Al-Tahaddi University

Faculty of Science
Department of Computer science
Sirite - G. S. P. L. A. J.

New Implementation of Unsupervised ID3 Algorithm (NIU-ID3)

Using Visual Basic.net

By:
Ahmed Ali Mohammed Alhouni

Supervisor:
Dr. Faraj A. El-Mouadib

A thesis submitted to the department of computer science in partial fulfillment of the requirements for the degree of Master of Science

Academic year 2008-2009

إن الدراسة ليست غاية في حد ذاتها
وإنما الغاية هي خلق الإنسان النموذجي الجديد

الجماهيرية العربية الليبية الشعبية الاشتراكية العظمى
اللجنة الشعبية العامة للتعليم العالي
جامعة التحدي

الرقم الإشاري: ك ع/ ط2/ 1450/ 11/ 2009م

التاريخ: ....
الموافق: 2009 - 7 - 29

Faculty of Science

Department Of Computer Science

Title of Thesis

New Implementation of Unsupervised ID3 Algorithm (NIU-ID3) Using Visual Basic .net

By

Ahmed Ali Mohammed Alhouni

Approved by:

Dr.Faraj A. El-Mouadib ....
(Supervisor)

Dr. Ahmed M. Abushaala ....
(External examiner)

Dr. Abdalraheem Nasr Alsagheer ....
(Internal examiner)

Countersigned by:
Dr. Ahmed Farag Mhgoub
(Dean of faculty of science)

29
07
09

Dedication

To my parents, Family, friends

and G. S. P. L. A. J.

Acknowledgment

I am very grateful to my supervisor, Dr. **Faraj A. El-Mouadib**, *for his supervision in this MS study and for his support, advice, generous, help and assistance throughout my work. His steadiness and patient support were essential to completion of this work. Without his support, this thesis would have not been accomplished.*

I would like to take this opportunity to thanks Al-tahadi University for my available to study master thesis in site my country.

Finally, I must thank my family for supporting and understanding me throughout the duration of my MS study.

Ahmed A. Alhouni

Abstract

The data volumes have increased noticeably in the few passed years, for this reason some researchers think that the volume of data will be duplicated every year. So data mining seems to be the most promising solution for the dilemma of dealing with too much data and very little knowledge.

Database technology has dramatically evolved since 1970s and DM became the area of attraction as it promises to turn those raw data into meaningful Knowledge which businesses can use to increase their profitability.

Data mining is set of computer assisted techniques designed to automatically mine large volumes of integrated data for new, hidden or unexpected information, or patterns. The progress of data-collection technology, such as bar-code scanners in commercial domains and sensors in scientific and industrial domains, generates huge amounts of data. This explosive growth in data and database generates the need for new techniques and tools that can intelligently and automatically transform the data into useful information and knowledge.

By using pattern recognition technologies, statistical methods and mathematical techniques to sift through warehoused information, data-mining tools helps analysts to recognize the significant facts, relationships, trend, patterns, exceptions and anomalies.

Data warehousing and data mining are collection of tools for managing, analyzing large datasets and discovering novel patterns. Data mining has been widely used by statisticians, data analysts, management information systems community, and other professionals.

Data mining can be used in so many different types of databases (relational database, transactional database, object-oriented database and data warehouse) or other kinds of information repositories (spatial database, time-series database, text or multimedia database, legacy database and the World Wide Web).

Generally, data mining is process of analyzing data from different perspectives and summarizing it into useful information. It starts with raw data and gets results that may be insights, rules, or predictive models.

The data mining systems can be classified based on specific set of criteria as follows:

- Classification according to kinds of databases mined.
- Classification according to kinds of knowledge mined.
- Classification according to kinds of techniques utilized.

- Classification according to applications adapted.

This classification can also be helpful to potential users to distinguish data mining systems and identify those that are best match their specific needs.

The purpose of this research is to implement one of the data mining techniques (classification) to deal with labeled data sets and merging it with another data mining technique (clustering) to deal with unlabeled data sets in a computer system using VB.net 2005.

Our system (NIU-ID3), can deal with two types of data files namely; text data files and access database files. It can also preprocess unlabeled data (clustering of data objects) and process label data (classification).

The NIU-ID3 can discover knowledge in two different forms, namely; decision trees and decision rules (classification rules), this approach is implemented in Visual Basic.net language with SQL. The system is tested with access database, text data (labeled datasets and unlabeled datasets) and presents the results in the form of decision trees, decision rules or simplified rules.

February, 2009

Table of contents

List of figures

List of tables

Chapter 1

Introduction to Data mining

1.1 Introduction

In recent years, there has been a rapid increase in the use of computer technology in decision support systems. The wide use of computers in so many fields such as; supermarket transactions, phone call records, has made it possible to collection of huge amount of data. "This has resulted in the capture and availability of data in immense volume and proportion." [1]

The arrival of data from many sources is too fast through what is called data streams. In fact it is faster than the traditional techniques to process such data and make a good use of it. In today's world of technology many data streams can be cited such as; financial data in banking and securities, point of sale data in retail, medical history data in health care, policy and claim data in insurance, are some instances of the types of data that is being collected. The growing rate of data collection has exceeded the human ability to assimilate and use such data efficiently. "Data must be categorized in some manner if it is to be accessed, re-used, organized, or synthesized to build a picture of the company's competitive environment or solve a specific business problem."[2]

Presently, data repositories are increasing rapidly and include massive amounts of data from commercial, technical, and other domains. According to [3], the amount of collected data in the whole world is doubling every twenty months or so. The capabilities to analyze, summarize, and extract knowledge from the massive amount of data lack the ability to deal with flood of such stream of data.

