DISEASE PREDICTION OF HEALTH CARE SYSTEM BASED ON GINI INDEX CLASSIFICATION

THIRI

M.C.Sc. (THESIS)

JULY, 2016

DISEASE PREDICTION OF HEALTH CARE SYSTEM BASED ON GINI INDEX CLASSIFICATION

BY

THIRI

B.C.Sc. (Hons.)

A dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

(M.C.Sc.)

University of Computer Studies, Mandalay

JULY 2016

ACKNOWLEDGEMENTS

Firstly, I would respectfully like to thank **Dr. Moe Pwint**, Rector, the University of Computer Studies, Mandalay, for her kind permission to develop this thesis and for giving general guidance during the period of study.

I am grateful to my thesis supervisor **Dr. Myat Myat Min**, Associate Professor, Faculty of Computer Science, the University of Computer Studies, Mandalay, for her invaluable guidance regarding the selection of this topic, patient supervision, giving expert advice, motivation and support throughout this study.

I would also like to thank Daw Khin Nandar Htun, Lecturer, Department of Linguistics, the University of Computer Studies, Mandalay, for editing my thesis from the language point of view.

I would like to express my special thanks to **all my teachers** who gave me their time and guidance, and all my friends who helped in the task of developing this thesis.

Finally, I would like to thank **my parents** for their continuous support, encouragement, not only during the studies, but also throughout my whole life.

ABSTRACT

Data mining refers to extracting or mining knowledge from large amounts of data. Data mining plays an important role in various applications such as business organizations, e-commerce, health care industry, scientific and engineering. Classification is a form of data analysis that can be used to extract models describing important data classes or predict future data trends. In classification techniques, decision tree classifier is one of the simplest probabilistic classifiers. Medical diagnosis can be considered as a classification problem. The diagnosis of diseases is a vital and intricate job in medicine. This system is to classify the diagnosis of heart and diabetes diseases by using Gini Index classification method. Depending on the input symptoms, the classified disease can be given by the system. And then, the evaluation of classifier accuracy is calculated by using the hold-out method on the testing data set. This system is implemented by using Visual C# programming language.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1

INTRODUCTION

There is an old say in "Arawja parama Laba" which means good health is a blessing. In fact, life is man's most valuable possession and next in order of value is "Health". Heart and diabetes are rapid increasing diseases and the horror diseases for human beings. Heart disease is the No.1 killer which is imperative and can cause sudden death.

The use of data mining approaches in medical domain is rapidly increasing because of the effectiveness of these approaches to classification and prediction system improved.

This system is presented as the implementation of diagnosis system for diseases using Gini Index decision tree algorithm. This system takes the symptoms as input, classifies them and states whether disease is possible or not and presents the accuracy of prediction. This system is leaded to apply Gini Index algorithm for learning the symptoms and the producing prediction.

1.1 Introduction to Heart and Diabetes Diseases

Heart Disease

Heart attack occurs when the heart's supply of blood is stopped. The most common heart attack symptom in men and women is chest pain or discomfort. However, only half of women who have heart attacks have chest pain. Women are more likely than men to report back or neck pain, indigestion, heartburn, nausea (feeling sick to the stomach), vomiting, extreme fatigue (tiredness), or problems breathing.

Heart attacks also can cause upper body discomfort in one or both arms, the back, neck, jaw, or upper part of the stomach. Other heart attack symptoms are light-headedness and dizziness, which occur more often in women than men. Men are more likely than women to break out in a cold sweat and to report pain in the left arm during a heart attack. Some people have heart attack without symptoms. A silent MI (Myocardial Infarct) can occur in anyone, though it occurs more often among people with diabetes.

Sudden death occurs when there is an abrupt loss of the heart's ability to pump blood. This may be because of heart attack or serious abnormality of the heart's rhythm.

Diabetes Disease

Diabetes is a disease in which blood glucose levels are above normal. Most of the food we eat is turned into glucose, or sugar, for our bodies to use for energy. The pancreas, an organ that lies near the stomach, makes a hormone called insulin to help glucose get into the cells of our bodies. When you have diabetes, your body either does not make enough insulin or cannot use its own insulin as well as it should. This causes sugar to build up in your blood.

Diabetes can cause serious health complications including heart disease, blindness, kidney failure, and lower-extremity amputations. Diabetes is the seventh leading cause of death in the United States. Diabetes can often be detected by carrying out a urine test, which finds out whether excess glucose is present. This is normally backed up by a blood test, which measures blood glucose levels and can confirm if the cause of your symptoms is diabetes.

1.2 Data Classification

Data classification is a two-step process. In the first step, a model is built describing a predetermined set of data classes or concepts. Each tuple is assumed to be a predefined class, as determined by one of the attribute called the class label attributes. The data tuples are analyzed to build the model collectively from the training dataset. In the second step, the model is used for classification. First, the predictive accuracy of the model or classifier is estimated.

In Figure 1.1 training data are analyzed by a classification algorithm. The class label attribute is credit_rating. The classifier is represented in the form of classification rules. In Figure 2.2, test data are used to estimate the accuracy. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.

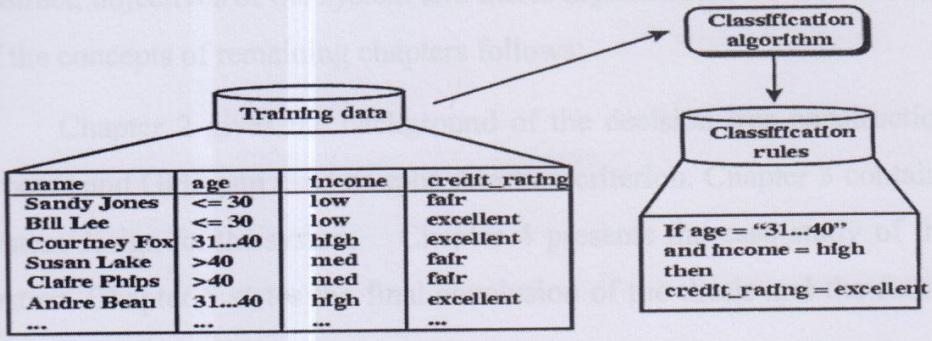


Figure 1.1 Learning Phase: Model Construction

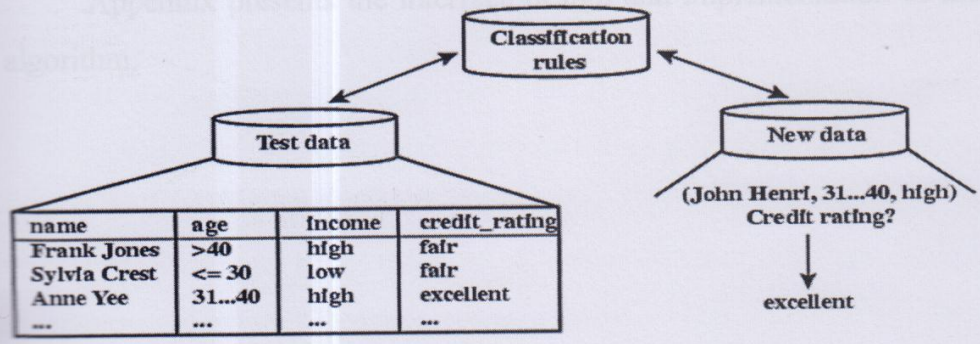


Figure1.2 Testing Phase: Model Usage

1.3 Objectives of the Thesis

There are five main objectives in this system.

- to know the important role of classification technique in medical domain
- to provide the computerized system in medical field using classification method
- to support knowledge about heart and diabetes diseases
- to help clinicians for improving the quality of decision making in medical domain

1.4 Organization of the Thesis

This thesis is organized into five chapters. This chapter introduces abstract, objectives of the system and thesis organization. A brief outline of the concepts of remaining chapters follows:

Chapter 2 gives the background of the decision tree construction process and Gini gain used as split selection criterion. Chapter 3 contains system design for the system. Chapter 4 presents the case study of the system. Chapter 5 states the final conclusion of the thesis and the future work that can be done.

Appendix presents the interface design and implementation of the algorithm.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Introduction

Data mining is the extracting or mining knowledge from large amounts of data. It uses machine learning, statistical and visualization techniques to discover and present knowledge in a form, which is easily comprehensible to humans. Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. Data mining task can be classified into two categories: Descriptive and Predictive. Descriptive mining tasks characterize the general properties of the data in the database. Predictive mining tasks perform inference on the current data in order to make prediction.

Machine learning is an important form of intelligent data analysis. It is common in machine learning to distinguish symbolic and sub-symbolic approaches. Symbolic approaches employ some kind of description language in which the learned knowledge is expressed. Knowledge Discovery in Databases (KDD) is an active and important research area with the promise for a high payoff in many business and scientific applications. One of the main tasks in KDD is classification. A particular efficient method for classification is decision tree induction. Classification Trees is a classification method which uses historical data to construct so-called decision trees. Decision trees are then used to classify new data [4]. One area of machine learning is called classification rule induction. One of the main attractions of rule induction is that the rules are much more transparent and easier to interpret than, say, a regression model or a trained neural network. The classification rule learning task can be defined as follows: Given a set of training examples

(instances for which the classification is known), and a set of classification rules that can be used for prediction or classification of new instances, i.e., cases that haven't been presented to the learner before.

The traditional recursive partitioning approaches CART by Breiman, Friedman, Olshen, and Stone (1984) and C4.5 by Quinlan (1993) use empirical entropy based measures, such as the Gini gain or the Information gain, as split selection criteria. The intuitive approach of impurity reduction added to the popularity of recursive partitioning algorithms, and entropy based measures are still the default splitting criteria in most implementations of classification trees.

Data mining applications has got rich focus due to its significance of classification algorithms. The comparison of classification algorithm is a complex and it is an open problem. First, the notion of the performance can be defined in many ways: accuracy, speed, cost, reliability, etc. Second, an appropriate tool is necessary to quantify this performance. Third, a consistent method must be selected to compare with the measured values. In this thesis it focuses on the decision tree algorithms from classification methods, which is used to assess the classification performance and to find the best algorithm in obtaining qualitative data. Decision Tree algorithm is very useful and well known for their classification. It has an advantage of easy to understand the process of creating and displaying the results. Given a data set of attributes together with its classes, a decision tree produces sequences of rules that can be used to recognize the classes for decision making. The Decision tree method has gained popularity due to its high accuracy of classifying the data set.

2.2 Rule Induction

Machine learning is concerned with the question of how to construct a computer program that automatically learns new facts and theories from data [1]. Rule induction is a special kind of machine learning technique that reasons from specific cases to general principles that is expressible as if-then rules. A number of rule induction systems, such as C4.5 and CN2, have been constructed and applied to discover knowledge from data in different applications.

Given a set of classified examples, a rule learning system constructs a set of if-then rules. An if-then rule has the form:

IF Conditions **THEN** Class.

Conditions contains one or more attribute tests, i.e., features of the form $A_i = v_{ij}$ for discrete attributes, and $A_i <$ v or $A_i >=$ v for continuous attributes (here, v is a threshold value that does not need to correspond to a value of the attribute observed in examples). The conclusion part has the form Class = c_i, assigning a particular value ci to Class.

Machine learning is an important form of intelligent data analysis. The machine learning methods described in this chapter construct explicit symbolic classification rules that generalize the training cases, and are thus instances of symbolic machine learning. This area of machine learning is called classification rule induction. Non-symbolic learning approaches are covered in others. One of the main attractions of rule induction is that the rules are much more transparent and easier to interpret than, say, a regression model or a trained neural network. The classification rule learning task can be defined as follows: Given a set of training examples (instances for which the classification is known), and a set of classification rules that can be used for prediction or classification

of new instances, i.e., cases that haven't been presented to the learner before.

2.3 Classification

Classification is the task of assigning objects to one of several predefined categories, is a pervasive problem that encompasses many diverse applications.

