-6}}{0.0031265}\times\left(-\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}}log_2\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}}-\frac{0}{1.50203\times10^{-6}}log_2\frac{0}{1.50203\times10^{-6}}\right)$$

$$+\frac{0.0015625}{0.0031265}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$

$$+\frac{0.0015625}{0.0031265}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$

= 0 bit

The gain in material from such a separating could be

Gain (Marginal_Adhesion) = $0.0059888 - 0$ = 0.0059888bits

Computation of gain ratio for the attribute Marginal_Adhesion

SplitInfo $_{\text{Marginal\_Adhesion}}$ (D)

$$=-\frac{1.50203\times10^{-6}}{0.0031265}log_2\left(\frac{1.50203\times10^{-3}}{0.0031265}\right)$$

$$-\frac{0.0015625}{0.00312650.011765862}log_2\left(\frac{0.0015625}{0.0031265}\right)-\frac{0.0015625}{0.0031265}log_2\left(\frac{0.0015625}{0.0031265}\right)$$

= 0.5060950bits

Therefore, GainRatio(Marginal_Adhesion)= $\frac{0.0059888}{0.5060950}$ = 0.011833 bits

The predictable material required to organize a tuple in D if the tuples are according to **Single_Epithelial_Cell_Size** is

Info $_{\text{Single\_Epithelial\_Cell\_Size}}$ (D)

$$=\frac{1.50203\times10^{-6}}{0.0031265}\times\left(-\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}}log_2\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}}-\frac{0}{1.50203\times10^{-6}}log_2\frac{0}{1.50203\times10^{-6}}\right)$$

$$+\frac{0.0015625}{0.0031265}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$

$$+\frac{0.0015625}{0.0031265}\times\left(-\frac{0}{0.0015625}log_2\frac{0}{0.0015625}-\frac{0.0015625}{0.0015625}log_2\frac{0.0015625}{0.0015625}\right)$$

= 0 bit

The gain in material from such a separating could be

Gain (Single_Epithelial_Cell_Size) = $0.0059888 - 0$ = 0.0059888bits

Calculation of gain ratio for the attribute Single_Epithelial_Cell_Size

SplitInfo $_{\text{Single\_Epithelial\_Cell\_Size}}$ (D)

$$=-\frac{1.50203\times10^{-6}}{0.0031265}log_2\left(\frac{1.50203\times10^{-3}}{0.0031265}\right)$$

$$-\frac{0.0015625}{0.00312650.011765862} log_2 \left(\frac{0.0015625}{0.0031265}\right) - \frac{0.0015625}{0.0031265} log_2 \left(\frac{0.0015625}{0.0031265}\right)$$

= 0.5060950bits

Therefore, GainRatio(Single_Epithelial_Cell_Size)= $\frac{0.0059888}{0.5060950}$ = 0.011833 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Bare_Nuclei** is

Info $_{\text{Bare\_Nuclei}}$ (D)

$$=\frac{0.0015625}{0.0031265} \times \left(-\frac{0}{0.0015625} log_2 \frac{0}{0.0015625} - \frac{0.0015625}{0.0015625} log_2 \frac{0.0015625}{0.0015625}\right)$$

$$+\frac{1.50203\times10^{-6}}{0.0031265} \times \left(-\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}} log_2 \frac{1.50203\times10^{-6}}{1.50203\times10^{-6}} - \frac{0}{1.50203\times10^{-6}} log_2 \frac{0}{1.50203\times10^{-6}}\right)$$

$$+\frac{0.0015625}{0.0031265} \times \left(-\frac{0}{0.0015625} log_2 \frac{0}{0.0015625} - \frac{0.0015625}{0.0015625} log_2 \frac{0.0015625}{0.0015625}\right)$$

= 0 bit

The gain in material from such a separating could be

Gain (Bare_Nuclei) = $0.0059888 - 0$ = 0.0059888bits

Calculation of gain ratio for the attribute Bare_Nuclei

SplitInfo $_{\text{Bare\_Nuclei}}$ (D)

$$=-\frac{0.0015625}{0.00312650.011765862} log_2 \left(\frac{0.0015625}{0.0031265}\right) - \frac{1.50203\times10^{-6}}{0.0031265} log_2 \left(\frac{1.50203\times10^{-3}}{0.0031265}\right)$$

$$-\frac{0.0015625}{0.0031265} log_2 \left(\frac{0.0015625}{0.0031265}\right)$$

= 0.5060950bits

Therefore, GainRatio(Bare_Nuclei)= $\frac{0.0059888}{0.5060950}$ = 0.011833 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Bland_Chromatin** is

Info $_{\text{Bland\_Chromatin}}$ (D)

$$=\frac{1.50203\times10^{-6}}{0.0031265} \times \left(-\frac{1.50203\times10^{-6}}{1.50203\times10^{-6}} log_2 \frac{1.50203\times10^{-6}}{1.50203\times10^{-6}} - \frac{0}{1.50203\times10^{-6}} log_2 \frac{0}{1.50203\times10^{-6}}\right)$$

$$+\frac{0.0015625}{0.0031265} \times \left(-\frac{0}{0.0015625} log_2 \frac{0}{0.0015625} - \frac{0.0015625}{0.0015625} log_2 \frac{0.0015625}{0.0015625}\right)$$

$$+\frac{0.0015625}{0.0031265} \times \left(-\frac{0}{0.0015625} log_2 \frac{0}{0.0015625} - \frac{0.0015625}{0.0015625} log_2 \frac{0.0015625}{0.0015625}\right)$$

= 0 bit

The gain in material from such a separating could be

Gain (Bland_Chromatin) = $0.0059888 - 0$=0.0059888bits

Calculation of gain ratio for the attribute Bland_Chromatin

SplitInfo $_{\text{Bland\_Chromatin}}$ (D) $=-\frac{1.50203\times10^{-6}}{0.0031265}log_2\left(\frac{1.50203\times10^{-3}}{0.0031265}\right)$

$-\frac{0.0015625}{0.00312650.011765862}log_2\left(\frac{0.0015625}{0.0031265}\right)-\frac{0.0015625}{0.0031265}log_2\left(\frac{0.0015625}{0.0031265}\right)$

= 0.5060950bits

Therefore, GainRatio (Bland_Chromatin)= $\frac{0.0059888}{0.5060950}$ = 0.011833 bits

The predictable material required to organize a tuple in D if the tuples are separated according to **Mitoses** is

Info $_{\text{Mitoses}}$ (D)

$=\frac{0.0031265}{0.0031265}\times\left(-\frac{1.502037\times10^{-6}}{0.0031265}log_2\frac{1.502037\times10^{-6}}{0.0031265}-\frac{0.003125}{0.0031265}log_2\frac{0.003125}{0.0031265}\right)$

= 0.00598588 bit

The gain in material from such a separating could be

Gain (Mitoses) = 0.0059888 − 0.00598588 = 0bits

Calculation of gain ratio for the attribute Mitoses

SplitInfo $_{\text{Mitoses}}$ (D) = $-\frac{0.0031265}{0.0031265}log_2\frac{0.0031265}{0.0031265}$

= 6.9293439bits

Therefore, GainRatio(Mitoses)= $\frac{0}{6.9293439}$ = 0bits

As a result, the attribute “Clump Thickness” with the maximum Gain Ratio is chosen to carry on the expansion of the classification. The decision tree generated by weighted C4.5 for 10 tuples of Breast Cancer dataset is shown in Figure 3.3.