Dealing with the huge amount of data produced by businesses has brought the concept of information architecture which started new project such as Data Warehousing (DW). The purpose of DW is to provide the users and analysts with an integrated view of all the data for a given enterprise. Data Mining (DM) and Knowledge Discovery in Databases (KDD) is one of the fast growing computer science fields. The popularity and importance is caused by an increased demand for analysis tools that help analysts and users to use, understand and benefit from these huge amounts of data. One theme of knowledge discovery is to gain general ideas from specific ones, which is the basic idea of learning.

Machine learning is subfield of artificial intelligence field that deals with programs that learn from experience.

1.2 Concept learning

Concept learning can be defined as the general description of a category giving some positive and negative examples. So, it is an automatic inference of a general definition of some concept, given examples labeled as members or non-members of the concept. "The aim of concept learning is to induce a general description of a concept from a set of specific examples." [4]. One of human cognition capabilities is the skills to learn concepts from examples. Human have remarkable ability for concept learning with the help of only a small number of positive examples of a concept. As the concept learning problems became more complex has necessitate the existence of more expressive representative language. According to [4], most of the concept learning systems uses attribute-value language where the data (examples) and output (concepts) are represented as conjunctions of attribute-value pairs. "Concept Learning is inferring a Boolean-valued function from training examples of its input and output."[5]. So, this simplicity of representation allowed efficient learning systems to be implemented. On the other hand this simplicity had represented the difficulty of inducing descriptions involving complex relations. "Inductive Learning, a kind of learning methods, has been applied extensively in machine learning."[6]. Current machine learning paradigms are divided to two groups, learning with teacher which is called supervised learning, and learning without teacher which is called unsupervised learning.

1.2.1 Supervised learning

Supervised learning is learning process that supplied with a set of example. The set of examples consists of the input data along with the correct output (class) for each example. "The supervised learning paradigm employs a teacher in the machine learning process." [7]. Some examples of the well known supervised learning models include back propagation in neural networks, K-nearest neighbor, minimum entropy, and decision trees.

1.2.2 Unsupervised learning

In unsupervised learning there is no teacher nor is the data pre-classified. So, the algorithm is never giving training set and is basically left on its own to classify its inputs.

"The unsupervised learning paradigm has no external teacher to oversee the training process, and the system forms (natural grouping) of the input patterns." [7]. One of the most well-known unsupervised methods is clustering. In unsupervised learning, the final achieved result reflects the input data in a more objectively manner and the disadvantage of such learning process is that the achieved classes are not necessarily have subjective meaning.

1.3 Data mining

As recently as two decades ago, data mining was new concept for many people. Creating data mining solution is practical one to many applications. The solution may combine database management, data warehouses, text mining tools to structure data, commercial off-the-shelf data mining software tools, data visualization technology, and advanced data analysis all filtered through lens of business and domain expertise. Data Mining (DM) is the extraction of hidden predictive knowledge from large databases which is a powerful new technology. Data mining is process that uses variety of data analysis tools to discover patterns, relationships and regularities in the data that may be used to make valid predictions. "DM and KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data." [8, 9 and 10]

Most DM algorithms have been drawn from areas of statistics and machine learning adapted to induce knowledge from data contained within databases.

According to [11], Data mining is the process of discovering meaningful new correlations, patterns and trends by sifting through large amounts of data stored in repositories, using pattern recognition technologies as well as statistical and mathematical techniques. Data Mining is considered to be a revolution in information processing and there are many definitions in the literature to what constitute data mining.

According to [9], the attraction of the wide use of data mining is due to:

- The availability of very large databases.
- The massive use of new techniques coming from other disciplines of computer science community like neural networks, decision trees, induction rules.
- Commercial interests in order to propose individual solutions to targeted clients.

- New software packages, more user-friendly, with attractive interfaces, directed as much towards decision makers as professionals analysts, but much more expensive.

The main objective of DM is to use the discovered knowledge for purposes of explaining current behavior, predicting future outcomes, or providing support for business decision. Data mining enables corporations and government agencies to analyze massive volumes of data quickly and relatively inexpensively. Today, mining can be performed on many types of data, including those in structured, textual, spatial, Web, or multimedia forms. "Data mining is the process of discovering advantageous patterns in data." [12].

1.3.1 Data mining process

The main steps of data mining process are:

- Analyzing the problem involves the understanding of the application domain, relevant prior knowledge, and what results the end-user is seeking.
- Preparing the data involves the creation of the target data set and data preprocessing (i.e. data cleaning, data reduction, data projection, etc...).
- Choosing the data mining task aims at on the type of DM functionality that is needed to carry out. Possible tasks are classification, regression, clustering, or others.
- Choosing the data mining algorithms that are appropriate for the mining task to search for patterns and regularities.
- Generating pattern is accomplished by using rule induction and selected algorithm. The pattern could be in any of several representational forms. Possible representations are; classification rules, classification trees, regression, clustering or others.
- Interpreting pattern involves the validation and interpretation of the mind patterns and regularities.
- Consolidating knowledge is to report on resulted patterns and incorporate it into real-world performance system, or simply reported to interested parties.
- Monitoring pattern is to assure that data mining strategy is correctly accomplished.