Examples of classification problem include detecting spam email messages based upon the message header and content, categorizing cells as malignant or benign based upon the results of MRI scans, and classifying galaxies based upon their shapes. This introduces the basic concepts of classification, and presents methods for evaluating and comparing the performance of a classification technique. The input data for a classification task is a collection of records [2]. Each record, also known as an instance or example, is characterized by a tuple (x, y), where x is the attribute set and y is a special attribute, designated as the class label (also known as category or target attribute). Although the attributes are mostly discrete, the attribute set can also contain continuous features. The class label must be a discrete attribute.

Table 2.1 Play Golf Dataset

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	Hot	Sunny	High	False	No
07-06	Hot	Sunny	High	True	No
07-07	Hot	Overcast	High	False	Yes
07-09	Cool	Rain	Normal	False	Yes
07-10	Cool	Overcast	Normal	True	Yes
07-12	Mild	Sunny	High	False	No
07-14	Cool	Sunny	Normal	False	Yes
07-15	Mild	Rain	Normal	False	Yes
07-20	Mild	Sunny	Normal	True	Yes
07-21	Mild	Overcast	High	True	Yes
07-22	Hot	Overcast	Normal	False	Yes
07-23	Mild	Rain	High	True	No
07-26	Cool	Rain	Normal	True	No
07-30	Mild	Rain	high	false	yes

Table 2.1 shows a sample data set used for classifying into Yes or No. The attribute set includes properties such as Day, Temperature, Outlook, Humidity, and Windy. Although the attributes presented in Table 2.1 are mostly discrete, the attribute set can also contain continuous features. The class label, on the other hand, must be a discrete attribute. This is a key characteristic that distinguishes classification from regression, a predictive modeling task in which y is a continuous attribute.

A classification model can also be used to predict the class label of unknown records and that automatically assigns a class label when presented with the attribute set of an unknown record. Suppose we are given the following characteristics of a record,

Today	Cool	Sunny	Normal	False	?
Tomorrow	Mild	Sunny	Normal	False	?

It can use a classification model built from the dataset shown in Table 2.1 to determine the class to which belongs.

2.3.1 General Approach to Solve Classification Problem

A Classification technique is a systematic approach to building classification models from an input data set. Examples include decision tree classifiers, rule-based classifiers, neural networks, support vector machines, and naive Bayes classifiers. Each technique employs a learning algorithm to identify a model that best fits the relationship between the attributes set and class label of the input data. The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seen before. Therefore, a key objective of the learning algorithm is to build models with good generalization capability; i.e., models that accurately predict the class labels of previously unknown records.

General approach for solving classification problems is shown in Figure 2.1. First, a training set consisting of records whose class labels are known must be provides. The training set is used to build a classification model, which is subsequently applied to the test set, which consists of records with unknown class labels [3].

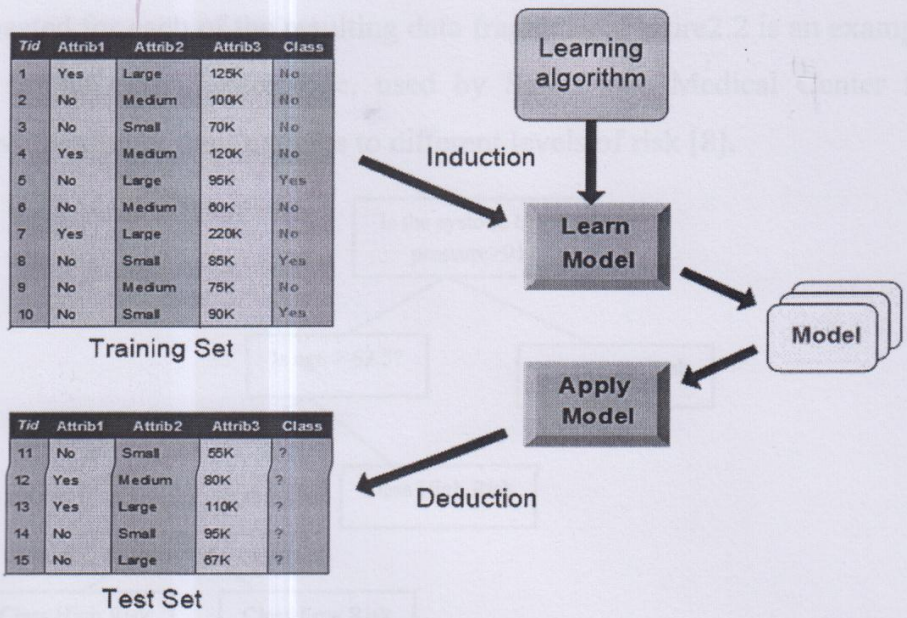


Figure 2.1 General Approach for Building a Classification Model

2.4 Decision Tree

A decision tree is a hierarchical structure of nodes and directed edges. Decision trees are powerful and popular tools for classification and prediction. Decision trees represent rules, which can be understood by humans and used in knowledge system such as database. Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables [7].

Decision trees are represented by a set of questions which splits the learning sample into smaller and smaller parts. CART asks only yes/no questions. A possible question could be: "Is age greater than 50?" or "Is sex male?". CART algorithm will search for all possible variables and all possible values in order to find the best split - the question that splits the data into two parts with maximum homogeneity. The process is then

repeated for each of the resulting data fragments. Figure2.2 is an example of simple classification tree, used by San Diego Medical Center for classification of their patients to different levels of risk [8].

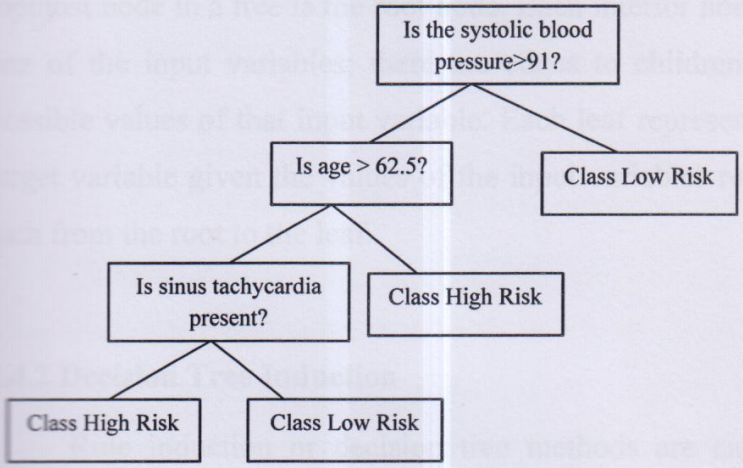


Figure 2.2 Classification Tree of San Diego Medical Center Patient

A decision tree is a tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a decision [9]. Decision tree are commonly used for gaining information for the purpose of decision making. Decision tree starts with a root node which is for users to take actions. From this node, users split each node recursively according to decision tree learning algorithm. The final result is a decision tree in which each branch represents a possible scenario of decision and its outcome.

2.4.1 Classification by Decision Tree Induction

Decision Tree Induction is the learning of decision trees from class labeled training tuples. Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables.

A decision tree is a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf nodes represent class labels. The topmost node in a tree is the root node. Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

2.4.2 Decision Tree Induction

Rule induction or decision tree methods are capable of culling through a set of predictors by successively splitting a data set into subgroups on the basis of the relationships between predictors and the output field.

The tree has three types of nodes:

- A root node that has no incoming edges and zero or more outgoing edges.
- Internal nodes, each of which has exactly one incoming edge and two or more outgoing edges.
- Leaf or terminal nodes, each of which has exactly one incoming edge and no outgoing edges

In a decision tree, each leaf node is assigned a class label. The non-terminal nodes, which include the root and other internal nodes, contain attribute test conditions to separate records that have different characteristics.

Decision tree classifier is a simple widely used classification technique. Classifying a test record is straightforward once a decision tree

has been constructed. Starting from the root node, we apply the test condition to the record and follow the appropriate branch based on the outcome of the test. This will lead us either to another internal node, for which a new test condition is applied, or to a leaf node. The class label associated with the leaf node is then assigned to the record.

2.4.3 How to Construct a Decision Tree

Decision tree learning is the construction of a decision tree from class-labeled training tuples. A decision tree is a flow-chart-like structure, where each internal (non-leaf) node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf (or terminal) node holds a class label. The topmost node in a tree is the root node [11]. In principle, there are many decision trees that can be constructed from a given set of attributes. Efficient algorithms have been developed to induce a reasonably accurate, suboptimal, decision tree in a reasonable amount of time. These algorithms usually employ a greedy strategy that grows a decision tree by making a series of locally optimum decisions about which attribute to use for partitioning the data.

The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification. There are many specific decision-tree algorithms [9]. Notable ones include:

- ID3 (Iterative Dichotomiser 3)
- C4.5 (successor of ID3)

- CART (Classification And Regression Tree)
- CHAID (CHi-squared Automatic Interaction Detector).
- MARS: extends decision trees to handle numerical data better.

ID3, C4.5 and CART adopt a greedy approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner. Most algorithms for decision tree induction also follow such a top-down approach, which starts with a training set of tuples and their associated class label. The training set is recursively partitioned into smaller subsets as the tree is being built. A basic decision tree algorithm is summarized as below.

Basic algorithm (a greedy algorithm)

- Tree is constructed in a top-down recursive divide-and-conquer manner
- At start, all the training examples are at the root
- Attributes are categorical (if continuous-valued, they are discretized in advance)
- Examples are partitioned recursively based on selected attributes
- Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)

Conditions for stopping partitioning

- All samples for a given node belong to the same class
- There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
- There are no samples left

A tree can be learned by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset

in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions. This process of top-down induction of decision trees (TDIDT) is an example of a greedy algorithm, and it is by far the most common strategy for learning decision trees from data.

2.4.4 Design Issues of Decision Tree Induction

A learning algorithm for inducing decision trees must address the following two issues [2].

1. How should the training records be split? Each recursive step of the tree growing process must select an attribute test condition to divide the records into smaller subsets. To implement this step, the algorithm must provide a method for specifying the test condition for different attribute types as well as an objective measure for evaluating the goodness of each test condition.
2. How should the splitting procedure stop? A stopping condition is needed to terminate the tree-growing process. A possible strategy is to continue expanding a node until either all the records belong to the same class or all the records have identical attribute values.

2.5 Methods for Expressing Attribute Test Conditions

Decision tree induction algorithms must provide a method for expressing an attribute test condition and its corresponding outcomes for different attribute types.

2.5.1 Binary Attributes

The test condition for a binary attribute generates two potential outcomes, as shown in Figure 2.3.

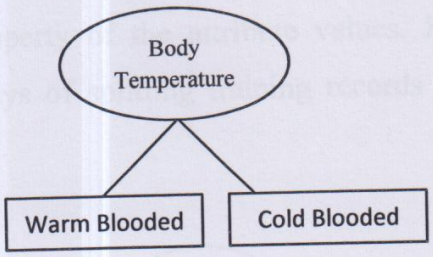


Figure 2.3 Binary Split

2.5.2 Nominal Attributes

Since a nominal attribute can have many values, its test condition can be expressed in two ways, as shown in Figure 2.4 and 2.5. For a multiway split as Figure 2.4, the number of outcomes depends on the number of distinct values for the corresponding attribute. For example, if an attribute such as marital status has three distinct values—single, married, or divorced—its test condition will produce a three-way split. On the other hand, some decision tree algorithms, such as CART, produce only binary splits as shown in Figure 2.5.