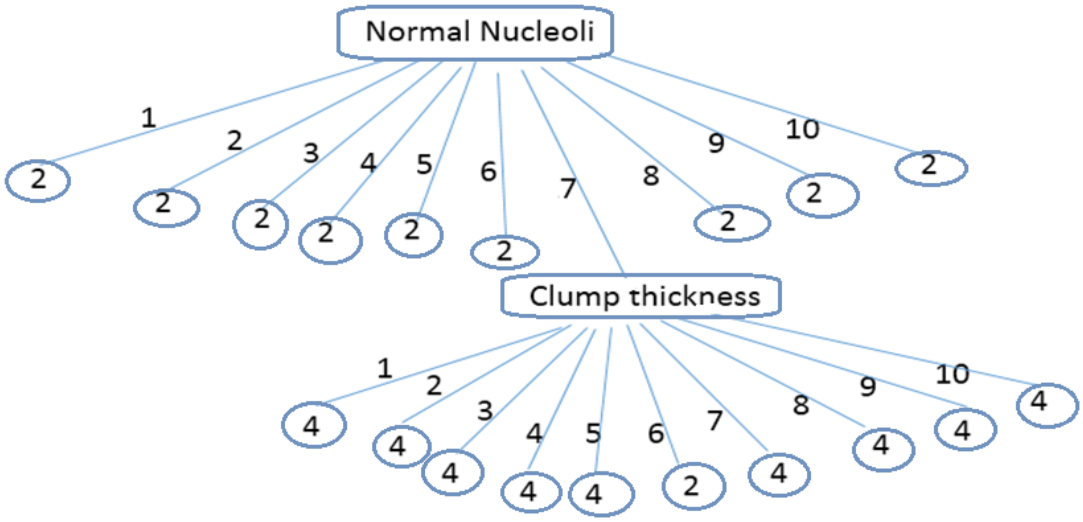


Figure 3.3 Complete Decision Tree using Breast Cancer Dataset

As the result of the testing phase, the accuracy measure is used to compare the performance of two algorithms: C4.5 and weighted C4.5.

- Accuracy is the proportion of correct predictions the classifier makes relative to the size of dataset.

$$Accuracy = \frac{number\ of\ correct\ classification}{total\ number\ of\ tuples\ in\ the\ dataset} \times 100$$

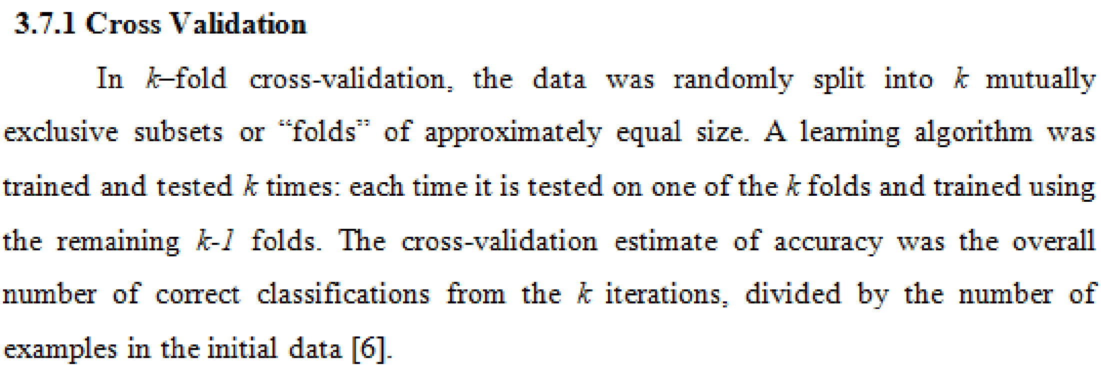
3.6 Filling in Missing Values

In some cases, some of the values of features or attributes may not be available. In such cases, handling the values of missing attributes must be considered. There are number of way:

- Ignoring any instance with a missing value of an attribute. This will decrease the number of available instances.
- Filling in with the most possible value of the attribute with missing value of the instance.
- Combining the outcomes of classification using each possible value according to the probability of that value [17].

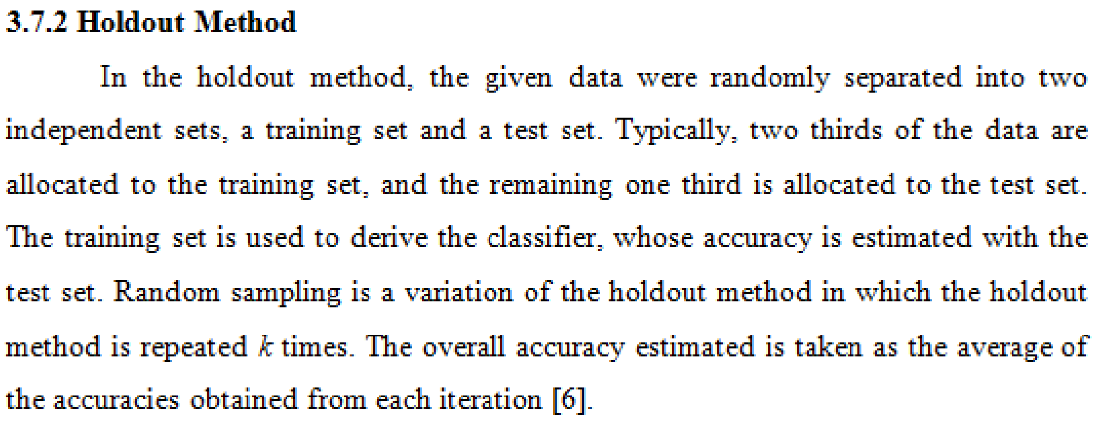
3.7 Evaluating the Accuracy of a Classifier

Cross-validation, Holdout, random subsampling, and the bootstrap [22] are mutual techniques for measuring accuracy grounded on random sampled separates of the assumed data. The habit of such techniques to approximation accuracy surges the general working out time and is suitable for model variety.



3.7.1 Cross Validation

In *k*–fold cross-validation, the data was randomly split into *k* mutually exclusive subsets or “folds” of approximately equal size. A learning algorithm was trained and tested *k* times: each time it is tested on one of the *k* folds and trained using the remaining *k-1* folds. The cross-validation estimate of accuracy was the overall number of correct classifications from the *k* iterations, divided by the number of examples in the initial data [6].



3.7.2 Holdout Method

In the holdout method, the given data were randomly separated into two independent sets, a training set and a test set. Typically, two thirds of the data are allocated to the training set, and the remaining one third is allocated to the test set. The training set is used to derive the classifier, whose accuracy is estimated with the test set. Random sampling is a variation of the holdout method in which the holdout method is repeated *k* times. The overall accuracy estimated is taken as the average of the accuracies obtained from each iteration [6].

3.7.3 Random Subsampling

Using the random subsampling method, the assumed data are arbitrarily divided into two mutually exclusive sets, a preparation set and a check set. Naturally, two-thirds of the data are used as the preparation set, and the residual one-third is used as the check set. Then, the preparation set is used to develop the model, whose accuracy is probable with the check set. To make it more objective, random subsampling is done with *k* iterations. The estimated accuracy is calculated by taking the regular of the precisions got from each repetition. For forecast, the regular of the analyst mistake rates is taken. Figure 2.3 illustrates the random subsampling method. The whole dataset (100%) is divided to a preparation set (67%) and check set (33%). This is done k-times. After that, the model is learned from each combination of preparation and check sets. The final accuracy comes from the average of all acquired accuracies (Accuracy *1* to Accuracy *k*).

To compare the models derived by C4.5 and weighted C4.5, random subsampling technique is performed on each dataset in this study. The original dataset are regular separated into two mutually exclusive sets, a preparation set and a check set. Therefore, k-times of model construction and the accuracy estimation of the models derived by the algorithms are implemented. To compute the accuracy of the model, the number of right categorized tuples is divided by the total number of tuples in check set.

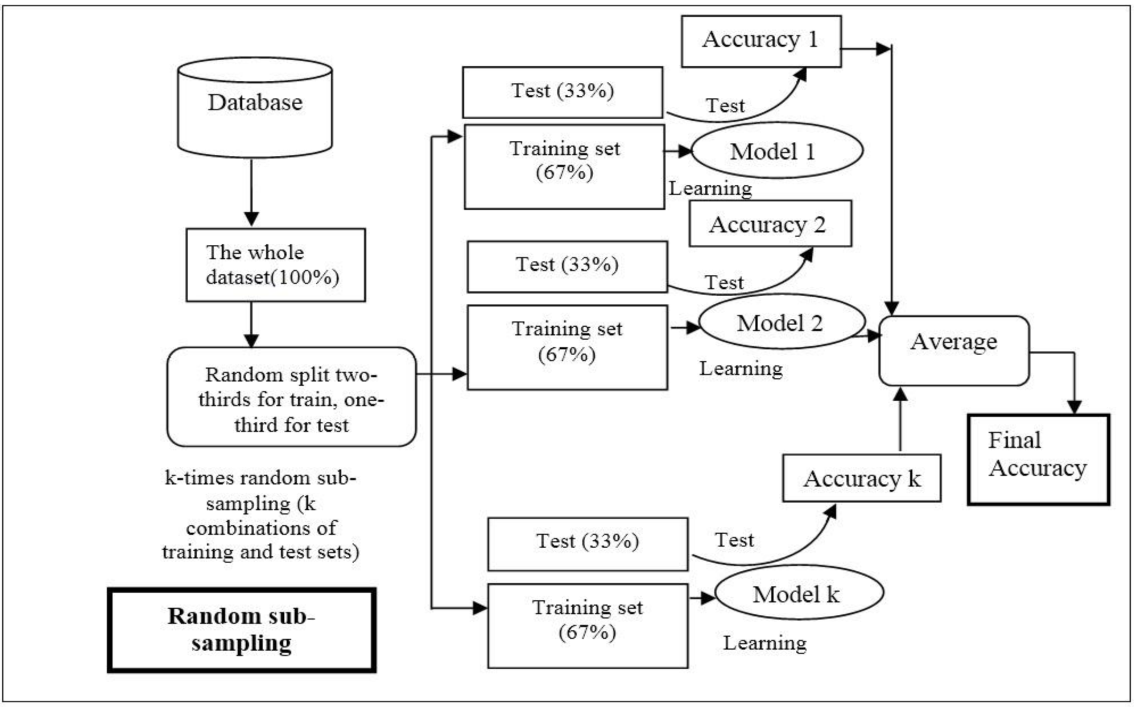


Figure 3.4 Random Subsampling (graphical representation)