1.3.2 Data mining techniques

According to [13], most data-mining methods (techniques) are based on very-well tested techniques from machine learning, pattern recognition, and statistics: classification, clustering, regression, etc... Data mining techniques are applicable to wide variety of problem areas. Some of these techniques are:

- Classification is a supervised technique that maps (classifies) data object into one of several predefined classes, i.e. given an object and its input attributes, classification output is one of possible mutually exclusive classes. The aim of classification task is to discover some kind of relationship between inputs attributes and output class, so that discovered knowledge can be used to predict the class of new unknown object.
- Regression is considered to be a supervised learning technique to build more or less transparent model, where the output is a continuous numerical value or vector of such values rather than discrete class.
- Clustering is unsupervised learning technique, which aims at finding clusters of similar objects sharing number of interesting properties.
- Dependency modeling consists of discovering model which describes significant dependencies among attributes.
- Change and deviation detection is the task of focusing on discovering most significant changes or deviations in data between actual content of data and its expected content (previously measured) or normative values.
- Summarization aims at producing compact and characteristic descriptions for a given set of data. It can take of multiple forms such as; numerical (simple descriptive statistical measures like means, standard deviations...), graphical forms (histograms, scatter plots...), or on the form of "if-then" rules.

1.3.3 Goals and tools of data mining

The goal of data mining is to uncover relationships in data that may provide useful insights. The goals of data mining are prediction and description. Prediction uses supervised learning technique to predict values of data using known values found from different data. Description focuses on employing unsupervised learning technique to find human interpretable patterns describing data.

Tools for data mining include mixture of well know mathematical algorithms and techniques applied to general business problems made possible by increased availability of data and inexpensive storage and processing power. The data mining tools can play essential role, because they can quickly generate scenarios regarding effects of growth and development based on available data. Data mining tools provide both developers and business users with interface for discovering, manipulating, and analyzing corporate data. Data mining tools can sweep through databases and identify previously hidden patterns in one step. Data mining tools can also automate the process of finding predictive information in large databases. Data mining tools produce better results with larger, broader databases. When data mining tools are implemented on high performance parallel processing systems, they can analyze massive databases in minutes.

1.3.4 Applications of data mining

The data mining has numerous applications. The most widely known applications are those that require some sort of prediction. We can mine our data wherever they reside in data warehouse, data mart, database, legacy system, or even in external information such as survey and purchased data sets.

Some uses of data mining include:

- Network optimization and intrusion detection.
- Call center analysis.
- Human genome analysis.
- Insurance claims processing analysis.
- Accident trending.
- E-Business analysis.
- Crime pattern detection.
- Disease outbreak detection.
- Fraud detection.
- Direct and Interactive marketing.
- Market basket.
- Web traffic analysis.

1.4 Thesis objectives

Due to the fact that the ID3 is a classification algorithm that works in supervised fashion. This work is mainly concerned with two aspects which are:

1. The implementation of the original ID3 algorithm using Visual basic.net programming language.
2. Adding a front-end to the ID3 algorithm so it will work in unsupervised fashion.

This work will commence with a theoretical aspects and grounds of the ID3 algorithm and then an implementation of the algorithm in Visual basic.net programming language will be carried out. Then a front-end module will be implemented using one of the methods from cluster analysis to label the data to be used by the new implementation of the ID3. The newly created version of the ID3 algorithm will be tested by some of the well known databases (i.e. Iris database...).

1.5 Thesis outline

The thesis is organized into six chapters and they are as follows: **Chapter 1** introduced the concept learning and data mining was given. Also the chapter introduced the data mining process, techniques, goals, tools and applications.

Chapter 2, the ideas about classification will be reviewed and the development of the Decision Tree algorithm will be given. The theoretical and practical aspects of the algorithm ID3 will also be presented. The features of ID3 will be explained in more details. **Chapter 3**, the dealing with different types of attributes will be given and our implementation of ID3 in Visual basic.net given. **Chapter 4**, we will be concerned with the implementation of the front-end module by one of the clustering methods to label the unlabeled data and make it ready to be used by the ID3 algorithm. **Chapter 5**, the newel created version of the ID3 algorithm will be tested by a number of well Known databases. **Chapter 6**, some comments and discussions on the obtained results will be given and the study will be concluded by given some future work suggestions.

Chapter 2

Classification and decision tree

2.1 Introduction

The wide availability of huge amounts of data and the imminent need for turning such data into useful information has generated an urgent need for new techniques and automated tools that can intelligently assist us in transforming the vast amounts of data into useful knowledge. One of the techniques used in turning the data into knowledge is to use decision trees.

Decision tree is a classification scheme that can be used to produce classification rules. In this chapter, we will review some basic ideas of classification and the development of decision trees algorithm. The theoretical and practical aspects of the ID3 algorithm will also be presented and the features of ID3 will be explained in details.

2.2 Basic ideas of classification

With enormous amounts of data stored in databases and data warehouses, it is increasingly important to develop powerful tools for data analysis to turn such data into useful knowledge that can be used decision-making.

One of the most well studied data mining functionalities is classification due to its wide used in many domains. "Classification is an important data mining problem. Given a training database of records, each tagged with a class label." [14].

The task of classification is first step to build a model (classifier) from the given data (pre-classified data objects) and second step is to use the model to predict or classify unknown data objects.

The aim of a classification problem is to classify transactions into one of a discrete set of possible categories. The input is a structured database comprised of attribute-value pairs. Each row of the database is a transaction and each column is an attribute taking on different values. One of the attributes in the database is designated as the class attribute; the set of possible values for this attribute being the classes.

Classification is a data mining technique that typically involves three phases, learning phase, testing phase and application phase. The learning model or classifier is built during learning phase. It may be in form of classification rules, decision tree, or mathematical formula. Since, class label of each training sample is provided, this approach is known as supervised learning. In testing phase test data are used to assess the accuracy of classifier. If classifier passes the test phase, it is used for classification of new, unclassified data tuples. The application phase, the classifier predicts class label for these new data objects. According to [6], classification has been applied in many fields, such as medical diagnosis, credit approval, customer segmentation and fraud detection.