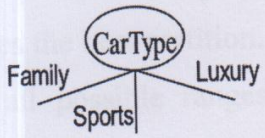


Figure 2.4 Multi-way Split

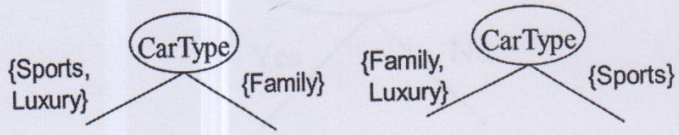


Figure 2.5 Binary Split for Nominal Attributes

2.5.3 Ordinal Attributes

Ordinal attributes can also produce binary or multiway splits. Ordinal attribute values can be grouped as long as the grouping does not violate the order property of the attribute values. Figure 2.6 and 2.7 illustrates various ways of splitting training records based on the Shirt Size attribute [2].

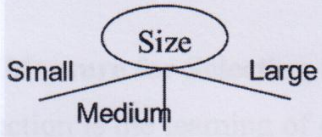


Figure 2.6 Multi-way Split for Ordinal Attributes

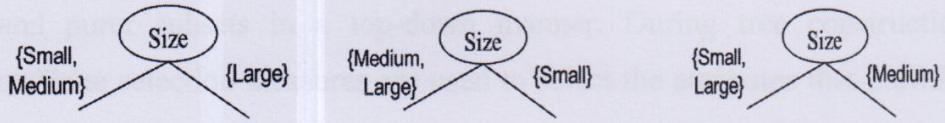


Figure 2.7 Binary Split for Ordinal Attributes

2.5.4 Continuous Attributes

In continuous attribute, for the binary case, the decision tree algorithm must consider all possible split positions as in Figure 2.8 and it selects the one that produces the best partition. For the multiway split, the algorithm must consider all possible ranges of continuous values as shown in Figure 2.9.

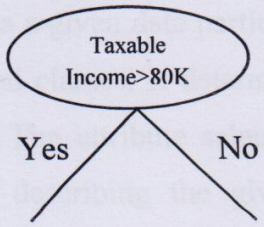


Figure2.8 Binary Split for Continuous Attributes

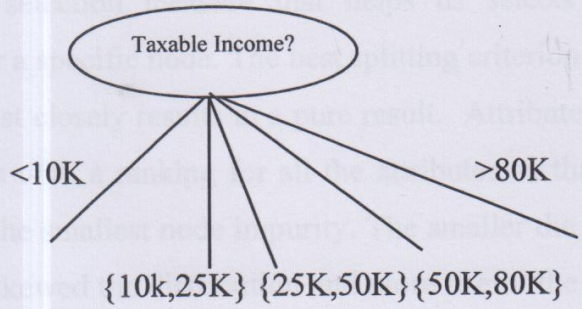


Figure 2.9 Multi- way Split for Continuous Attributes

2.6 Attribute Selection Measure for Selecting the Best Split

Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree model is build using the data at hand that is the training data, by successively splitting the data into purer and purer subsets in a top-down manner. During tree construction, attribute selection measures are used to select the attributes that partition the tuples into distinct classes. There are many measures that can be used to determine the best way to split the records. These measures are defined in terms of the class distribution of the records before and after splitting. The measures developed for selecting the best split are often based on the degree of impurity of the child nodes. The smaller the degree of impurity, the more skewed the class distribution. The quality of any potential split of the data is measured by one of a handful of split quality measures such as the Gini index or the entropy measure.

Attribute selection measure is a heuristic for selecting the splitting criterion that "best" separates a given data partition, *D*, of a class-labeled training tuples into individual classes. It determines how the tuples at a given node are to be split. The attribute selection measure provides a ranking for each attribute describing the given training tuples. The attribute having the best score for the measure is chosen as the splitting attribute for the given tuples.

An attribute selection measure that helps us selects the best splitting criterion for a specific node. The best splitting criterion would be the attribute that most closely results in a pure result. Attribute selection measures provide us with a ranking for all the attributes so that we can select the one with the smallest node impurity. The smaller the degree of impurity, the more skewed the distribution and more useful the split is to isolating the data sets into unique classes.

Different algorithms use different metrics for measuring "best". These generally measure the homogeneity of the target variable within the subsets. These metrics are applied to each candidate subset, and the resulting values are combined (e.g., averaged) to provide a measure of the quality of the split. The quality of any potential split of the data is measured by one of a handful of split quality measures. The rest of this section covers three attribute selection measures that it focus such as

- Information Gain
- Gini index
- Classification error

2.6.1 Information Gain

The Information gain is used to select the splitting attribute in each node in the tree. The attribute with the highest information gain is chosen as the splitting attribute for the current node.

This measure is based on pioneering work by Claude Shannon on information theory, which studied the value or "information content" of message. Let node N represents or hold the tuple of partition D. The attribute with the highest information gain is chosen as the splitting attribute for the node N. The expected information needed to classify a tuple in D is given by,

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

Where Pi is the probability, that an arbitrary tuple in D belongs to class Ci and is estimated by |Ci,D| / |D|. Info (D) is the average amount of information needed to identify the class label of a tuple in D. Info (D) is also known as the entropy of D. The expected information required to classify a tuple from D, based on the partitioning by attribute A is calculated by,

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

Information gain is defined as the difference between the original information requirement (i.e. based on the classes) and the new requirement (i.e. obtained after partitioning on A)

$$\text{Gain (A)} = \text{Info (D)} - Info_A(\text{D}) \qquad (2.1)$$

2.6.2 Gini Index

Gini inequality index (Gini, 1939) represents one of the most used and widespread inequality measures, both in methodological studies and in applied researches. The Gini index measures consider binary split for each attribute. The attribute with the minimum Gini index is selected as the splitting attribute. The Gini Index measures the impurity of D, a data partition or set of training tuples as,

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

Where Pi is the probability that a tuple in D belongs to Class Ci and is estimated by |Ci, D| / |D|.

When considering a binary split, we compute a weighted sum of the impurity of each resulting partition. For example, if a binary split on A partitions D into D1 and D2, the Gini index of D given that partitioning is

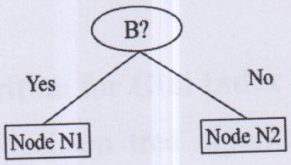
$$Gini_A(D) = \frac{|D1|}{|D2|} \text{Gini (D1)} + \frac{|D2|}{|D|} \text{Gini (D2)}$$

$$\Delta \text{ Gini (A)} = \text{Gini (D)} - Gini_A(D) \qquad (2.2)$$

For each attribute, each of the possible binary split is considered. For a discrete valued attribute, the subset that gives the minimum Gini index for that attribute is as its splitting attribute.

For continuous-valued attributes, each possible split point must be considered. The strategy is similar to information gain, where the midpoint between each pair of sorted adjacent values is taken as a possible split point. The point giving the minimum Gini index for a continuous-valued attribute is taken as the split point of that attribute. For a possible split point of A, D1 is the set of tuples in D satisfying A ≤ split point, and D2 is the set of tuples in D satisfying A > split point[3].

(a) Binary Attributes: Computing GINI Index



	Parent
C1	6
C2	6
Gini = 0.500	

Gini(N1)= 1- $(\frac{5}{6})^2$- $(\frac{2}{6})^2$

= 0.194

Gini(N2)= 1- $(\frac{1}{6})^2$-$(\frac{5}{6})^2$

= 0.528

Gini(Children)= 7/12 * 0.194 + 5/12 * 0.528

= 0.333

	N1	N2
C1	5	1
C2	2	4
Gini=0.333		

(b) Categorical Attributes: Computing Gini Index

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini = 0.393			

Multi-way split

	CarType	
	{Spotrs,Luxury}	{Family}
C1	3	1
C2	2	4
Gini = 0.400		

	CarType	
	{Sports}	{Family,Luxury}
C1	2	2
C2	1	5
Gini = 0.419		

Two-way split

(find best partition of values)

2.7 Algorithm for Gini Index Decision Tree

A decision tree algorithm called **Gini Index Decision Tree** is shown in Figure 2.10. The input to this algorithm consists of the training database D.

Input: The training database D

Output: A decision tree

Step1:

Create a node N

If D are all of the same class C then return N as a leaf node with the class C.

If D has no Non-label attribute then return N as a leaf node with the most common class.

Step2:

Select an attribute, say A, with the minimum Gini index value. Label node N with A.

Step3:

Partition the database D into subsets D1,D2 with respect to the attribute A.

Step4:

For each value ai of A Grow a branch from node N for the condition A_i.

If D_i is empty then attach a leaf labeled with the most common class in D.

Else attach the node returned by Gini Index(D_i)

Figure 2.10 Gini Index Decision Tree Algorithm

CHAPTER 3

SYSTEM DESIGN

3.1 Use Case Diagram

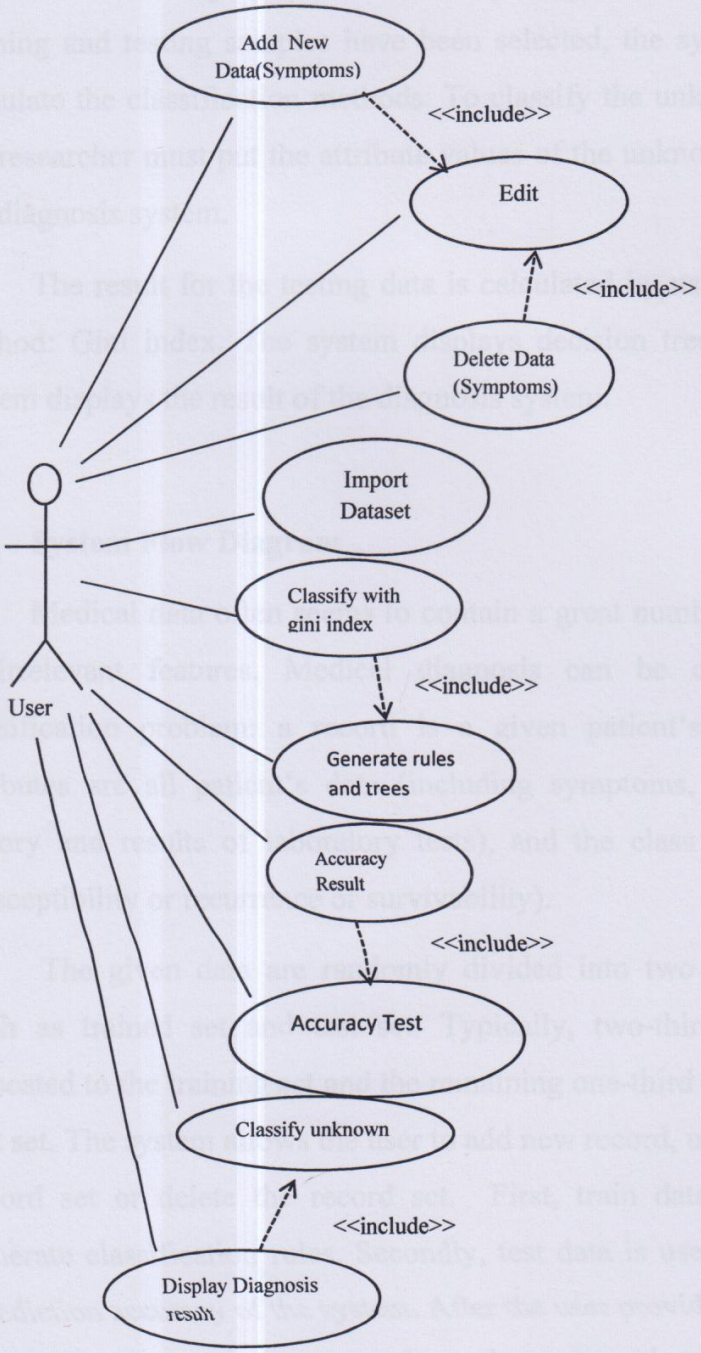


Figure 3.1 Use Case Diagram for Disease Diagnosis System

The user can add, delete and modify sample data record sets as necessary for disease diagnosis system. Testing sample record sets are used to test the accuracy of the classification system. The user may view the selected training samples and testing samples record sets. Once the training and testing samples have been selected, the system is ready to calculate the classification methods. To classify the unknown record set, the researcher must put the attribute values of the unknown symptoms of the diagnosis system.