CHAPTER 4
DESIGN AND IMPLEMENTATION

The system presents the comparative study for classification of Breast Cancer classification using decision tree algorithms; C4.5 and weighted C4.5. We compare the accuracy of the weighted C4.5 and traditional C4.5 algorithms. There are various classification algorithms, among them; decision tree algorithm is a tree like structure. It is well known algorithm and works well in the area of diagnosis problems and decision support systems.

4.1 System Design

In this thesis, the whole dataset is randomly separated into two mutually exclusive sets, a preparation set and a check set. Typically, two-thirds of the data are used as the preparation set, and the residual one-third is used as the check set. The random is done with *k* iterations. According to random method, preparation set is imported into the system. The imported preparation set is trained to build the decision tree model to get the maximum rule length, the amount of rules created, and the total number of condition checks to classify the whole preparation set as results from the derived model by each algorithm. And then the check set is expended to check the model in order to obtain accuracy estimation. The results from each iteration are averaged. The system reports results from each iteration and its average in terms of bar chart.

Experimental results from both preparation and checking phases are used to compare two decision tree algorithms: traditional C4.5 and weighted C4.5 algorithms. As the preparation phase results, the total number of leaves of decision tree representing the total number of rules generated by each algorithm, the maximum depth the decision tree representing the maximum length of the rules, the total number of nodes representing the condition check required to classify the whole preparation set and processing time representing the time required to build the model are obtained from each iteration. Each of preparation results is calculated to make comparison. As the checking phase, the average accuracy estimation from each iteration is used to compare performance of two algorithms. For analysis purpose random iteration runs are performed on each dataset. An iteration run for one dataset is performed as follows:

Setp 1. Original dataset is randomly separated into two mutually exclusive sets, a preparation set and a check set. Typically, two-thirds of the facts are used as the preparation set, and the remaining one-third is used as the check set.

Setp 2. The system implements the algorithm on the preparation set to construct the model and checks on the checking set to estimate accuracy.

Setp 3. For each algorithm, the system produces preparation results from the model in preparation phase and divides the number of exact classification to the element size of the check set to estimation accuracy in checking phase.

Setp 4. Step 2 and step 3 have to be done until each part is left one time for checking set.

After all these iterations, the preparation results and the accuracy estimations obtained from iterations are averaged. The resulting average results are used to compare two algorithms.

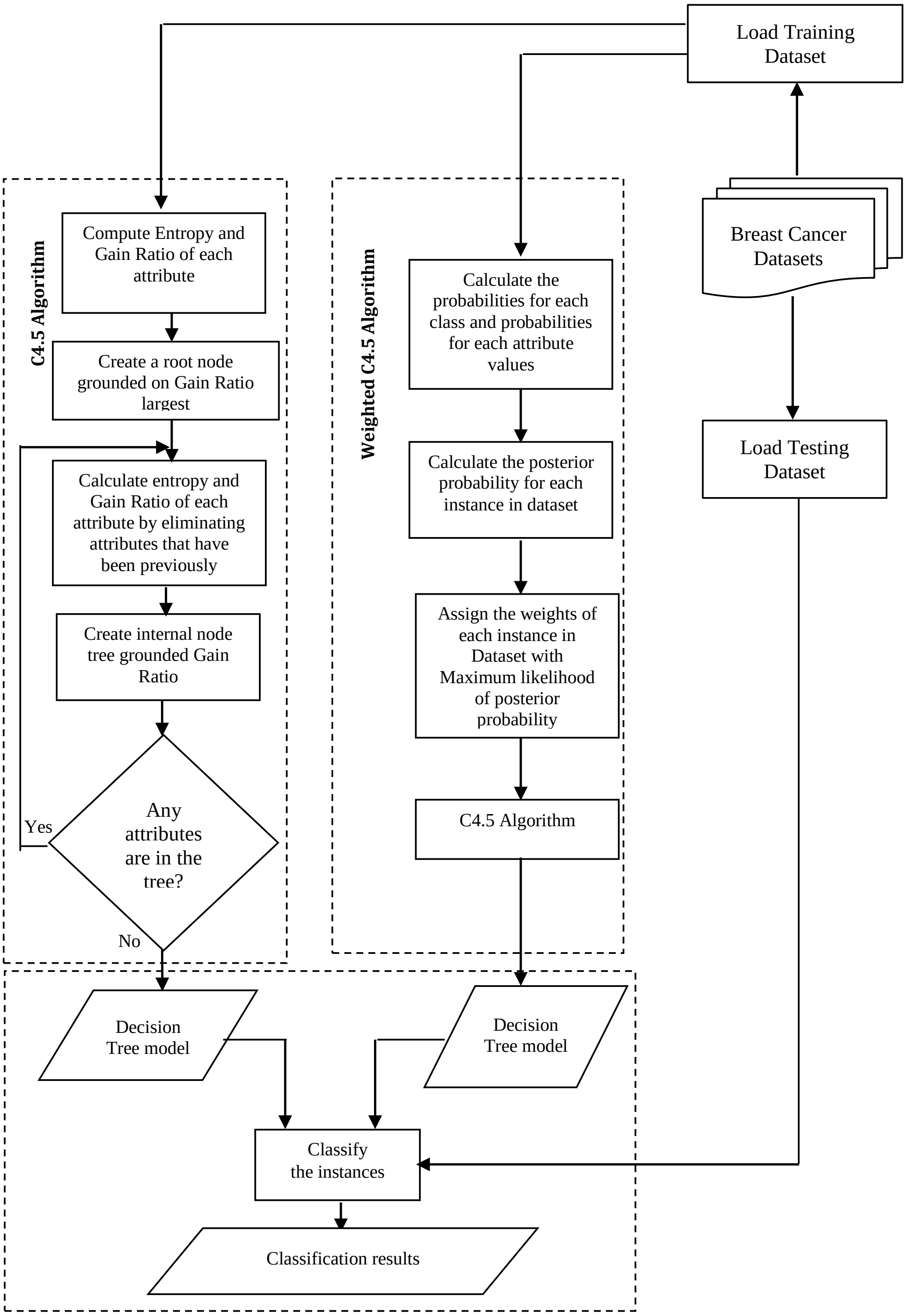
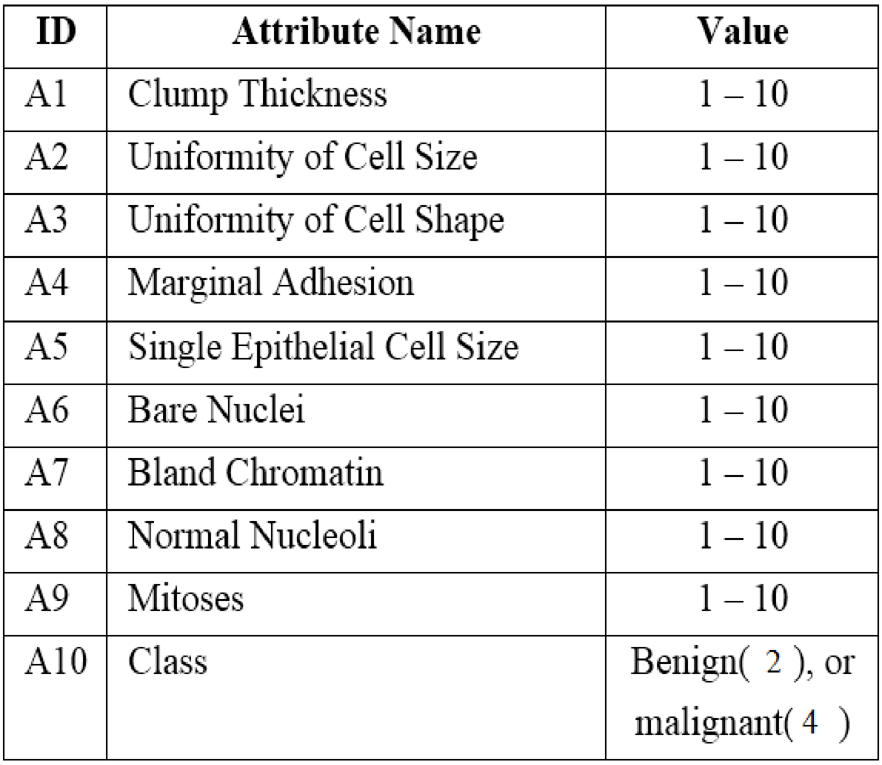