There are several techniques (methods) of classification:

- Classification by decision tree induction such as: ID3 (Iterative Dichtomizer 3rd), C4.5, SLIQ, SPRINT and rainforest algorithms.
- Bayesian classification by the use of Bayes theorem.
- Classification by back propagation in the area of NN.
- Classification based on the concepts from association rule mining.
- Other classification methods: KNN classifiers, case based reasoning, genetic algorithms, rough set approach, fuzzy set approaches.

2.3 Decision tree algorithm

A decision tree is a flow chart like tree structure. The top most node in the tree is the root node. Each node in the tree specifies a test on some attribute and each branch descending from the node corresponds to one of the possible values of the attribute except for the terminal nodes that represent the class. An instance is classified by starting at the root node of the tree, testing the attribute specified by the given node, then moving down the tree branch corresponding to the value of the attribute in the given example. This process is repeated for the sub tree rooted at the current node. In order to classify an unknown sample, the attribute values of the sample are tested against the decision tree. A path is traced from the root to a leaf node that holds the class for that instance.

According to [15], Top-Down Induction of Decision Tree (*TDIDT*) is general purpose systems which classify sets of examples based on their attribute values pairs. The *TDIDT* algorithm can be rerun to include new example of the data sets. While this is useful feature, it is also time consuming. One of earliest *TDIDT* algorithms is the Concept Learning System (***CLS***) by Hunt in 1966. The algorithm works by presenting system with training data from which top-down decision tree is developed based on frequency of information.

In 1986, Quinlan had modified ***CLS*** algorithm by enhancing it by the addition of the concept of windowing and information-based measure called *entropy*. The entropy is used to select the best attribute to split the data into two subsets, so every time the produced decision tree will be the same. The concept of windowing is used to ensure that all the cases in the data are correctly classified.

According to [16], there are several reasons that make decision tree very attracting learning tool. Such as:

- Decision tree learning is a mature technology. It has been in existence for 20+ years, has been applied to various real world problems, and the learning algorithm has been improved by several significant modifications.
- The basic algorithm and its underlying principles are easily understood.
- It is easy to apply decision tree learning to a wide range of problems.
- Several good, easy to use decision tree learning packages are available.
- It is easy to convert the induced decision tree to a set of rules, which are much easier for human to evaluate and manipulate, and to be incorporated into an existing rule based systems than other representations.

2.4 Theoretical and practical aspects of the ID3 algorithm

According to [15], the ***ID3*** algorithm is a decision tree building algorithm which determines classification of objects by testing values of their properties. It builds tree in top down fashion, starting from set of objects and specification of properties. At each node of tree, the properties are tested and the result is used to partition data object set. This process is recursively carried out till each subset of the decision tree is homogeneous. In other words it contains objects belonging to same category. This then becomes leaf node. At each node of the tree, the tested property is chosen on bases of information theoretic criteria that seek to maximize the information gain and the minimize entropy. In simpler terms, the chosen property is the one that divides the set of objects in the most possible homogeneous subsets. The ID3 algorithm has been successfully applied to wide variety of machine learning problems. It is well known algorithm, however such approach has some limitations.

In ID3, windowing is to select a random subset of the training set to be used to build the initial tree. The remaining input cases are then classified using the tree. If the tree gives correct classification for these input cases then it is accepted for training set and the process ends. If this is not the case then the misclassified cases are appended to the window and the process continues until the tree gives the correct classification.

The information theoretic heuristic is used to produce shallower trees by deciding an order in which to select attributes. The first stage in applying the information theoretic heuristic is to calculate the proportions of positive and negative training cases that are currently available at a node. In the case of the root node this is all the cases in the training set. A value known as the information needed for the node is calculated using the following formula where p is the proportion of positive cases and q is the proportion of negative cases at the node:

$$-p\log_2 p - q\log_2 q \text{ ...... } (3.1)$$

2.4.1 The basic algorithm of ID3

According to [17, 18, 19, and 20], given a set of examples ***S***, each of which is descried by number of attributes along with the class attribute ***C***, the basic pseudo code for the ID3 algorithm is:

- If (all examples in ***S*** belong to class ***C***) then make leaf labeled ***C***

  Else select the "most informative" attribute ***A***

  Partition ***S*** according to ***A***'s values ($v_1, \ldots, v_n$)
- Recursively construct sub-trees $T_1, T_2, \ldots, T_n$ for each subset of ***S***.

ID3 uses a statistical property, called information gain measure, to select among the candidates attributes at each step while growing the tree. To define the concept of information gain measure, it uses a measure commonly used in information theory, called entropy. The entropy is calculated by:

$$Entropy(S) = \sum_{i=1}^{c} - p_i \log_2 p_i \text{ ...... } (3.2)$$

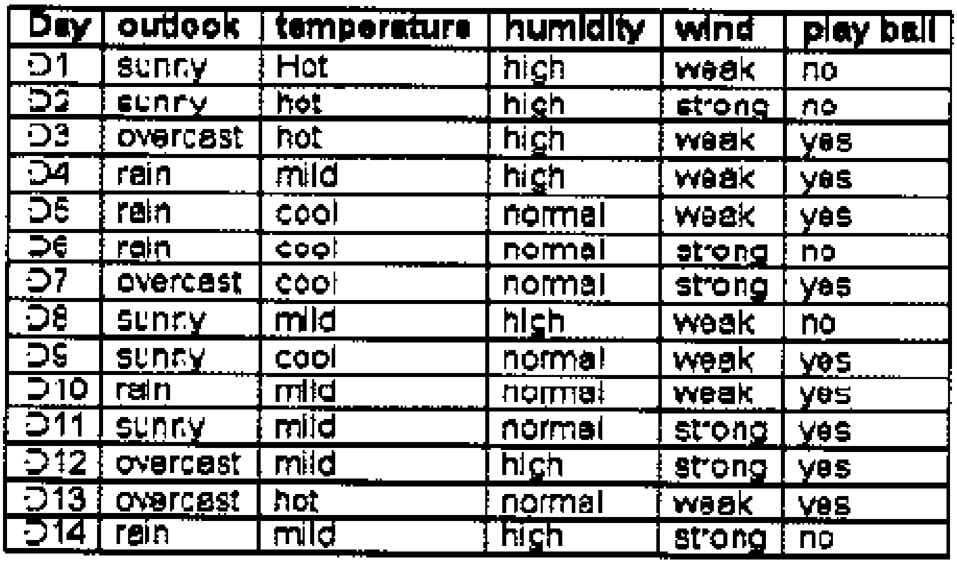
Where ***S*** is a set, consisting of ***s*** data samples, ***Pi*** is the portion of ***S*** belonging to the class *i*. Notice that the entropy is **0** when all members of S belong to the same class and the entropy is 1 when the collection contains an equal number of positive and negative examples. If the collection contains unequal numbers of positive and negative examples, the entropy is between **0** and **1**. In all calculations involving entropy, the outcome of (**0 log$_2$ 0**) is defined to be **0**. With the Information gain measure, given entropy as a measure of the impurity in a collection of training examples, a measure of effectiveness of an attribute in classifying the training data can be defined. This measure is called information gain and is the expected reduction in entropy caused by partitioning the examples according to this attribute. More precisely, the information gain is *Gain*(***S***, ***A***) of an attribute ***A***, relative to a collection of examples ***S***, is defined as:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \ldots\ldots \quad (3.3)$$

Where values of ***A*** is the set of all possible values for attribute ***A***, and ***S$_v$*** is the subset of ***S*** for which attribute ***A*** has value ***v***. The first term in equation (3.3) is the entropy of the original collection ***S***, and the second term is the expected value of the entropy after ***S*** is partitioned, using attribute ***A***.

Gain(***S***, ***A***) is the expected reduction in entropy caused by knowing the value of attribute ***A***. Therefore the attribute having the highest information gain is to be preferred in favor of the others. Information gain is precisely the measure used by ID3 to select the best attribute at each step in growing the decision tree.

As an example [8, 15, 19, 21, 22, 23, 24 and 25], consider decision model to determine whether the weather is amenable to play baseball. Historic data of observations over period of two weeks is available to build a model as depicted in table-2.1.



Day	outlook	temperature	humidity	wind	play ball
D1	sunny	Hot	high	weak	no
D2	sunny	hot	high	strong	no
D3	overcast	hot	high	weak	yes
D4	rain	mild	high	weak	yes
D5	rain	cool	normal	weak	yes
D6	rain	cool	normal	strong	no
D7	overcast	cool	normal	strong	yes
D8	sunny	mild	high	weak	no
D9	sunny	cool	normal	weak	yes
D10	rain	mild	normal	weak	yes
D11	sunny	mild	normal	strong	yes
D12	overcast	mild	high	strong	yes
D13	overcast	hot	normal	weak	yes
D14	rain	mild	high	strong	no

Table-2.1: Sample data to build a decision tree using ID3 algorithm.

The weather data attributes are: outlook, temperature, humidity, and wind speed. The target class is the classification of the given day as being suitable (*yes*) or not suitable (*no*). The domains of each of the attributes are:

outlook = (sunny, overcast, rain).

temperature = (hot, mild, cool).

humidity = (high, normal).

wind = (weak, strong).

To determine attribute that would be root node for the decision tree; the gain is calculated for all four attributes. First, we must calculate the entropy for all examples, with ***S*** by equation (3.2) as follows:

*Entropy(S) = - (9/14) * Log2 (9/14) - (5/14) * Log2 (5/14) = 0.940*

After that we can calculate the information gain for all four attributes by the use of equation (3.3) as follows:

Entropy (weak) = - (6/8) * log2 (6/8) - (2/8) * log2 (2/8) = 0.811

Entropy (strong) = - (3/6) * log2 (3/6) - (3/6) * log2 (3/6) = 1.00

Gain(*S*, wind) = *Entropy*(*S*) - (8/14) * *Entropy* (weak) - (6/14) * *Entropy* (strong)

= 0.940 - (8/14) * 0.811 - (6/14) * 1.00= 0.048

Similarly the gain is calculated for the other attributes,

Gain(*S*, outlook) = 0.246

Gain(*S*, temperature) = 0.029

Gain(*S*, humidity) = 0.151

Because the outlook attribute has the highest gain, therefore it is used as the decision tree root node. The outlook attribute has three possible values; the root node has three branches labeled with sunny, overcast and rain.

The next step is to develop the sub tree, one level at time, starting from the left (under sunny) using the remaining attributes namely humidity, temperature and wind.

The calculation of gain is carried out for each of the attributes given the value of the previous value of the attribute. The final decision tree obtained as the result of ID3 algorithm is depicted in figure-2.1:

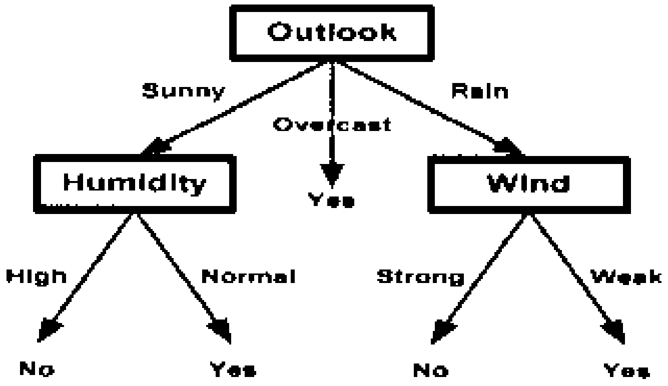


Figure-2.1: Decision tree to determine whether the weather is amenable to play baseball.