The result for the testing data is calculated by using classification method: Gini index. The system displays decision tree and rules. The system displays the result of the diagnosis system.

3.2 System Flow Diagram

Medical data often seems to contain a great number and uncertain or irrelevant features. Medical diagnosis can be considered as a classification problem: a record is a given patient's case, predictor attributes are all patient's data (including symptoms, signals, clinical history and results of laboratory tests), and the class is the diagnosis (susceptibility or recurrence or survivability).

The given data are randomly divided into two independent sets such as trained set and test set. Typically, two-third of the data is allocated to the training set and the remaining one-third is allocated to the test set. The system allows the user to add new record, update the existing record set or delete the record set. First, train data is processed to generate classification rules. Secondly, test data is used to calculate the prediction accuracy of the system. After the user provided all the required attributes of the unknown record set, the system classify those unknown

record sets. The approach developed in this thesis is a method for split selection for classification trees based on the Gini index. Figure 3.2 shows the flow diagram of the system.

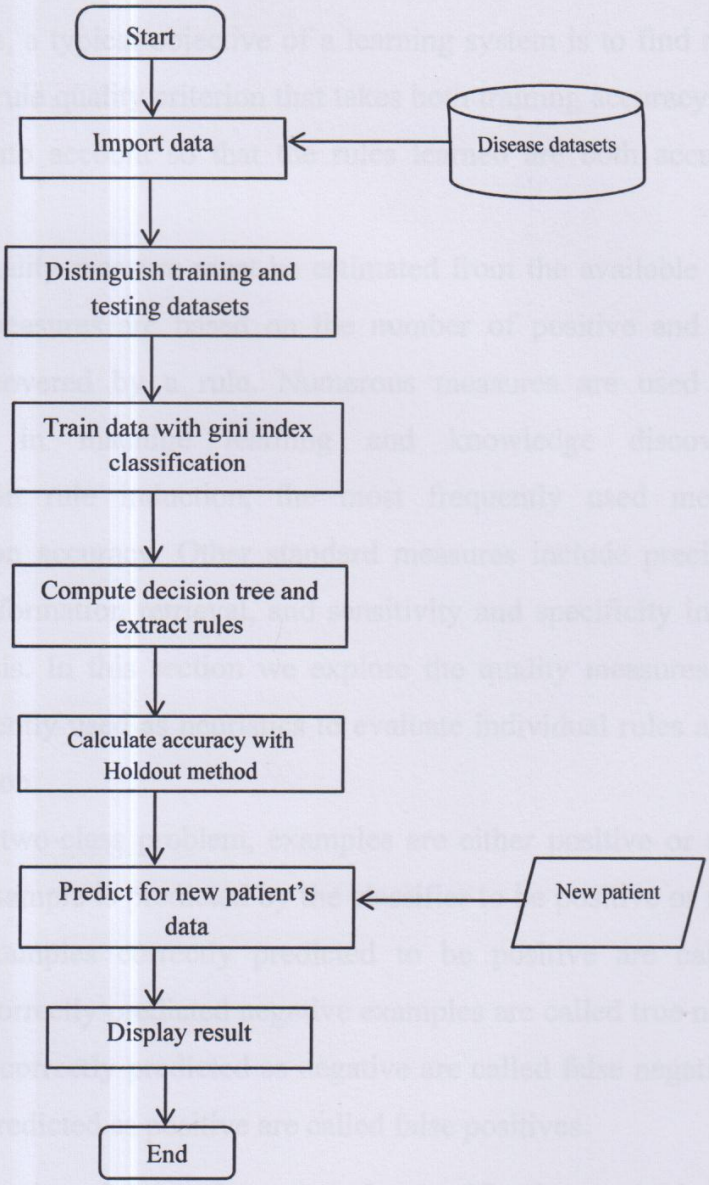


Figure 3.2 System Flow Diagram

3.3 Performance Evaluation

Given that the rule induction process could be conceived as a search process, a metric is needed to estimate the quality of rules found in the search space and to direct the search towards the best rule. The rule quality measure is a key element in rule induction. In real-world applications, a typical objective of a learning system is to find rules that optimize a rule quality criterion that takes both training accuracy and rule coverage into account so that the rules learned are both accurate and reliable.

A quality measure must be estimated from the available data. All common measures are based on the number of positive and negative instances covered by a rule. Numerous measures are used for rule evaluation in machine learning and knowledge discovery. In classification rule induction, the most frequently used measure is classification accuracy. Other standard measures include precision and recall in information retrieval, and sensitivity and specificity in medical data analysis. In this section we explore the quality measures that are most frequently used as heuristics to evaluate individual rules and guide rule induction.

In a two-class problem, examples are either positive or negative, and each example is predicted by the classifier to be positive or negative. Positive examples correctly predicted to be positive are called true positives, correctly predicted negative examples are called true negatives, positives incorrectly predicted as negative are called false negatives, and negatives predicted as positive are called false positives.

Evaluation of the performance of a classification model is based on the counts of test records correctly and incorrectly predicted by the model. One of the evaluation criteria is Classification accuracy [6]: this

refers to the ability of the model to correctly predict the class label of new or previously unseen data and can be express as;

Accuracy = (correct-record / total-record) * 100% (3.1)

3.3.1 Estimation of Classification Accuracy

In Holdout method, the given data are randomly partitioned 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 model, whose accuracy is estimated with the test set. The estimate is pessimistic because only a portion of the initial data is used to derive the model. In this thesis, we use holdout method to test the accuracy of the generate decision rules.

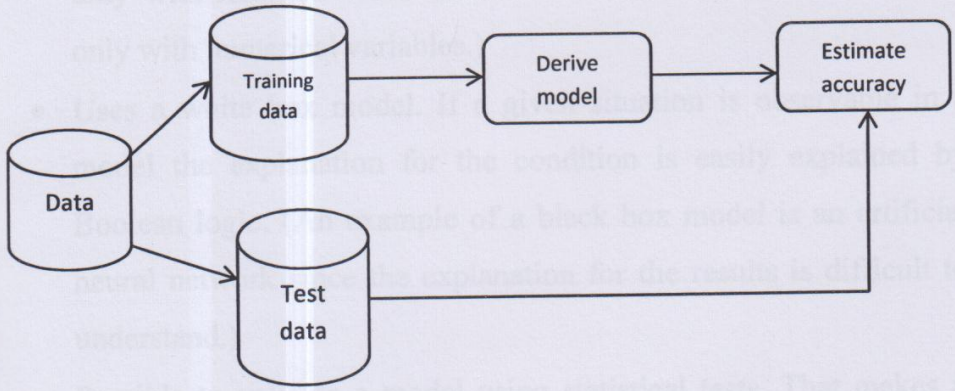


Figure 3.3 Estimating the accuracy with Holdout Method

3.4 Advantages of Decision Tree

Decision trees are popular as predictive models because of their intuitiveness and competitive performance with respect to other model building methodologies.

Among other data mining methods, decision trees have various advantages:

- Simple to understand and interpret. People are able to understand decision tree models after a brief explanation.
- Requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed.
- Able to handle both numerical and categorical data. Other techniques are usually specialized in analyzing datasets that have only one type of variable. (For example, relation rules can be used only with nominal variables while neural networks can be used only with numerical variables.)
- Uses a white box model. If a given situation is observable in a model the explanation for the condition is easily explained by Boolean logic. (An example of a black box model is an artificial neural network since the explanation for the results is difficult to understand.)
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
- Robust. Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.
- Performs well with large datasets. Large amounts of data can be analyzed using standard computing resources in reasonable time.

3.5 Algorithm for Decision Tree Induction

A skeleton decision tree induction algorithm called TreeGrowth is shown in Figure 3.4. The input to this algorithm consists of the training records E and the attribute set F. The algorithm works by recursively selecting the best attribute to split the data and expanding the leaf nodes of the tree until the stopping criterion is met.

```
TreeGrowth (E,F)
1: if stopping_cond (E,F) = true then
2:     leaf = createNode( ).
3:     leaf.label = Classify (E).
4:     return leaf.
5: else
6:     root = createNode( ).
7:     root.test_cond = GiniIndex(E,F).
8:     Let V= {v| v is a possible outcome of root.test_cond}.
9:     for each v € V do
10:        E_v = {e| root.test_cond(e) = v and e € E }.
11:        child = TreeGrowth (E_v , F).
12:        add child as descendent of root and label the edge(root- child)as v
13:    end for
14: end if
15: return root.
```

Figure 3.4 A skeleton decision tree induction algorithm

3.6 Example Decision Tree Using Diabetes Data

Example of diabetes data has 50 tuples. 30 tuples are used for train data and the rest of 20 is used for testing data. Train data are analyzed by Gini index classification method. The attribute insulin has the minimum Gini index and therefore becomes the splitting attribute at the root node of the decision tree. Branches are grown for each outcome of insulin and

the tuples are partitioned accordingly. The final decision tree is shown in Figure 3.5.

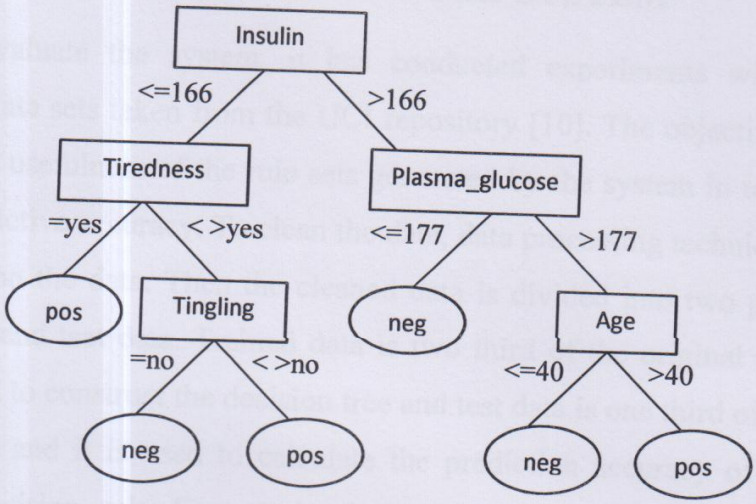


Figure 3.5 Decision tree for diabetes data

From the tree, the following rules are induced.

1. IF Insulin <= 166 AND Tiredness = "yes" THEN Class = positive

2. ELSE IF Insulin <= 166 AND Tiredness <> "yes" AND Tingling = "no" THEN Class = negative

3. ELSE IF Insulin <= 166 AND Tiredness <> "yes" AND Tingling <> "no" THEN Class = positive

4. ELSE IF Insulin > 166 AND Plasma_glucose_concentration <= 177 THEN Class = negative

5. ELSE IF Insulin > 166 AND Plasma_glucose_concentration > 177 AND Age <= 40 THEN Class = negative

6. ELSE IF Insulin > 166 AND Plasma_glucose_concentration > 177 AND Age > 40 THEN Class = positive

CHAPTER 4

IMPLEMENTATION OF THE SYSTEM

To evaluate the system, it has conducted experiments with a number of data sets taken from the UCI repository [10]. The objective is to check the usefulness of the rule sets generated by the system in terms of their predictive accuracy. To clean the data, data processing techniques are applied on the data. Then the cleaned data is divided into two parts trained data and test data. Trained data is two third of the original data and it is used to construct the decision tree and test data is one third of the original data and it is used to calculate the prediction accuracy of the generated decision rules.First, train dataset is processed to construct decision tree by using Gini Index approach. Secondly, test dataset is used to calculate the prediction accuracy of the algorithm; the result is shown by tables with the classifier accuracy.

4.1 Experimental Discussion

The main objective of this thesis is to select the best attribute measure to construct decision tree. To evaluate the system performances on Heart and Diabetes, datasets collected from the UCI data repository. Heart and Diabetes Disease datasets are classical datasets which can be easily used for decision trees.