Load Training Dataset
Breast Cancer Datasets
Load Testing Dataset
C4.5 Algorithm
Compute Entropy and Gain Ratio of each attribute
Create a root node grounded on Gain Ratio largest
Calculate entropy and Gain Ratio of each attribute by eliminating attributes that have been previously
Create internal node tree grounded Gain Ratio
Any attributes are in the tree?
Yes
No
Weighted C4.5 Algorithm
Calculate the probabilities for each class and probabilities for each attribute values
Calculate the posterior probability for each instance in dataset
Assign the weights of each instance in Dataset with Maximum likelihood of posterior probability
C4.5 Algorithm
Decision Tree model
Decision Tree model
Classify the instances
Classification results

Figure 4.1 System Flow Diagram

4.1.1 Breast Cancer Dataset

In this thesis, there are 2 class labels and 10 attributes in the classification process. The breast cancer dataset contains 683 instances and 10 attributes. Each of the characteristics is assigned a value from 1 to 10 by the pathologist. The larger the value of attribute the greater the likelihood of malignancy. There are two types of classes in dataset, benign (It does not invade nearby tissue or spread to other parts of the body), or malignant (It is serious and likely to spread other parts of the body). Attributes and values used in the preparation datasets are shown in Table 4.1.

Table 4.1. Attribute Names and Values

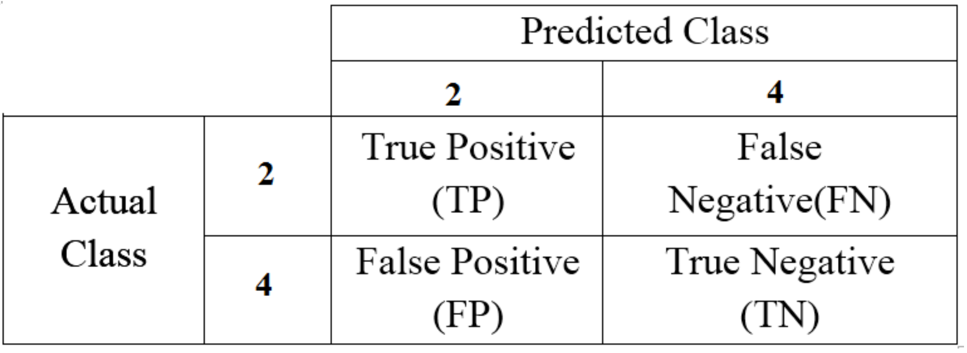


ID	Attribute Name	Value
A1	Clump Thickness	1 – 10
A2	Uniformity of Cell Size	1 – 10
A3	Uniformity of Cell Shape	1 – 10
A4	Marginal Adhesion	1 – 10
A5	Single Epithelial Cell Size	1 – 10
A6	Bare Nuclei	1 – 10
A7	Bland Chromatin	1 – 10
A8	Normal Nucleoli	1 – 10
A9	Mitoses	1 – 10
A10	Class	Benign(2), or malignant(4)

4.2 Classifier Accuracy Measure

Using the preparation set to spring a classifier or analyst and approximating the accuracy of the ensuring learned model can result in ambiguous overoptimistic evaluations due to overspecialization of the studying algorithm to the data. The accuracy of a classifier on a given check set is the percentage of check set tuples that are correctly classified by the classifier.

The confusion matrix is a expedient tool for examining how well a classifier can identify tuples of dissimilar classes. Given m classes, a confusion matrix is a table of at smallest size m by m. An entry, $CM_{i,j}$ in the first m rows and m columns designates the number of tuples of class i that were branded by the classifier as class j. For a classifier to have moral accuracy, preferably greatest of the tuples would be denoted along the slanting of the confusion matrix, from entry $CM_{1,1}$ to entry $CM_{m,m}$, with the respite of the entrances being near to zero. Figure 4.2 represents confusion matrix for multi-classes.



		Predicted Class	
		2	**4**
Actual Class	**2**	True Positive (TP)	False Negative(FN)
	4	False Positive (FP)	True Negative (TN)

Figure 4.2 A Confusion Matrix for Positive and Negative Tuples

Given two classes, true positives refer to the positive tuples (tuples of the main class of interest) that were properly branded by the classifier, while true negatives are the negative tuples that were right labeled by the classifier. False positive tuples are the negative tuples that were mistake labeled. Similarly, false negatives are the positive tuples that were incorrectly labeled. These terms are suitable when evaluating a classifier's aptitude.

If how well the classifier can identify the positive tuples and how well it can recognize the negative tuples would be able to be accessed, then the recall, precision, f-measure and specificity measures can be used respectively. Accuracy is the percentage measure of correctly classified instances for all instances. These measures are defined as

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad 4.1$$

Recall is the measure of the positive instance that is correctly classified and it can be calculated of the following equation.

$$Recall = \frac{TP}{TP+FN} \qquad 4.2$$

Precision is of correctly classified instances for those instances that are classified as positive of the following equation.

$$Precision = \frac{TP}{TP+FP} \qquad 4.3$$

F-measure is the combined metric of precision and recall, i.e., it is harmonic mean of both. It shows how precise the classifier is and also how well the classifier is robust of the following equation.

$$F - measure = \frac{2 \times Recall \times Precision}{Precision + Recall} \qquad 4.4$$

Specificity is the measure of correctly classified negative instances to the total number of negative instances of the following equation.

$$Specificity = \frac{TN}{TN+FP} \qquad 4.5$$

Biometric evaluation system that assigns all authentication attempts a 'score' between closed interval [0, 1]. 0 means no match at all and 1 means a full match.

False Acceptance Rate (FAR) is calculated as a fraction of negative scores exceeding your threshold.

$$FAR = \frac{FP}{(FP+TN)} \qquad 4.6$$

False Rejection Rate (FRR) is calculated as a fraction of positive score falling below your threshold.

$$FRR = \frac{FN}{(TP+FN)} \qquad 4.7$$

where TP is the number of true positives, TN is the number of true negative, FP is the number false positive and FN is the number of false negative.

4.3 Main Page of the System

Figure 4.3 shows the main page of the system. In this form, there are two menus in menu bar. They are 'File' and 'Exit' menus.

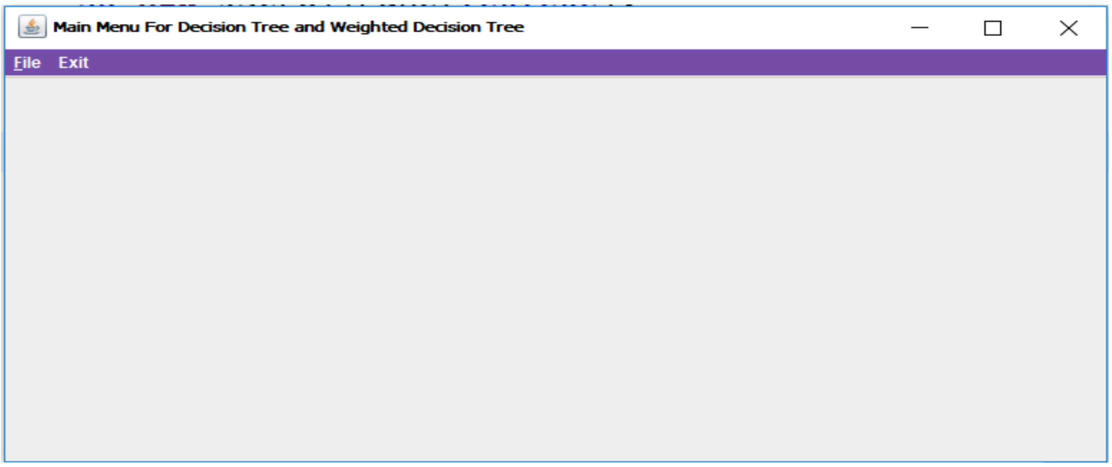


Figure 4.3 Main Page of the System

4.3.1 File Menu

Figure 4.4 shows file menu of the system. Using this menu, comparison of C4.5 and weighted C4.5 algorithms can be implemented.