The following rules are generated from the above decision tree:

IF outlook= overcast THEN play ball= yes

IF outlook= rain $\wedge$ wind= strong THEN play ball= yes

IF outlook= rain $\wedge$ wind= weak THEN play ball= yes

IF outlook= sunny $\wedge$ humidity= high THEN play ball= no

IF outlook= rain $\wedge$ humidity= high THEN play ball= no

2.4.2 Features of ID3

The most important feature of ID3 algorithm is its capability to break down a complex decision tree into a collection of simpler decision trees. Thus it provides a solution which is often easier to interpret. In addition, some of other important features are:

- Each attribute can provide at most one condition on a given path. This also contributes to comprehensibility of the resulted knowledge.
- Complete hypothesis space: any finite discrete valued function can be expressed.
- Incomplete search: searches incompletely through the hypothesis space until the tree is consistent with the data.
- Single hypothesis: only one current hypothesis (the best one) is maintained.
- No backtracking: once an attribute is selected, this can't be changed.
- Full training set: attributes are selection by computing information gain on the full training set.

Chapter 3

System design and implementation

3.1 Dealing with different types of attributes

Due to the fact that the ID3 algorithm deal with discrete valued attributes and we have decided to limit the number of values per attribute to four. The reason for that is to have simple decision trees and rules. If the number of different values per attribute is greater than four then the (NIU-ID3) system will reduce it to four by the use of discretization (normalization) techniques. This process is carried out on numerical attributes and for symbolic attribute values the same process will be carried out after coding the attributes values. For a continuous valued attribute *A*, the system partitions the attribute values into four intervals by:

$$\mathit{Length\_of\_interval} = \frac{Max_A - Min_A}{C} \ \ldots\ldots \quad (3.4)$$

Where: *Max_A* is the maximum value of attribute *A*, *Min_A* is the minimum value of attribute *A* and *C* is the number of *intervals* (default value is 4).

For example: if we have a numerical attribute with the following values: 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 58, 69, 70, 71, 72, 73, 74, 76, 77, 79.

We can calculate the interval length as: (79 – 43) / 4 = 9. The intervals and the corresponding values can be seen in the diagram below:

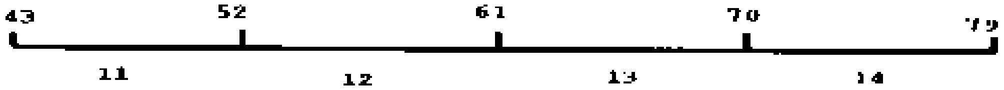


Table-3.1 depicts the original values with their corresponding new values.

Original values	Corresponding values
43.44.45.46.47.48.49.50.51	11
52,53,54,55.56.57.58.59.60	12
61,62.63.64,65.66.67.68.69	13
70.71.72.73.74.76.77.79	14

Table-3.1: The original values with their corresponding new values.

Another example is for a symbolic attribute that has the following values: BB, CC, DD, EE, FF, GG and HH.

Such attribute values is dealt with in the following way:

First, the values will be coded as: 01, 02, 03, 04, 05, 06 and 07.

Second, we calculate the interval width as: (7 - 1) / 4 = 1.5.

The intervals and the corresponding values can be seen in the diagram below:

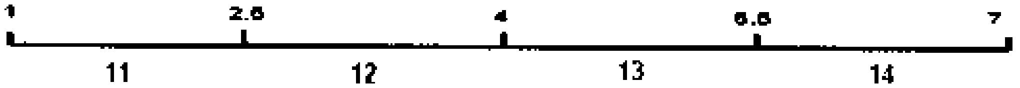


Table-3.2 depicts the original values with their corresponding codes and new values.

Original values	Corresponding values	Corresponding discretized values
BB, CC	01, 02	11
DD	03	12
EE, FF	04, 05	13
GG, HH	06, 07	14

Table-3.2: The original values with their corresponding codes and new discretized values.

Each code for the corresponding values consists of two digits the first represents the attribute number and the second represents the interval number (attribute's value). For example, the code 13 is for the third value of the first attribute.

3.2 System architecture

In the previous subsections we gave a brief review of classification, decision tress and the theoretical and practical aspects of the ID3 algorithm. The ID3 algorithm deals with the classification problem strictly in a supervised sense.

Here we will demonstrate the design and implementation of our system, which is a new implementation of the ID3 algorithm to make it work in unsupervised manner by building a font-end to it. Our system is called New Implementation of Unsupervised ID3 (***NIU-ID3***). We will also give an overview of the over all architecture of the NIU-ID3, the tasks that the system can deal with, the data format that it uses and the results that is produced from it.

The NIU-ID3 system accomplishes its task via several stages that are executed in a serial fashion in the form of a wizard form (Data Mining Wizard). These stages are grouped into four main components:

- Data set is the data to be used to discover knowledge from.
- Preprocessing is the process of preparing the data for classification if the data is continuous or unlabeled.
- Classification is the classification process (ID3 algorithm) to produce the knowledge.

- Knowledge is the results discovered by classification process.