Accuracy is measured using the hold-out method of a given datasets into disjoint train and test sets. In each experiment, an algorithm is trained on the training dataset and the induced theory is evaluated on the test set. Table 4.1 shows the train and test set size used with the heart and diabetes diseases.

Table 4.1 Training and Test Set Size of Heart and Diabetes Diseases

Datasets	Training set	Testing set	Total
Heart	140	69	209
Diabetes	160	80	240

The main characteristics of the datasets used in experiments are presented in Table 4.2 and Table 4.3. There are 209 instances for heart and 240 instances for diabetes. Heart dataset has eleven features and diabetes has ten features.

Table 4.2 Attribute Information for Heart Disease

No	Attribute Name	Description
1	Age	
2	Sex	(male, female)
3	Blood sugar	(1 if >120 mg/dl, 0 otherwise)
4	Chest pain	(asympt, atyp_angina, non_anginal, typ_angina)
5	Blood pressure	(92-200)
6	Heart rate	(82-188)
7	Nausea	(yes, no)
8	Dizziness	(yes, no)
9	Fatigue	(yes, no)
10	Sweating	(yes, no)
11	Exercise angina	(yes no)

Table 4.3 Attribute Information for Diabetes Disease

No	Attribute Name	Description
1	Age	
2	Dry skin	(yes, no)
3	Frequent urination	(yes, no)
4	Tingling	(yes, no)
5	Weight loss	(yes, no)
6	Tiredness	(yes, no)
7	Extreme hunger	(yes, no)
8	Diastolic blood pressure	(28-110)
9	Plasma glucose concentration	(56-198)
10	Insulin	(14-846)

4.1.1 Generate Rules for Heart Disease

The data for heart disease are randomly partitioned into two third of training and one third of testing data by hold-out method. The following rules are presented for Case Study 1 and have 16 rules. Next, the same data set of heart disease is partitioned into different training and testing data for Case Study 2 and Case Study 3. And then, by using the system, Case Study 2 has 19 rules and Case Study 3 has 24 rules.

1. IF exercice_angina = "yes" AND max_heart_rate <= 159 AND chest_pain = "non_anginal" THEN disease = negative

2. ELSE IF exercice_angina = "yes" AND max_heart_rate <= 159 AND chest_pain <> "non_anginal" AND Fatigue = "yes" THEN disease = positive

3. ELSE IF exercice_angina = "yes" AND max_heart_rate <= 159 AND chest_pain <> "non_anginal" AND Fatigue <> "yes" AND Nausea = "no" AND Sweating = "no" AND age <= 49 THEN disease = positive

4. ELSE IF exercice_angina = "yes" AND max_heart_rate <= 159 AND chest_pain <> "non_anginal" AND Fatigue <> "yes" AND Nausea = "no" AND Sweating = "no" AND age > 49 AND blood_sugar = "t" THEN disease = positive

5. ELSE IF exercice_angina = "yes" AND max_heart_rate <= 159 AND chest_pain <> "non_anginal" AND Fatigue <> "yes" AND Nausea = "no" AND Sweating = "no" AND age > 49 AND blood_sugar <> "t" THEN disease = negative

6. ELSE IF exercice_angina = "yes" AND max_heart_rate <= 159 AND chest_pain <> "non_anginal" AND Fatigue <> "yes" AND Nausea = "no" AND Sweating <> "no" THEN disease = negative

7. ELSE IF exercice_angina = "yes" AND max_heart_rate <= 159 AND chest_pain <> "non_anginal" AND Fatigue <> "yes" AND Nausea <> "no" THEN disease = positive

8. ELSE IF exercice_angina = "yes" AND max_heart_rate > 159 THEN disease = negative

9. ELSE IF exercice_angina <> "yes" AND max_heart_rate <= 131 AND age <= 53 AND Sex = "male" THEN disease = positive

10. ELSE IF exercice_angina <> "yes" AND max_heart_rate <= 131 AND age <= 53 AND Sex <> "male" AND rest_bpress <= 124 THEN disease = positive

11. ELSE IF exercice_angina <> "yes" AND max_heart_rate <= 131 AND age <= 53 AND Sex <> "male" AND rest_bpress > 124 AND chest_pain = "non_anginal" THEN disease = positive

12. ELSE IF exercice_angina <> "yes" AND max_heart_rate <= 131 AND age <= 53 AND Sex <> "male" AND rest_bpress > 124 AND chest_pain <> "non_anginal" THEN disease = negative

13. ELSE IF exercice_angina <> "yes" AND max_heart_rate <= 131 AND age > 53 THEN disease = negative

14. ELSE IF exercice_angina <> "yes" AND max_heart_rate > 131 AND chest_pain = "typ_angina" AND rest_bpress <= 130 THEN disease = negative

15. ELSE IF exercice_angina <> "yes" AND max_heart_rate > 131 AND chest_pain = "typ_angina" AND rest_bpress > 130 THEN disease = positive

16. ELSE IF exercice_angina <> "yes" AND max_heart_rate > 131 AND chest_pain <> "typ_angina" THEN disease = negative

4.1.2 Generate Rules for Diabetes Disease

Similarity the data for diabetes are randomly different partitioned for Case Study 1, 2 and 3. Case Study 1 has 32 rules, Case Study 2 has 27 rules and Case Study 3 has 31 rules. The following are the induced rules for Case Study 1.

1. IF Plasma_glucose_concentration <= 127 AND Tingling = "no" AND Tiredness = "no" AND Diastolic_blood_pressure <= 59 AND Age <= 45 THEN Class = negative

2. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling = "no" AND Tiredness = "no" AND Diastolic_blood_pressure <= 59 AND Age > 45 THEN Class = positive

3. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling = "no" AND Tiredness = "no" AND Diastolic_blood_pressure > 59 THEN Class = negative

4. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling = "no" AND Tiredness <> "no" AND Diastolic_blood_pressure <= 51 AND Age <= 43 THEN Class = positive

5. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling = "no" AND Tiredness <> "no" AND Diastolic_blood_pressure <= 51 AND Age > 43 THEN Class = negative

6. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling = "no" AND Tiredness <> "no" AND Diastolic_blood_pressure > 51 AND Age <= 36 THEN Class = negative

7. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling = "no" AND Tiredness <> "no" AND Diastolic_blood_pressure > 51 AND Age > 36 AND Weight_loss = "yes" THEN Class = positive

8. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling = "no" AND Tiredness <> "no" AND Diastolic_blood_pressure > 51 AND Age > 36 AND Weight_loss <> "yes" AND Extreme_hunger = "yes" THEN Class = positive

9. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling = "no" AND Tiredness <> "no" AND Diastolic_blood_pressure > 51 AND Age > 36 AND Weight_loss <> "yes" AND Extreme_hunger <> "yes" AND Frequent_urination = "no" THEN Class = positive

10. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling = "no" AND Tiredness <> "no" AND Diastolic_blood_pressure > 51 AND Age > 36 AND Weight_loss <> "yes" AND Extreme_hunger <> "yes" AND Frequent_urination <> "no" THEN Class = negative

11. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling <> "no" AND Tiredness = "yes" THEN Class = positive

12. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling <> "no" AND Tiredness <> "yes" AND Dry_skin = "no" AND Diastolic_blood_pressure <= 69 AND Frequent_urination = "no" THEN Class = negative

13. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling <> "no" AND Tiredness <> "yes" AND Dry_skin = "no" AND Diastolic_blood_pressure <= 69 AND Frequent_urination <> "no" AND Age <= 47 THEN Class = negative

14. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling <> "no" AND Tiredness <> "yes" AND Dry_skin = "no" AND Diastolic_blood_pressure <= 69 AND Frequent_urination <> "no" AND Age > 47 THEN Class = positive

15. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling <> "no" AND Tiredness <> "yes" AND Dry_skin = "no" AND Diastolic_blood_pressure > 69 THEN Class = positive

16. ELSE IF Plasma_glucose_concentration <= 127 AND Tingling <> "no" AND Tiredness <> "yes" AND Dry_skin <> "no" THEN Class = positive

17. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness = "no" AND Diastolic_blood_pressure <= 65 AND Tingling = "no" THEN Class = negative

18. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness = "no" AND Diastolic_blood_pressure <= 65 AND Tingling <> "no" AND Extreme_hunger = "no" THEN Class = negative

19. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness = "no" AND Diastolic_blood_pressure <= 65 AND Tingling <> "no" AND Extreme_hunger <> "no" THEN Class = positive

20. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness = "no" AND Diastolic_blood_pressure > 65 AND Dry_skin = "yes" AND Insulin <= 197 THEN Class = negative

21. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness = "no" AND Diastolic_blood_pressure > 65 AND Dry_skin = "yes" AND *Insulin > 197 AND Weight_loss = "no" AND Age <= 47 AND* Tingling = "no" THEN Class = positive

22. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness = "no" AND Diastolic_blood_pressure > 65 AND Dry_skin = "yes" AND Insulin > 197 AND Weight_loss = "no" AND Age <= 47 AND Tingling <> "no" THEN Class = negative

23. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness = "no" AND Diastolic_blood_pressure > 65 AND Dry_skin = "yes" AND Insulin > 197 AND Weight_loss = "no" AND Age > 47 THEN Class = negative

24. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness = "no" AND Diastolic_blood_pressure > 65 AND Dry_skin = "yes" AND Insulin > 197 AND Weight_loss <> "no" AND Age <= 44 THEN Class = negative

25. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness = "no" AND Diastolic_blood_pressure > 65 AND Dry_skin = "yes" AND Insulin > 197 AND Weight_loss <> "no" AND Age > 44 THEN Class = positive

26. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness = "no" AND Diastolic_blood_pressure > 65 AND Dry_skin <> "yes" THEN Class = negative

27. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness <> "no" AND Tingling = "no" AND Age <= 65 AND Insulin <= 137 AND Weight_loss = "no" THEN Class = negative

28. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness <> "no" AND Tingling = "no" AND Age <= 65 AND Insulin <= 137 AND Weight_loss <> "no" THEN Class = positive

29. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness <> "no" AND Tingling = "no" AND Age <= 65 AND Insulin > 137 AND Diastolic_blood_pressure <= 84 THEN Class = negative

30. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness <> "no" AND Tingling = "no" AND Age <= 65 AND Insulin > 137 AND Diastolic_blood_pressure > 84 THEN Class = positive

31. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness <> "no" AND Tingling = "no" AND Age > 65 THEN Class = positive

32. ELSE IF Plasma_glucose_concentration > 127 AND Tiredness <> "no" AND Tingling <> "no" THEN Class = positive

4.1.3 Evaluation of the Rules

Numerous measures are used for rule evaluation in machine learning and knowledge discovery. In classification rule induction, the most frequently used measure is classification accuracy [6].

Test dataset is used to calculate the prediction accuracy for the algorithm. In order to evaluate the system performances, Heart dataset and diabetes datasets are collected. The Table 4.4 shows that the system has the overall better accuracy.

In order to evaluate the system performances, used some datasets collected from the UCI data repository. Experiments are done on dataset and report of experimental result is shown.

Table 4.4 Evaluation Result for Heart and Diabetes Diseases Datasets

Disease	Total Tuples	Training Tuples	Testing Tuples	Case Study 1	Case Study 2	Case Study 3
Heart	209	140	69	87%	74%	92%
Diabetes	240	160	80	89%	72%	85%

4.1.4 Evaluation using New Dataset: Diabetes

The new diabetes dataset is taken from real world data combining with data from UCI data repository [10]. There are 240 tuples and the evaluation result for this dataset in shown in table 4.5.