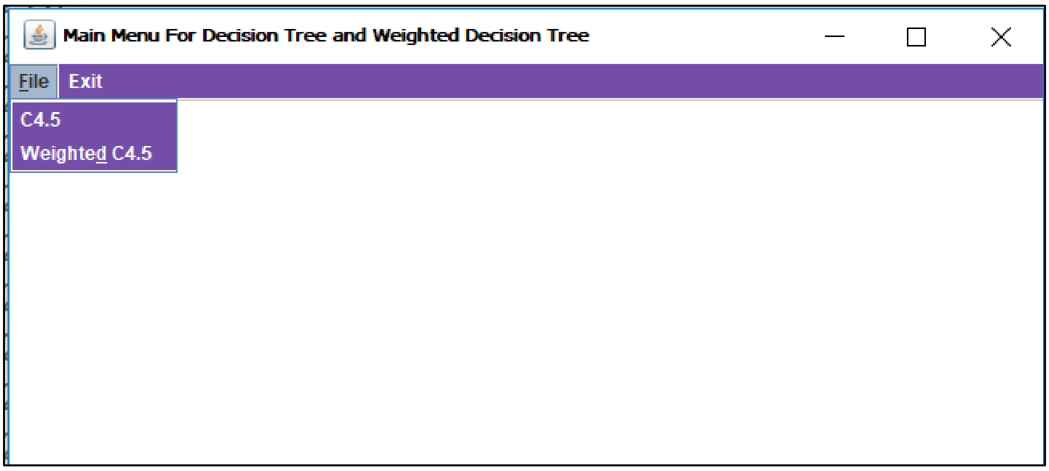
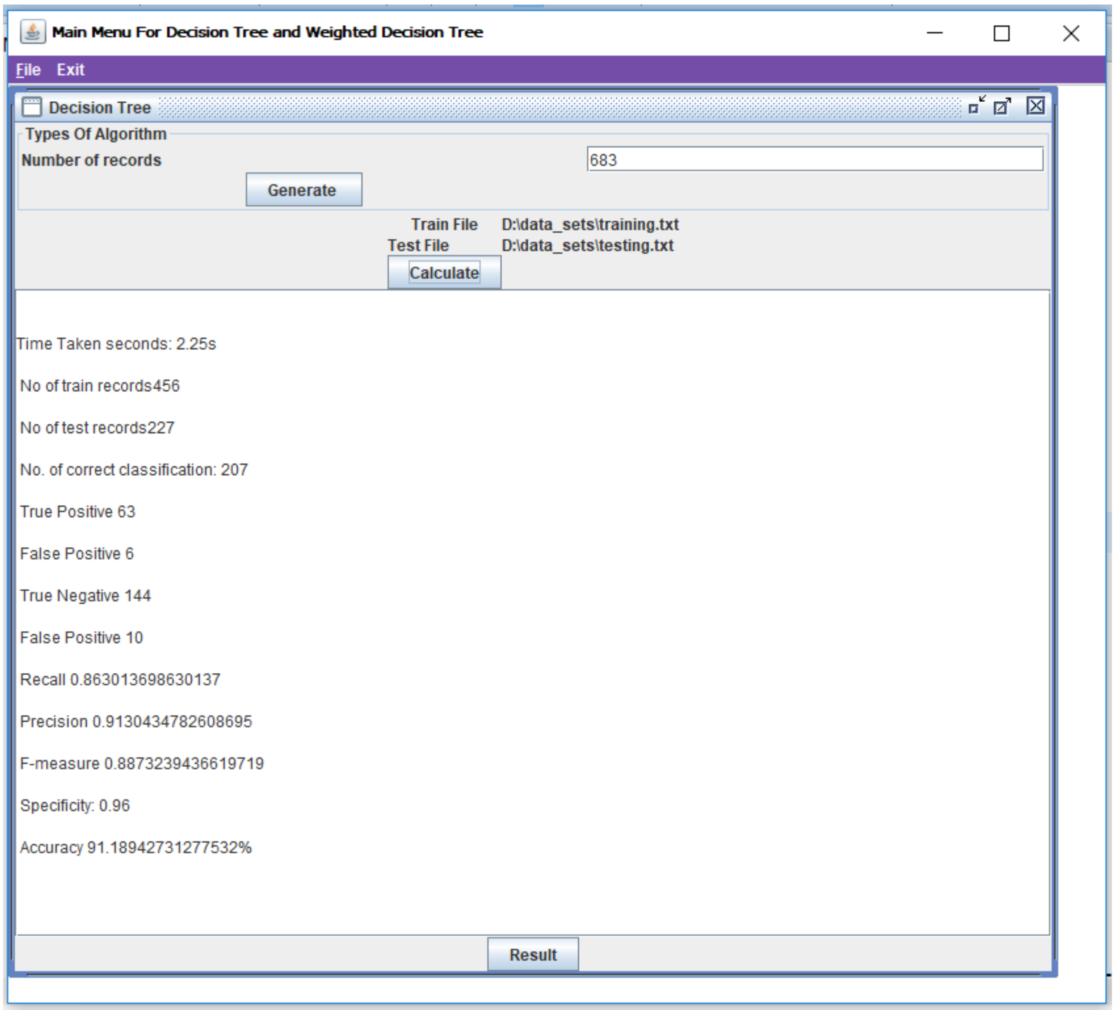


Figure 4.4 File Menu of the System

4.3.1.1 C4.5 Form

Figure 4.5 shows C4.5 algorithm implemented this system. In this form, there are one textbox, three buttons and one textarea. The number of record can be inserted into the textbox. And then, the 'Generate' button is clicked to see the 'Train File' and 'Text File'. Next, by clicking the 'Calculate' button, the preparation file and checking file are imported into the system. Imported preparation set is trained with C4.5

algorithm and to generate the decision tree model. The imported checking set is applied to check the model trained by C4.5 and the results of specificity, precision, recall, f-measure, FAR, FRR and accuracy to show in textarea. And then, the ‘Result’ button is clicked to see the checking file. The comparison between the class label and check label can be seen in Figure 4.6.

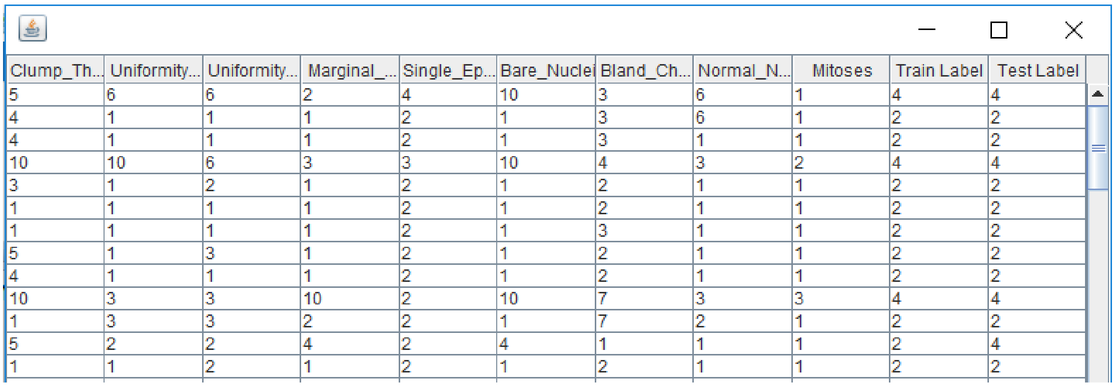


Number of records 683

Train File D:\data_sets\training.txt
Test File D:\data_sets\testing.txt

```
Time Taken seconds: 2.25s

No of train records456

No of test records227

No. of correct classification: 207

True Positive 63

False Positive 6

True Negative 144

False Positive 10

Recall 0.863013698630137

Precision 0.9130434782608695

F-measure 0.8873239436619719

Specificity: 0.96

Accuracy 91.18942731277532%
```