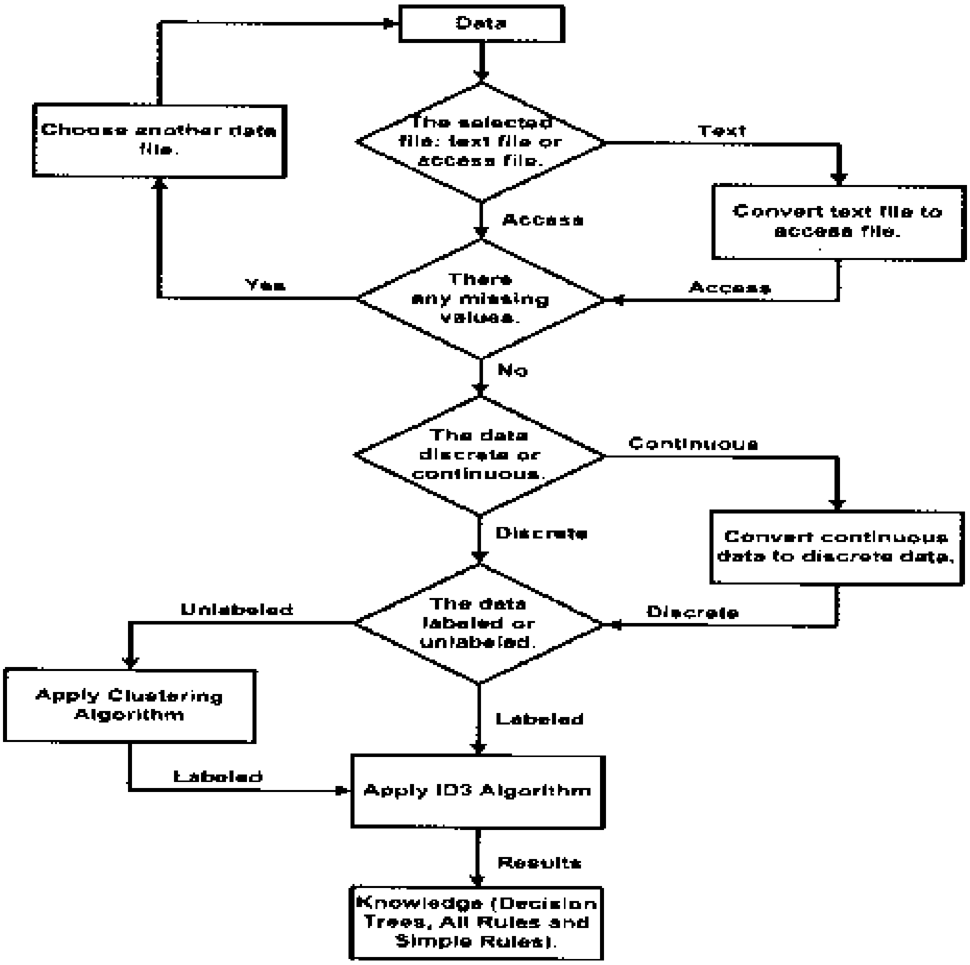

Data
Choose another data file.
The selected file: text file or access file.
Text
Convert text file to access file.
Access
Yes
There any missing values.
Access
No
The data discrete or continuous.
Continuous
Convert continuous data to discrete data.
Discrete
Unlabeled
The data labeled or unlabeled.
Discrete
Apply Clustering Algorithm
Labeled
Labeled
Apply ID3 Algorithm
Results
Knowledge (Decision Trees, All Rules and Simple Rules).

Figure-3.1: General architecture of NIU-ID3 system.

The NIU-ID3 system can manipulate with two types of data files; that is a text data or Access databases. It can also preprocess unlabeled data (clustering of data objects) and process label data (classification). Our system can discover knowledge into two different formats, namely; decision trees and classification rules.

3.3 System modeling

In general there are three important phases in building a computer system. These phases are being; model building phase, testing phase and application phase. At beginning we will talk about the model building phase and the other two phases will be in brief mentioned and deal with them in more detail in the next chapters.

3.3.1 Model building

The model building phase can start by building a simulated model of the problem. This model will provide a clear understanding of the problem in question. From the literature, there are three perspectives used in the development of simulated models and these are:

1. Use some Graphical User Interface (GUI) tools to develop the simulated model on the screen. Use arcs to connect the system components to create the logical model and the run simulated model. "In most cases, due to the limitation of the simulation program under use, some simplifications and approximations to the model may be required. Such simplifications or approximations can be very costly." [26].
2. Support the belief that any simulation program would not be able to model all tasks of systems without the need to make some modification. "This suggests that models should be developed from scratch by using a simulation modeling language." [26]. This approach may increase the time needed to produce the system and may divert the developer to pay more attention to the programming challenges than understanding the system.
3. Focuses on the use of GUI that will automatically generate code with the possibility of the developer intervention to make some changes to the code to match the system requirements. This is a very popular practice because it reduces the need time to produce the system, on the other hand code modification is a tedious task.

Figure-3.2 depicted the processes of system:

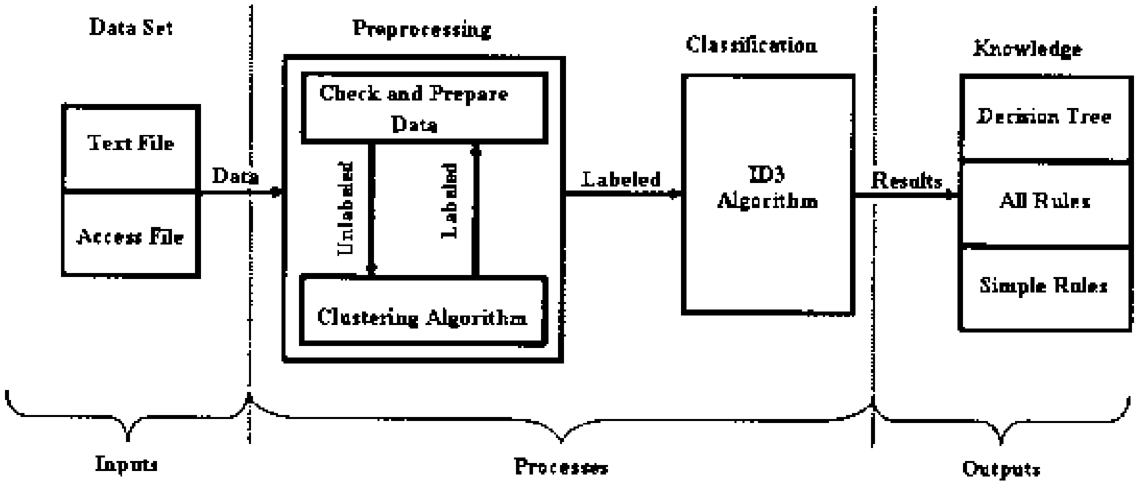


Figure-3.2: Build process of the NIU-ID3 system.