Table 4.5 Evaluation Result for New Diabetes Diseases Dataset

Disease	Total Tuples	Training Tuples	Testing Tuples	Case Study 1	Case Study 2	Case Study 3
Diabetes	240	160	80	75%	69%	70%

4.2 Classification of New Data

As the classification or regression tree is constructed, it can be used to classify of new data. The output of this stage is an assigned class or response value to each of the new observations. By set of questions in the tree, each of the new observations will get to one of the terminal nodes of the tree. A new observation is assigned with the dominating class/response value of terminal node, where this observation belongs to.

Table 4.4 Evaluation Result for Heart and Diabetes Diseases Datasets

Disease	Total Tuples	Training Tuples	Testing Tuples	Case Study 1	Case Study 2	Case Study 3
Heart	209	140	69	87%	74%	92%
Diabetes	240	160	80	89%	72%	85%

4.1.4 Evaluation using New Dataset: Diabetes

The new diabetes dataset is taken from real world data combining with data from UCI data repository [10]. There are 240 tuples and the evaluation result for this dataset in shown in table 4.5.

Table 4.5 Evaluation Result for New Diabetes Diseases Dataset

Disease	Total Tuples	Training Tuples	Testing Tuples	Case Study 1	Case Study 2	Case Study 3
Diabetes	240	160	80	75%	69%	70%

4.2 Classification of New Data

As the classification or regression tree is constructed, it can be used to classify of new data. The output of this stage is an assigned class or response value to each of the new observations. By set of questions in the tree, each of the new observations will get to one of the terminal nodes of the tree. A new observation is assigned with the dominating class/response value of terminal node, where this observation belongs to.

4.2.1 Classified Heart Disease and Diabetes Disease

Suppose there is a new patient with symptoms described in Table 4.6 and Table 4.7. The system would like to determine whether the patient has Heart Disease using the induced rules to classify the patient.

To classify this new data, it matches with the induced rules. Rule 14 is found to be matched with the sample data. Therefore, the new data is classified into class (Heart Disease = "Negative") in Table 4.6.

Table 4.6 A New Data for Heart Disease

1	2	3	4	5	6	7	8	9	10	11	Diagnosis
42	M	F	Ty	130	135	N	Y	Y	Y	N	?

To classify the following new data, the system matches the example with the induced rules from diabetes dataset. Rule 17 is found to be matched with the example. Therefore, the new example is classified into class (Diabetes Disease = "Negative") in Table 4.7.

Table 4.7 A New Data for Diabetes Disease

1	2	3	4	5	6	7	8	9	10	Diagnosis
58	N	N	N	Y	N	N	68	135	250	?

The system matches 20 same new unknown data with the induced rules for Case Study 1, 2 and 3 of heart and diabetes diseases. Then the results are tested for ground truth with an expert. In heart dataset, 14 data give the acceptable diagnosis and so it has the average accuracy of 70%. Similarity, diabetes data set has the average accuracy of 75%.

4.2.2 Classified Diabetes Disease using New Dataset

Table 4.8 Unknown Data for New Diabetes Disease

No	1	2	3	4	5	6	7	8	9	10	Diagnosis	True value
1.	50	N	Y	Y	Y	Y	Y	70	99	54	Positive	Positive
2.	66	N	N	Y	N	N	N	74	167	144	Negative	Negative
3.	43	Y	N	Y	N	N	N	88	174	120	Negative	Negative
4.	30	Y	N	Y	Y	N	Y	72	121	112	Positive	Positive
5.	39	Y	Y	N	Y	N	Y	74	126	75	Negative	Positive
. . .												

The unknown data is classified into the induced rules from diabetes by using new data set and found to be matched with into class (Diabetes disease = "Positive" and "Negative"). In Table 4.8, the system is diagnosed diabetes disease by twenty patients with 80% accuracy.

CHAPTER 5

CONCLUSION

5.1 Conclusion

Classification is important in medical diagnosis as it gives predictive accuracy and precise result [5]. Many classification and prediction methods have been used in machine learning, expert systems, statistics and engineering. Most algorithms are memory resident, typically assuming a small data size. Recent database mining research has built on such work, developing scalable classification and prediction techniques capable of handling large disk-resident data. Decision tree learning is a method commonly used in data mining. It is a simple representation for classifying examples and one of the most successful techniques for classification learning.

This system is developed by using Gini Index decision tree, a simple method. The given data are partitioned into two sets, a training set and test set in the holdout method. This system provides the accuracy depending to the testing data, ranging from. This system generates the classification rules from decision tree. By entering symptoms and signs, the user can know classified diseases. This system can help the medical experts to do better decision making in healthcare problems by using classification technique.

In summary, decision tree induction algorithms have been used for classification in a wide range of application domains. The learning and classification steps of decision tree are generally fast. Experimented with benchmark medical Diseases dataset shows the selection of attribute using Gini Index in decision tree construction generates comparable

result. This system can take precise of result about heart and diabetes diseases.

5.2 Advantages of the System

The decision tree methods have several advantages. Decision trees are one of the most well-established classification methods .In medical field, this system can provide heart and diabetes diseases patients with precise and optimal prediction of diseases. Furthermore, it can help clinicians for improving the quality of decisions in diagnosis.

5.3 Limitation and Further Extension

In this system, classification is based on eleven and ten input attributes of heart and diabetes diseases to diagnosis diseases. It can be extended to consider with many attributes for more precisely determine. This system can only be classified the probability of disease and further work will classify the kind of diseases. Possible extension of this work will be developed to use various attribute selection measures like Information Gain. It can also test more dataset and can be tested with other applications such as biology, engineering research and physical science.

REFERENCES

[1] A. AN, Department of Computer Science, York University, Toronto, "Learning Classification Rules from Data", In Proceedings of the International Joint Conference on Artificial Intelligence (2003).

[2] Jiawei Han and Micheline Kamber, "Chapter 4, Classification: Basic Concepts, Decision Trees and Model Evaluation".

[3] Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques", Second Edition 2006.

[4] Laura E. Raileanu and Kilian Stoffel, "Theoretical Comparison between the Gini Index and Information Gain Criteria", Proceedings of the 17th International Conference on Machine Learning.

[5] N. Suneetha, "Modified Gini Index Classification: A Case Study of Heart Disease Dataset", Proceedings of the National Conference on Artificial Intelligence .

[6] Peter Flach and Nada Lavra c, "Chapter 1: Rule Induction", Proceedings of the 8th International Conference on Inductive Logic Programming

[7] R. Aruna devi[1], Dr. K. Nirmala[2], "Construction of Decision Tree: Attribute Selection Measures", international Journal of

Advancements in Research & Technology, Volume 2, Issue 4, April-2013.

[8] Roman Timofeev, "Classification and Regression Trees (CART) Theory and Applications", A Master Thesis Presented by Master of Art Berlin, December 20, 2004.

[9] T. Miranda Lakshmi, A.Martin, R.Mumtaj Begum, Dr.V.Prasanna Venkatesan, I.J.Modern, "An Analysis on Performance of Decision Tree Algorithms using Student's Qualitative Data", Proceedings of the ITI 2007 29thInternational Conference on Information Technology Interfaces.

[10] https:// archive.ics.uci.edu/,_UCI Data Repository

[11] https://en.wikipedia.org/wiki/, Wikipedia

APPENDIXES

Introduction

This section explains about design and implementation of the system. This system is implement by using C#, Net programming Language. There are four main menus in the system; File, Classification, View and Windows.

Main Window

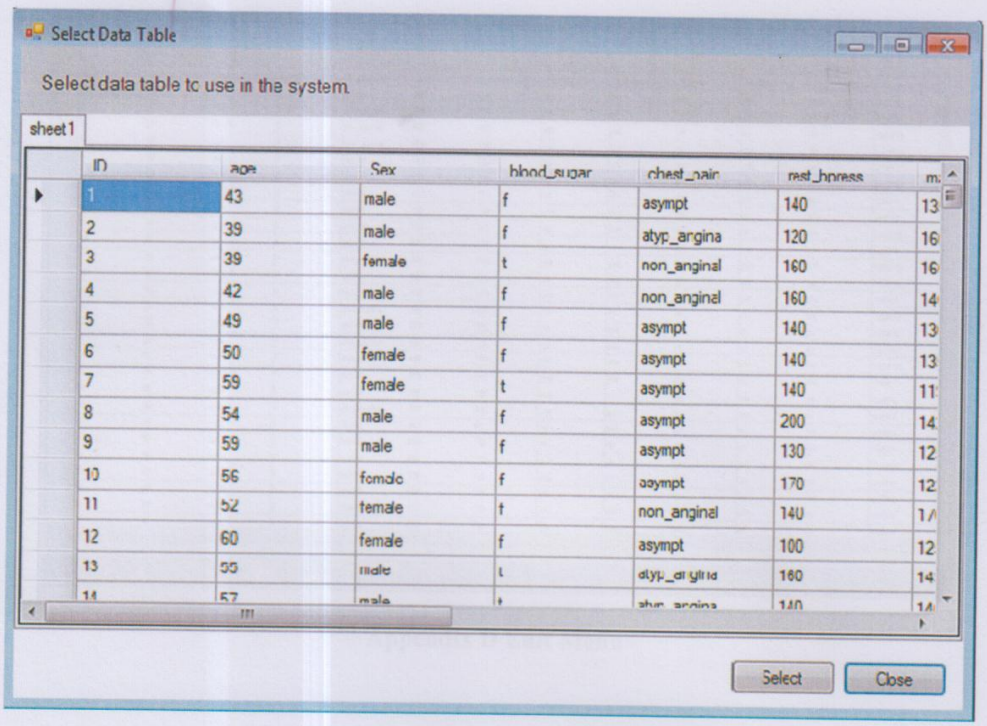
There are four main menu in this window; *File, Classification View* and *Windows*. File menu mainly serve to store training data and testing data. This data can be update by using this menu. Classification menu use to generate decision rules and classify unknown sample and also evaluate classifier accuracy. Windows menu is to see system related information in various views.

Appendix A Main Window

File Menu

This menu has three sub-menus; *Open, Edit* and *Exit* sub menus. Open and Edit menus use to select and update training and testing datasets and *Exit* menu use to exit from the system.

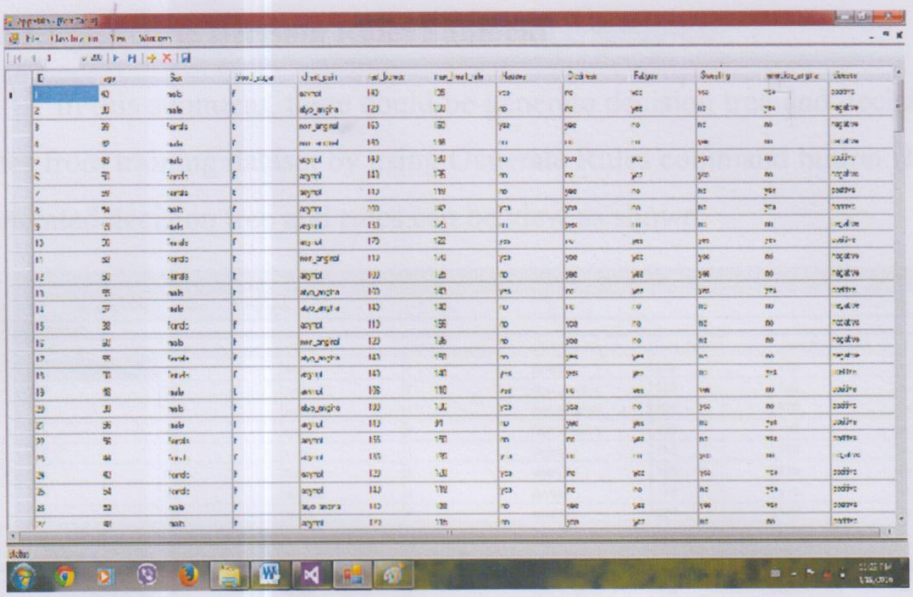
Appendix B File Menu



Appendix C Open Dataset Menu

Edit Submenu

By using this submenu user can update training data. User can add, delete and save operations in this windows by using the corresponding buttons.