Figure 4.5 C4.5 Algorithm of the System

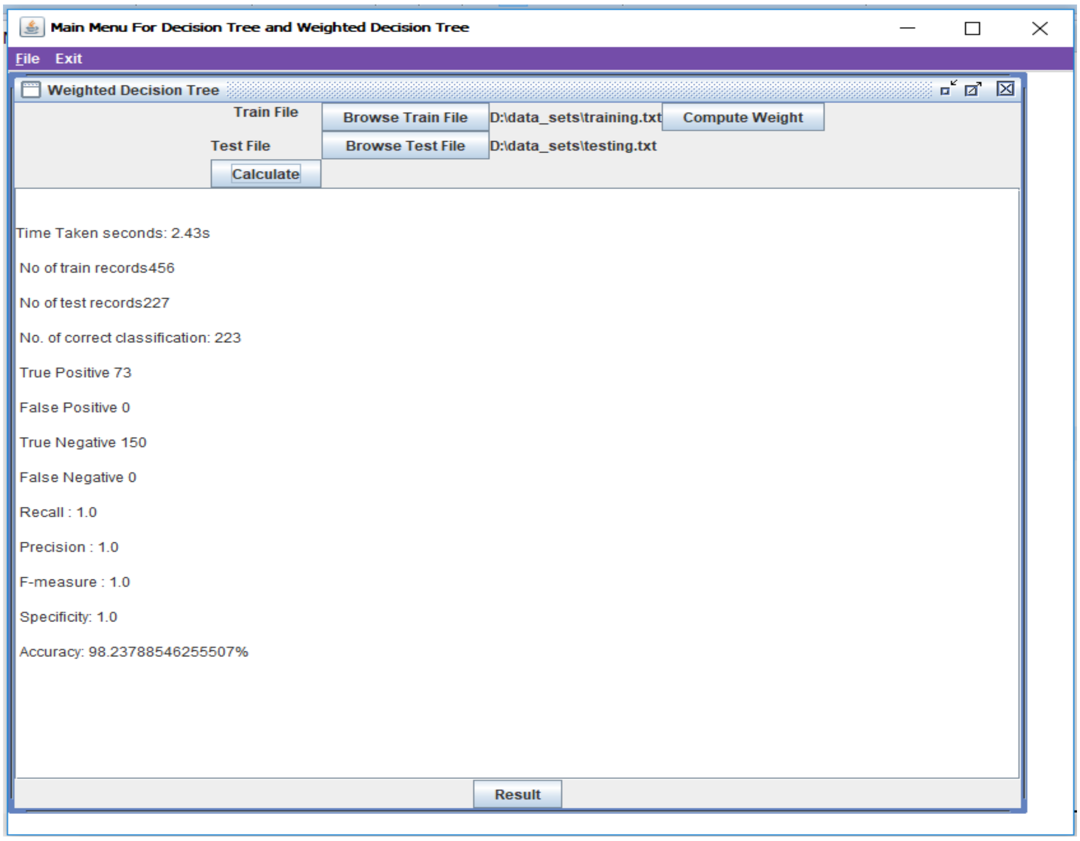


Clump_Th...	Uniformity...	Uniformity...	Marginal_...	Single_Ep...	Bare_Nuclei	Bland_Ch...	Normal_N...	Mitoses	Train Label	Test Label
5	6	6	2	4	10	3	6	1	4	4
4	1	1	1	2	1	3	6	1	2	2
4	1	1	1	2	1	3	1	1	2	2
10	10	6	3	3	10	4	3	2	4	4
3	1	2	1	2	1	2	1	1	2	2
1	1	1	1	2	1	2	1	1	2	2
1	1	1	1	2	1	3	1	1	2	2
5	1	3	1	2	1	2	1	1	2	2
4	1	1	1	2	1	2	1	1	2	2
10	3	3	10	2	10	7	3	3	4	4
1	3	3	2	2	1	7	2	1	2	2
5	2	2	4	2	4	1	1	1	2	4
1	1	2	1	2	1	2	1	1	2	2

Figure 4.6 The Results of Testing file

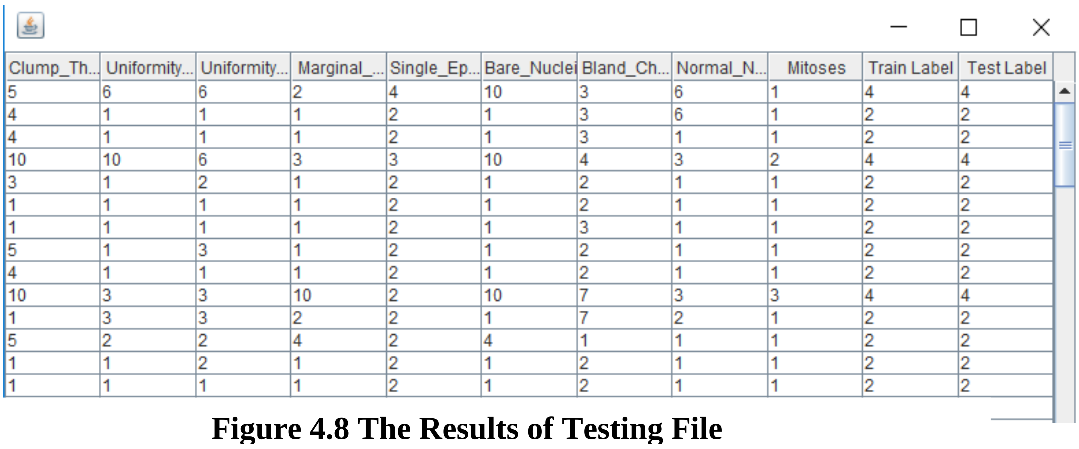
4.3.1.2 Weighted C4.5 Form

Figure 4.7 shows weighted C4.5 algorithm implemented this system. ‘Weighted Decision Tree’ form has five buttons namely ‘Browse Train File’, ‘Browse Check File’, ‘Compute Weight’, ‘Calculate’ and ‘Result’. And then, this form has one textarea. To import the preparation dataset into the system, the ‘Browse Train File’ button is used and to import the checking dataset into the system, the ‘Browse Check File’ button is used. And then, ‘Compute Weight’ button is used to calculate the preparation dataset by Naïve Bayes theorem. This theorem initializes the weights of each preparation data. The highest posterior probability is added to each class occurring in the preparation data. Next, by clicking the ‘Calculate’ button, the preparation file and checking file are imported into the system. Imported preparation set is trained with weighted C4.5 algorithm and to generate the decision tree model. The imported checking set is applied to check the model trained by weighted C4.5 and the results of specificity, precision, recall, f-measure, FAR, FRR and accuracy are shown in textarea. And then, the ‘Result’ button is clicked to see the checking file. The comparison between the class label and check label can be seen in Figure 4.8.



```
Time Taken seconds: 2.43s

No of train records456

No of test records227

No. of correct classification: 223

True Positive 73

False Positive 0

True Negative 150

False Negative 0

Recall : 1.0

Precision : 1.0

F-measure : 1.0

Specificity: 1.0

Accuracy: 98.23788546255507%
```

Figure 4.7 Weighted C4.5 Algorithm of the System



Clump_Th...	Uniformity...	Uniformity...	Marginal_...	Single_Ep...	Bare_Nuclei	Bland_Ch...	Normal_N...	Mitoses	Train Label	Test Label
5	6	6	2	4	10	3	6	1	4	4
4	1	1	1	2	1	3	6	1	2	2
4	1	1	1	2	1	3	1	1	2	2
10	10	6	3	3	10	4	3	2	4	4
3	1	2	1	2	1	2	1	1	2	2
1	1	1	1	2	1	2	1	1	2	2
1	1	1	1	2	1	3	1	1	2	2
5	1	3	1	2	1	2	1	1	2	2
4	1	1	1	2	1	2	1	1	2	2
10	3	3	10	2	10	7	3	3	4	4
1	3	3	2	2	1	7	2	1	2	2
5	2	2	4	2	4	1	1	1	2	2
1	1	2	1	2	1	2	1	1	2	2
1	1	1	1	2	1	2	1	1	2	2

Figure 4.8 The Results of Testing File

4.4 Experimental Results

The experimental results of categorize are to be analyzed weighted C4.5 decision tree and traditional C4.5 decision tree algorithm. The breast cancer dataset from UCI [25] is used for proportional analysis. This system is trained with 683 data records. For each categorize, 2/3 of the dataset is used for preparation and 1/3 of datasets is expended for checking. The following table compares the accuracy results of two classifiers.