Appendix D Edit Menu

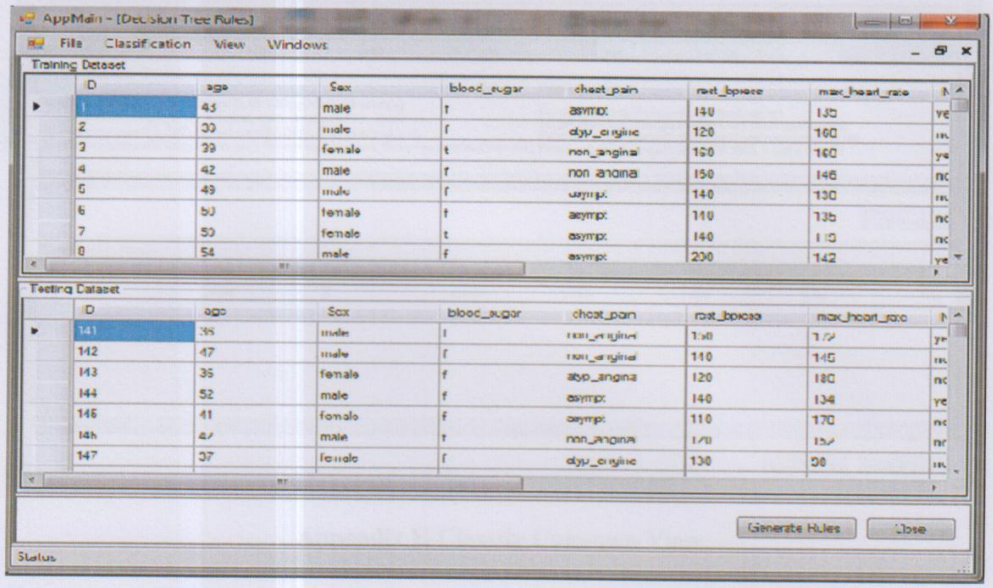
Classification Menu

In this menu there are three sub menu *Generate Decision Rules, Classify Unknown and Accuracy Test* as shown .

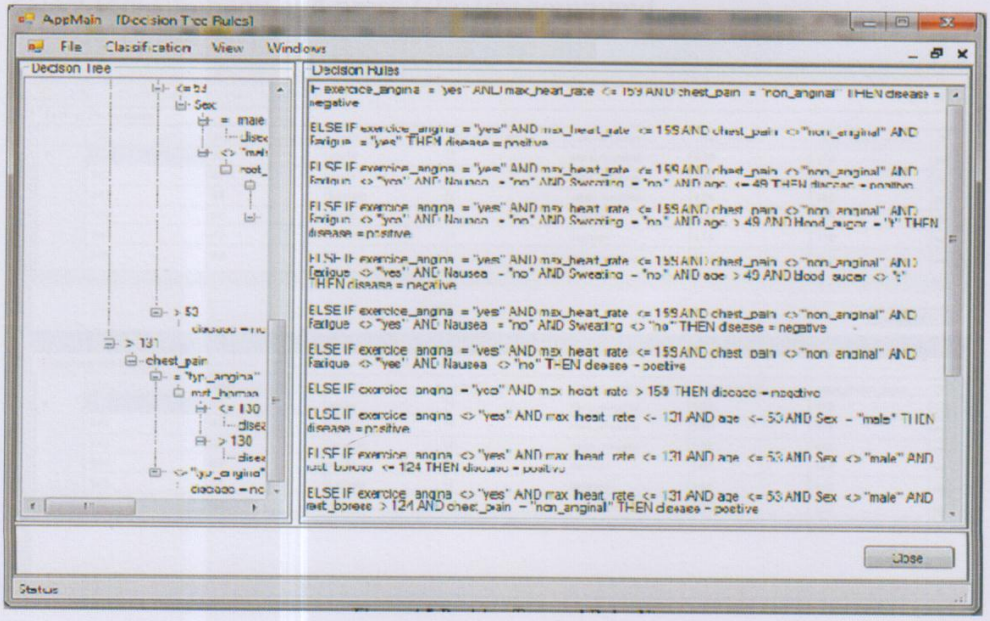
Appendix E Classification Menu

4.1.3.1 Generate Decision Rules Submenu

In this submenu, there could be generate decision tree and decision rules from training dataset by using Generate Rules command button. The generated decision tree and rules can be view as shown.



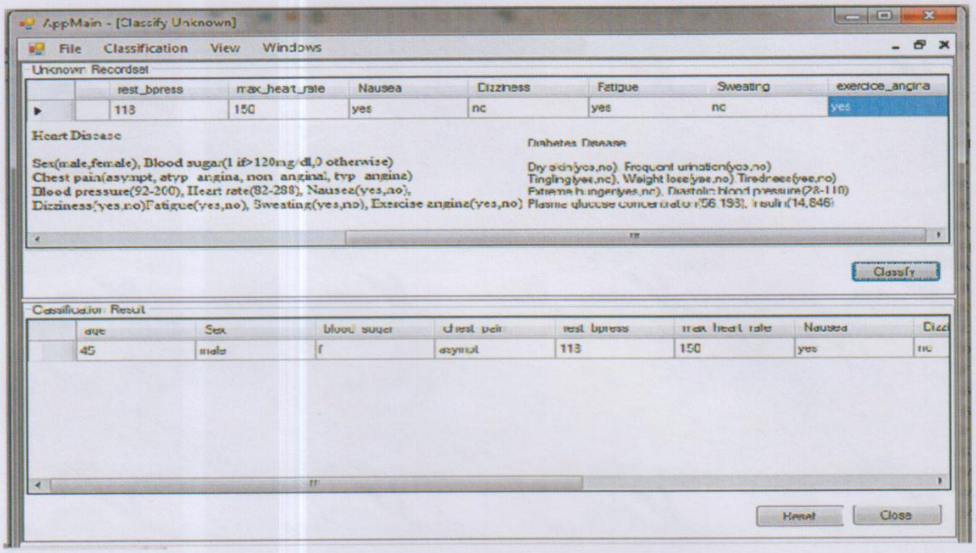
Appendix F Generate Decision Rules View



Appendix G Decision Tree and Rules View

Classify Unknown Submenu

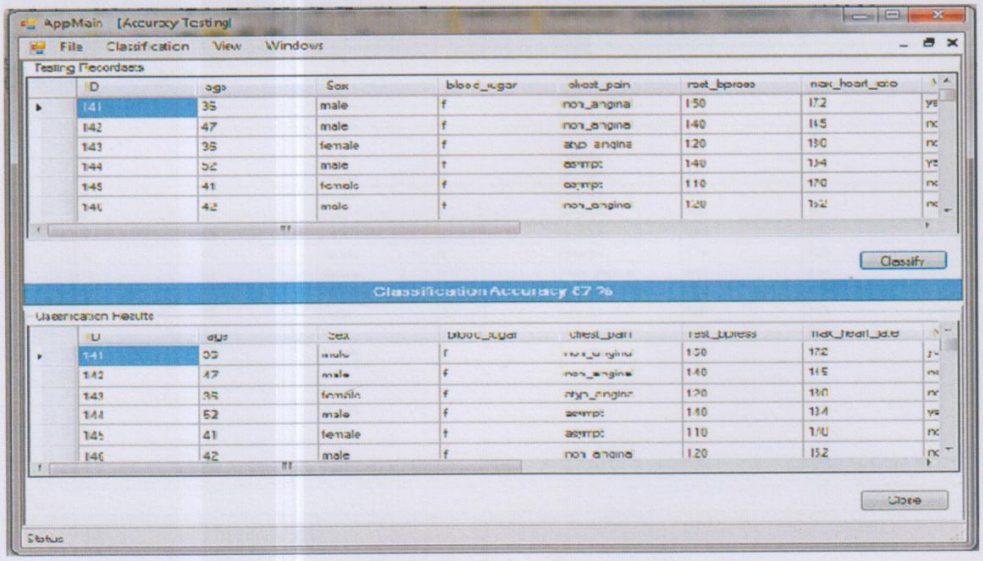
To classify unknown sample user must enter input values and user classify command button. The classified result can be known.



Appendix H Classify Unknown View

Accuracy Test Submenu

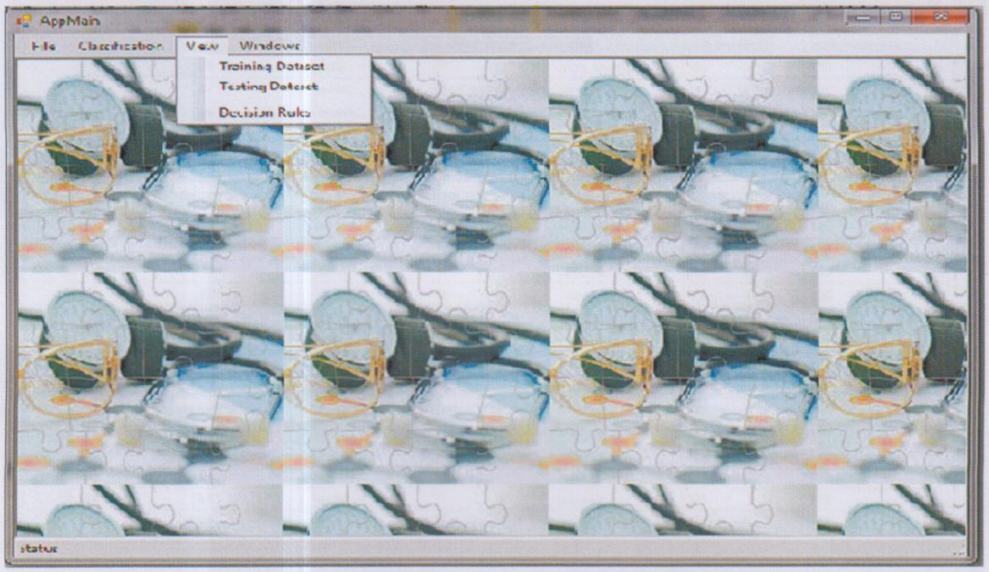
In order to test the accuracy of generate rules, the user need to select this submenu and press *Classify* command.



Appendix I Accuracy Test View

View Menu

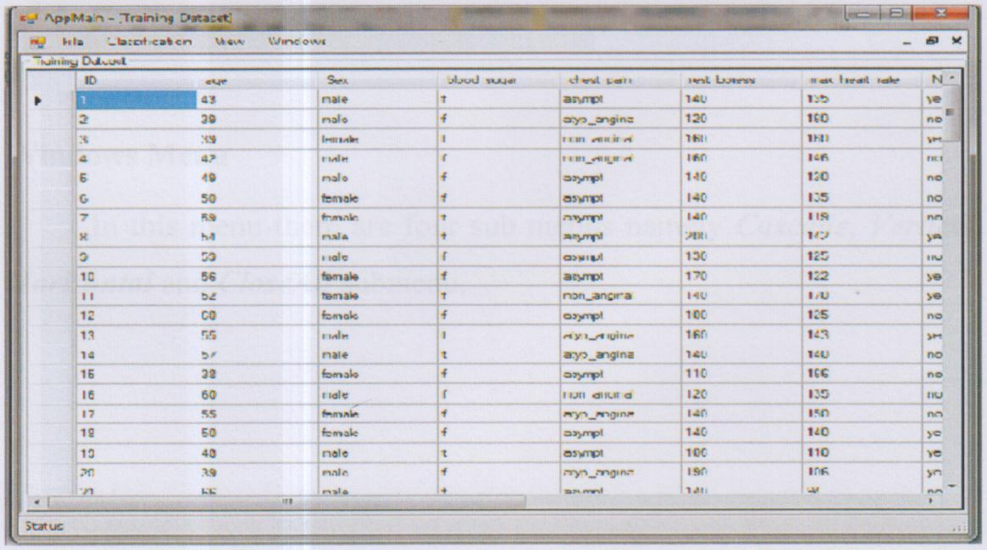
By using this menu the user can view training and testing dataset and generated decision rules.