Table 4.2: Experimental Results for Different Preparation Records

Experiment	No. of Record	C4.5 algorithm	Weighted C4.5 algorithm
1	100	80%	85%
2	200	87%	90%
3	400	91.25%	92.5%
4	683	94.27%	99.56%

The accuracy comparison on different preparation records by using C4.5 and weighted C4.5 algorithms are illustrated by Table 4.2. According to Figure 4.9 the more preparation data records are used to train C4.5 and weighted C4.5 algorithms, the best accuracy is achieved.

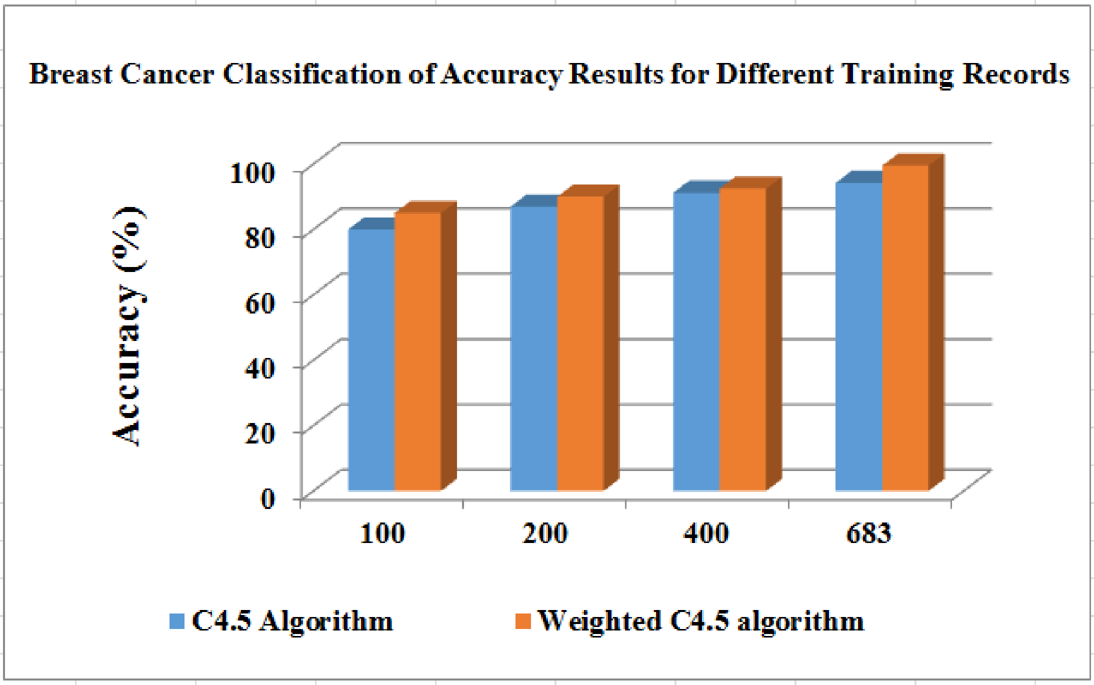


Figure 4.9 Breast Cancer Classifications of Accuracy Results for Different Preparation Records

According to the result, it can be observed that when the amount of patients has been increased, the percentage of system accuracy has been increased slightly. Therefore, it can describe the accuracy of Breast Cancer classification system has been increased when the amount of trained data is increased.

Figure 4.10 and 4.11 illustrates the performance and accuracy comparison of C4.5 and weighted C4.5 algorithms. The algorithm with the best accuracy is weighted C4.5 algorithm with the accuracy of 99.56%. The weighted C4.5 algorithm shows the best accuracy while C4.5 shows the accuracy of 94.27%. Therefore, the accuracy of weighted C4.5 decision tree algorithm is well than the accuracy of the C4.5 algorithm on Breast Cancer dataset.

Using partial percentage of the preparation data, the user can obtain exact result for their Breast Cancer classification. This system is trained with 683 data records. Decision tree is trained with 456 preparation records and checked with 227 checking records. The experimental results for two classifiers are shown in Table 4.3.

Table 4.3 Performance Evaluation Results of Breast Cancer Classification for 683 Data Records

	Recall	Precision	F-measure	Specificity	FAR	FRR	Accuracy (%)
C4.5 algorithm	86.3	95.5	90.6	98.1	19	14.4	94.27
Weighted C4.5 algorithm	100	98.8	99.4	99.3	7	0	99.56

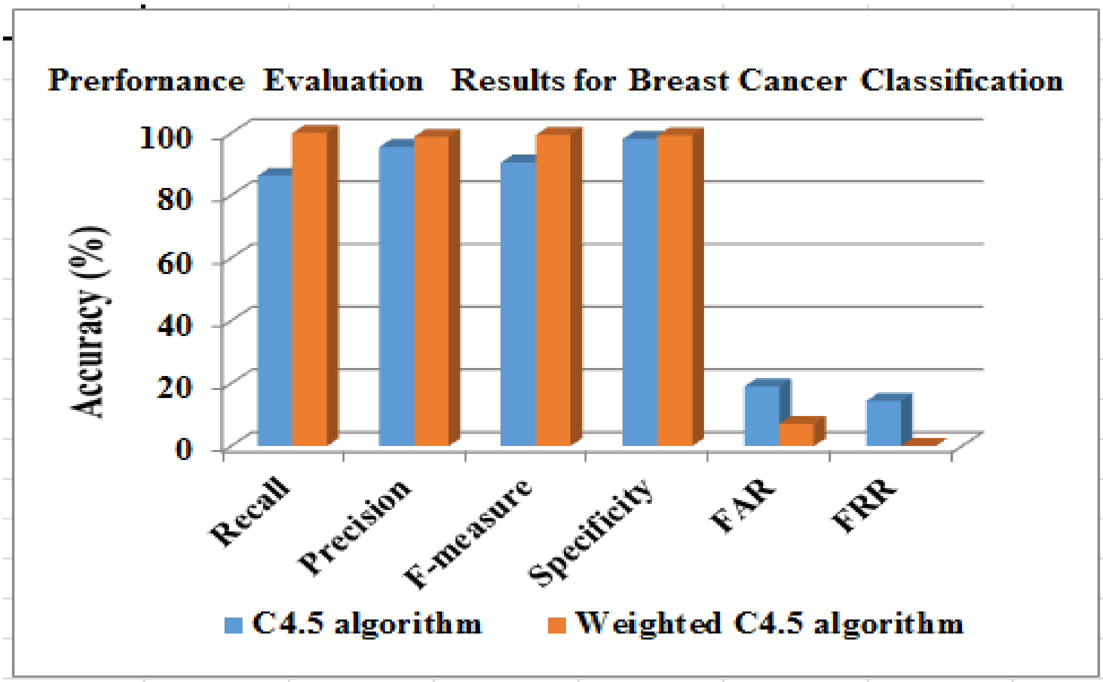


Figure 4.10 Performance Evaluation Results of Breast Cancer Classification for 683 Data Records

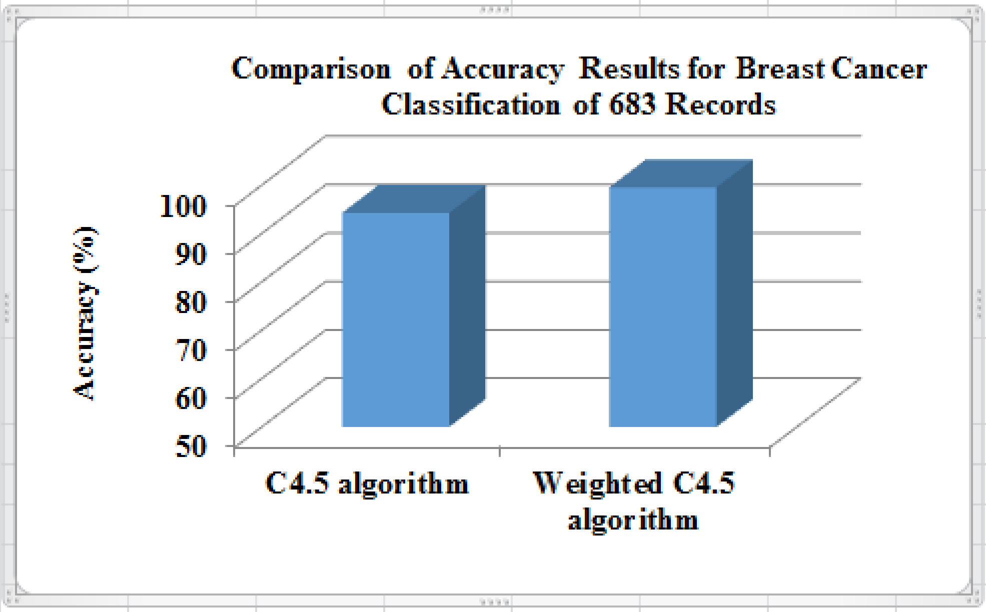


Figure 4.11 Comparison of Accuracy Results for Breast Cancer Classification of 683 Records

CHAPTER 5
CONCLUSION

Computer grounded diagnosis systems is play an increasingly important role in health care facilities. They may improve the quality of the diagnosis process in accuracy and efficiency and the patients can save their cost and time. The automatic diagnosis of Breast Cancer is an essential real-world medical problem. Detection of Breast Cancer in it early stage is the key of treatment.