Appendix J View Menu

Training Dataset Submenu

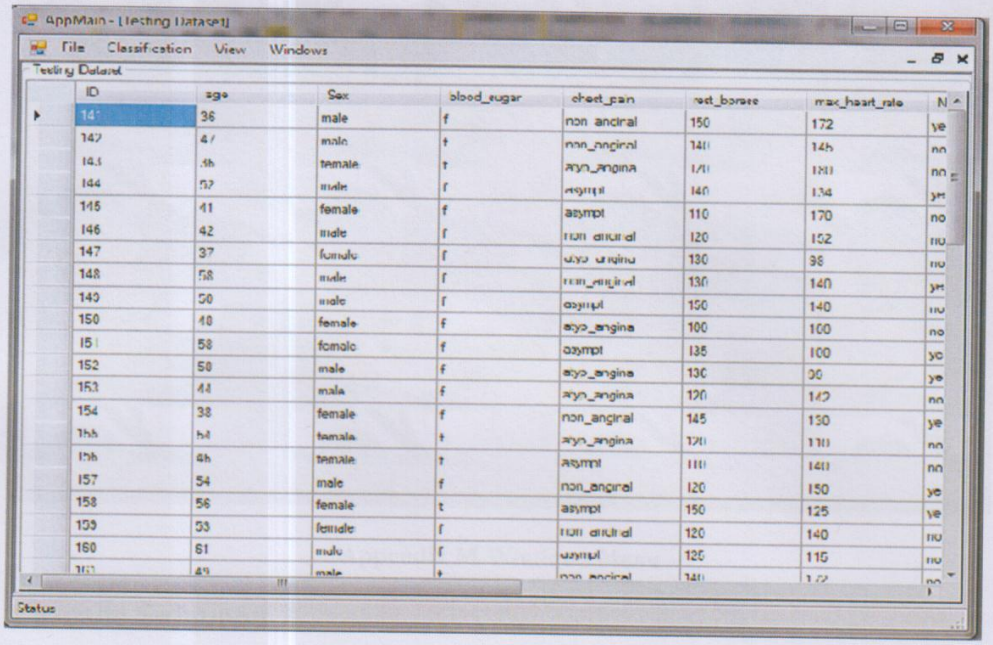
In order to view training dataset the user can select this submenu.



Appendix K Training Dataset View

Testing Dataset Submenu

In order to view testing dataset the user can select this submenu.



Appendix L Testing Dataset View

Decision Rules Submenu

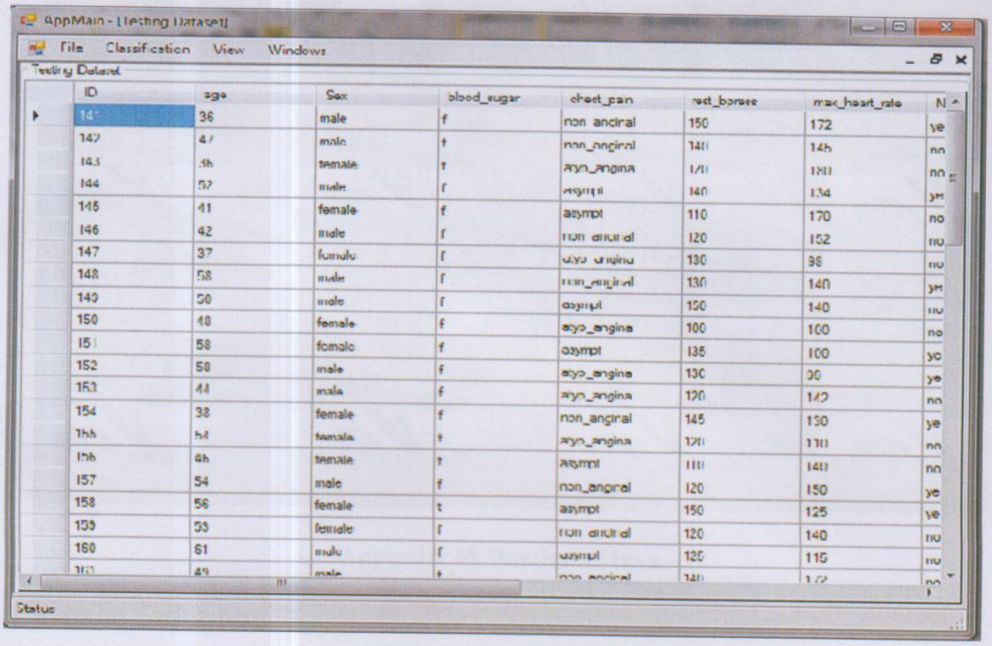
By using this sub menu, user can view generated decision tree and rules

Windows Menu

In this menu there are four sub menus namely ***Cascade, Vertical, Horizontal*** and ***CloseAll*** submenu.

Testing Dataset Submenu

In order to view testing dataset the user can select this submenu.



Appendix L Testing Dataset View

Decision Rules Submenu

By using this sub menu, user can view generated decision tree and rules

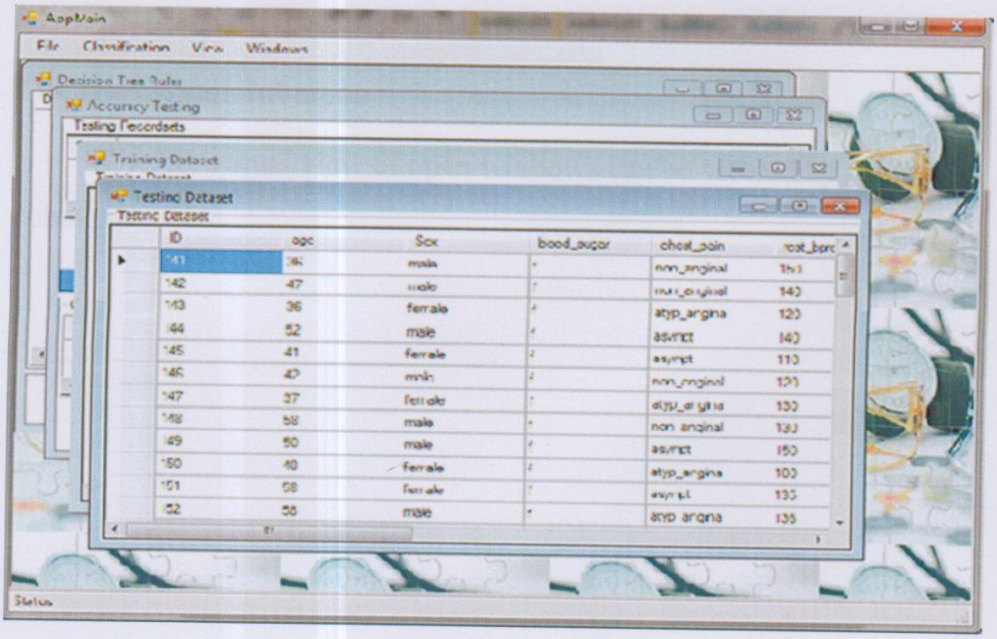
Windows Menu

In this menu there are four sub menus namely ***Cascade, Vertical, Horizontal*** and ***CloseAll*** submenu.

Appendix M Windows Menu

Cascade Submenu

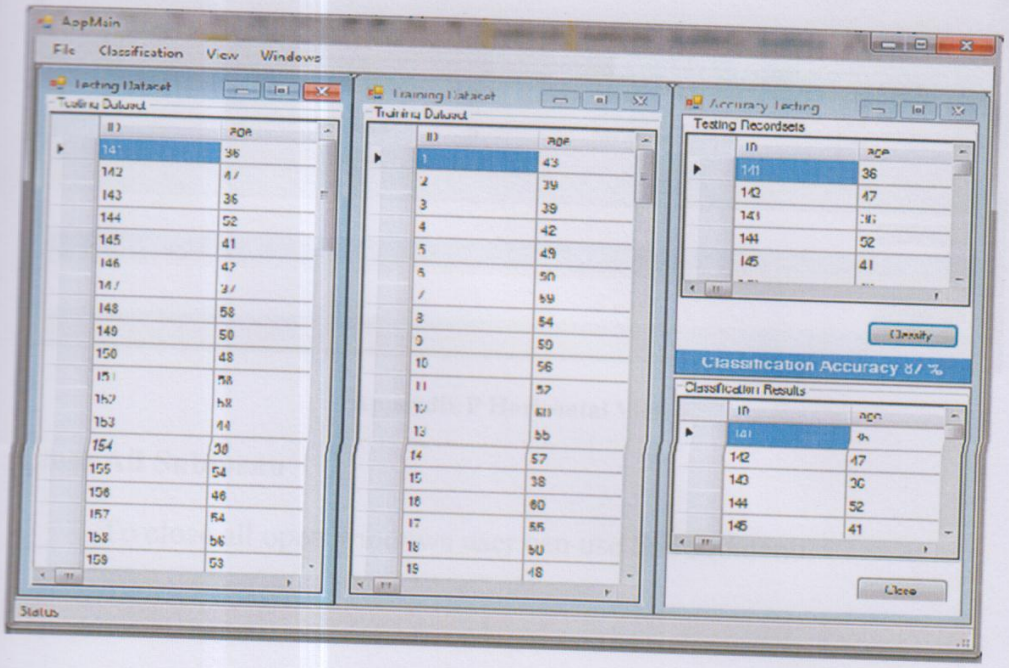
By using this submenu user can view the cascade view of all open windows.



Appendix N Cascade View

Tile Vertical Submenu

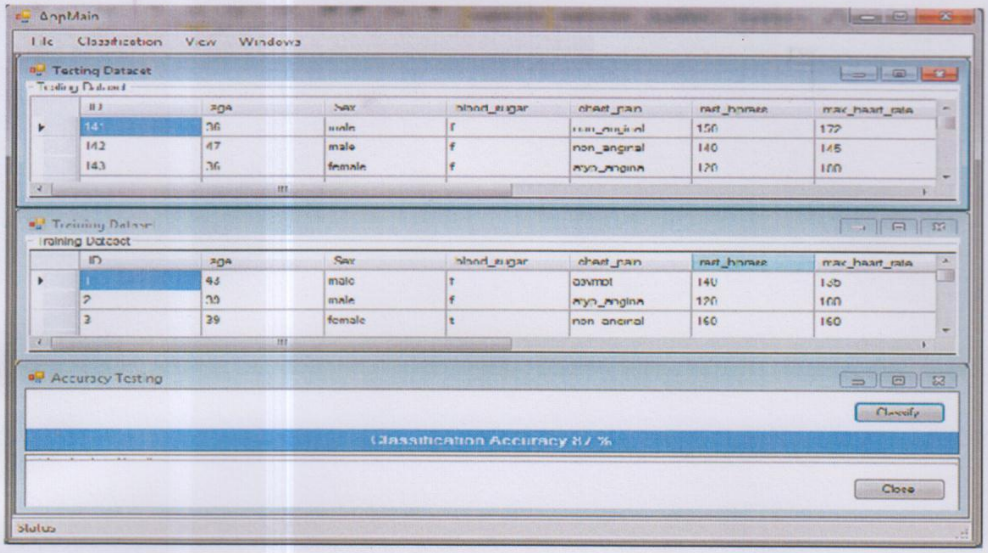
To view vertical view of all open windows, user can select this submenu.



Appendix O Vertical View

Tile Horizontal Submenu

The horizontal view of open windows can be viewed by using this submenu.



Appendix P Horizontal View

Close All Submenu

To close all open windows user can use this submenu.