This system performs the implementation of Decision Tree algorithms and compares their performance grounded on practical implementation. In this thesis, the comparative analysis of C4.5 and weighted C4.5 algorithms classification on Breast Cancer classification is presented. From this study it is found that accuracy of weighted C4.5 algorithm is better than traditional C4.5 algorithm. In this thesis, the system has used 683 records for breast cancer datasets and random subsampling to compute accuracy and confusion matrix of each class of the mode. The experimental results prove that the weighted C4.5 algorithm can achieve high classification rate because weighted C4.5 decision tree algorithm gets the system accuracy of 99.56% by using random subsampling method for accessing classifier accuracy.

5.1 Advantages of the System

This system presents the comparative study of different decision tree algorithms; traditional C4.5 and weighted C4.5. The main advantages of this system is performing the comparative study, building the decision trees with different algorithms. It helps the patients with Breast Cancer classification and medical staffs in deciding on Breast Cancer classification. Classification by weighted C4.5 reduces the errors of C4.5 decision tree algorithms and therefore it provides the better accuracy.

5.2 Limitations and Further Extensions

The comparative study has few limitations. This comparative study is only for Breast Cancer Classification. Other medical problems can be implemented to this comparative study in its further extension. More algorithms may also be implemented in this system presents only the classification and hence diagnosis features can be added in its further extension. The system can classify only two stages of breast cancer classification and the user must know the symptoms of the breast cancer. The system is implemented only by using the C4.5 algorithm and Bayesian method. The future work will extend weighted C4.5 algorithm to work on the datasets and other classification methods for accuracy classifier. And it can be planned to check other cancer datasets by using weighted C4.5 classification.

AUTHOR'S PUBLICATIONS

[1] Khin Thuzar Win, Aung Nway Oo, "*Classification with Weighted C4.5 Decision Tree Approach*" , the Proceedings of the 10th Conference on Parallel and Soft Computing (PSC 2019), Yangon, Myanmar, 2019.

[2] Khin Thuzar Win, Aung Nway Oo, "*Breast Cancer Classification with Weighted CART Decision Tree Approach*", University Journal of Science, Engineering and Rearch (UJSER, 2019), Volume-01, Issue-02, Technological University (Kyaukse), Myanmar, June 2019, pp. 247-252.

[3] Khin Thuzar Win, Aung Nway Oo, "*Breast Cancer Classification with Weighted Decision Tree Approach*", the 12th National Conference on Science and Engineering (NCSE), Yangon Technological University, Yangon, Myanmar, June, 2019, pp. 66.

REFERENCES

[1] Carr. J, “An Introduction to Genetic Algorithms”, International Journal of Computer Science and Information Security (IJCSIS), May, 2014.

[2] Chadha. P and Singh. G. N, “Classification Rules and Genetic Algorithm in Data Mining”, Global Journal of Computer Science and Technology Software & Data Engineering, Volume 12, Issue 15, Version 1.0, 2012.

[3] E. Frak and Witten I.H., “Data Mining: Practical Machine learning Tools and Techniques with Java Implementation”, second edition, Morgan Kaufman Publishers, 2005.

[4] Entezari-Maleki R, Minaei B and Rezaei A, “Comparison of Classification Methods Based on the Type of Attributes and Sample Size”, Journal of Convergence Information Technology, September 2009.

[5] Falangis. K, “Mathematical Programming Models for Classification Problems with Applications to Credit Scoring”, Ph.D Thesis, The University of Edinburgh, 2013.

[6] Han, Jiawei and Kamber, Micheline, “Data Mining, Concept and Techniques”, Sixed edition, Morgan Kaufmann Publishers, 2006, ISBN 1-55860-494-8.

[7] J. Han and M. Kamber, “Classification and Prediction”, Third edition, Morgan Kaufmann Publishers, 2012, ISBN 978-0-12-381479-1,pp. 330-348.

[8] J. Han, M. Kamber, “Data mining Concepts and Techniques” Third edition, Morgan Kaufmann Publishers, 2012, ISBN 978-0-12-381479-1, pp. 27-28.

[9] J. R. Quinlan, “C4.5: Programs for Machine Learning”, Morgan Kaufmann Publishers, Sam Mateo, CA, Machine Learning Vol. 16, 1993, pp. 235-240.

[10] J. R. Quinlan, “Induction of Decision Tree”, Machine Learning Vol. 1, 1986, pp.81-106.

[11] Kiang M. Y, “A Comparative Assessment of Classification Methods”, Decision Support Systems, 1st May 2002, pp.441-454.

[12] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone,“Classification and Regression Tree”, Statistics probability series, Wadsworth, Belmont, 1984.

[13] N. Lavrac, “Data Mining in Medicine; Selected Technique and Application”, Proceeding of Second International Conference on the Practical Application of Knowledge Discovery and Data Mining, London, 1998.

[14] P. Hamsagayathri, P. Sampath, “Decision Tree Classifiers for Classification of Breast Cancer”, International Journal of Current Pharmaceutical Research ISSN 0975-7066, Vol 9, Issue 2, 2017.

[15] Quinlan. J. R, “ C4.5 : Programs for Machine Learning”, Morgan Kaufmann Publications, ISBN 1-55869-238-0, Vol-16, 1993.

[16] Quinlan. J. R, “Induction of Decision Trees”, Boston: Kluwer, Academic Publishers, ISSN 0-88561-25, 1986, pp. 81-106.

[17] Quinlan, J.R, “Unknown Attribute Values in Induction” Proceeding of the Sixed International Workshop on Machine Learning, 1989, pp. 164-168.

[18] Rokah, Lior and Maimon, Oded. Z, “Data Mining With Decision Trees: Theory and Applications”, 1st Mar 2008, ISBN 978-981-277-171-1, pp. 1-11.

[19] Robu. R and Holban. S, “A Genetic Algorithm for Classification”, Recent Researches in Computers and Computing, ISBN: 978-1-61804-000-8.

[20] R. Nithya and B. Santhi, “A Data Mining Techniques for Diagnosis of Breast Cancer Disease”, Data Mining and Soft Computing Techniques, 2014, ISSN 1818-4952, pp. 18-23.

[21] Singh Y and Chauhan. A. S, “Neural Networks in Data Mining”, Journal of Theoretical and Applied Information Technology, 2009.

[22] Thanaruk Theeramunkong, “Introduction to Concepts and Techniques in Data Mining and Application to Text Mining”, Second edition, Sirindhorn International Institute of Technology Thammasat University, pp. 223-244, 2012.

[23] Online Document, Fayyad, et. al, “The Primary Tasks of Data Mining”, [online] available: http://www2.cs.uregina.ca/~dbd/cs831/notes/kdd/2_tasks .html, [accessed 1996].

[24] Online Document, Kilany and Rania M., “Efficient Classification and Prediction Algorithms for Biomedical Information”, [online] available:

http://digitalcommons.uconn.edu/dissertations/105, [accessed 2013].

[25] Online Document, WIlliam H. Wolberg , "Breast Cancer Wisconsin (Original) Data Set", [online] available: http://www.ics.uci.edu/mlearn/MLRepository.html, University of Wisconsin Hospitals Madison, Wisconsin, USA, [accessed July 1992].