The model building phase consists of a number of steps, as follows:

1. Data loading which consists of two sub steps:
   - In loading data, our system deals with two types of data files; text files data and Access database files. Our system will ask the user to select the file type. The

system initially designed to deal with only Access database files, so if the loaded data is not Access database file then a preprocessing step will be converted it to Access database file. So, the loaded database is called the training set. This data set consists of a set of tuples, each tuple consists of a number of values and an additional attribute called class attribute. At this stage by the use of ADO.NET technique to establish the connection between the database and the system. The ADO.NET is a part of the base class library that is included with the Microsoft .NET framework, and it is a set of computer software components that can be used by programmers to access the data. In reality, ADO.NET can not be connected with data sources directly, but it needs .NET data providers. Here we use OLE DB.NET data provider, and Microsoft® universal data access component 2.5 or 2.6 (MDAC 2.5 or MDAC 2.6).

- The data selection sub step will display all table's names that are available of type Access to the user, so he/she can select one of them and the system starts the discovery process.

  There are some conditions on the data file that is loaded to our system. We will explain them in the next chapter.

2. Preprocessing: consists of four sub steps:
   - Check missing values: if there are some missing values in the loaded data our system will ask the user to change the data file due to the fact that the system is not accommodated to deal with missing values.
   - Converting a text file to Access database file: If the loaded data file is of type text, the system will convert it into an Access database file.
   - Data labeling: if the loaded data is un-labeled then the system will label it via the clustering component of the system.
   - Continuous and discrete valued attributes: The ID3 does not work with continuous valued attributes. If the number of values per attribute is more than four then the system will divide the range of attribute values into intervals using the equation 3.4.
3. Classification is the process of building a model or a function that can be used to predict the class of unclassified data objects. In our system we use the ID3 algorithm for this task.

4. Knowledge is the end result that is produced from our system. The end result can be in one of different form such as; decision tree, decision rules or more general simplified rules.

In our system the end results can be saved in text files if the user desire along with the data.

3.3.2 Model testing

After the model is built, the model must be tested by a sample of data in the form of experiments. Such experiments will deal with different database varying in number of tuples, number of attributes and if the data is labeled or not. Testing of the model will be detailed in chapter five.

3.3.3 Application of the model

After testing the model and the results are satisfactory then the model can be put in use in the real world.

3.4 Physical design

3.4.1 Design considerations

In the system software design stage there is some aspects we must focuses on. These are:

1. Extensibility: When we need to add new capabilities to the original architecture of software should be no changes to its major components.
2. Robust ness: The software should tolerate unpredictable or invalid input.
3. Reliability: The software should perform all of its required tasks for a specified period of time.
4. Fault tolerance: The software should be able to recover from some of its component failure.
5. Security: The software should be able to block any unauthorized access.
6. Maintainability: The software can be restored to a specified condition within a specified period of time. For example, antivirus software may include the ability to periodically receive virus definition updates in order to maintain the software's effectiveness.
7. Compatibility: The software should be able to be incorporated with other products even when some new version has been issued.

8. Modularity: The software should be composed of modules each of which can be implemented and tested in isolation before integrated in the over all system. The modularity leads to an ease in maintenance and development of the system.
9. Reusability: Each module of the system should capture the essence of its functionality and nothing more or less. This single task approach renders the components reusable wherever similar needs arise.

3.4.2 Design methodologies

The design methodology purpose is to give a framework of the system actual design. It aims to simplify the design process and to enforce some standard design principles that improve the quality of the design. According to [27], one of the earlier design methodologies is the DFD that is still considered to be one of the best modeling techniques to represent the processing requirements of a system. DFD is a significant modeling technique that explains the course or movement of information in a process. The flow of the information in a process is based on the input and the output. A DFD can represent technical or business processes to illustrate the data flow from a process to the next and finally to the final results. DFD is common tool for designers to show the interaction between the system and outside entities.

According to [28], there are four symbols (notations) that are used in DFD diagrams, which depicted in figure-3.3.

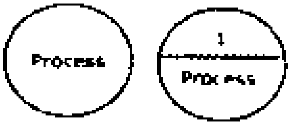
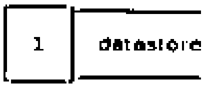
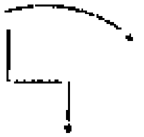
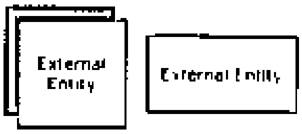
Notation names	Descriptions	Symbols
Process notation	The task of this component is to give and indication of transforming the input to output via the execution of some process on the data.	
Data store notation	Data store component are repositories of data in the system.	
Dataflow notation	Dataflow component represents the pipelines by which packets of information flow from one process to another.	
External entity notation	External entities component represent the means by which the system communicates with the outside world.	

Figure-3.3: DFD notations.

Figure-3.4 depicts the over all processes of our system using DFD notations to demonstrate data flows, data processes and external entities that are needed to implementation of the NIU-ID3 system.

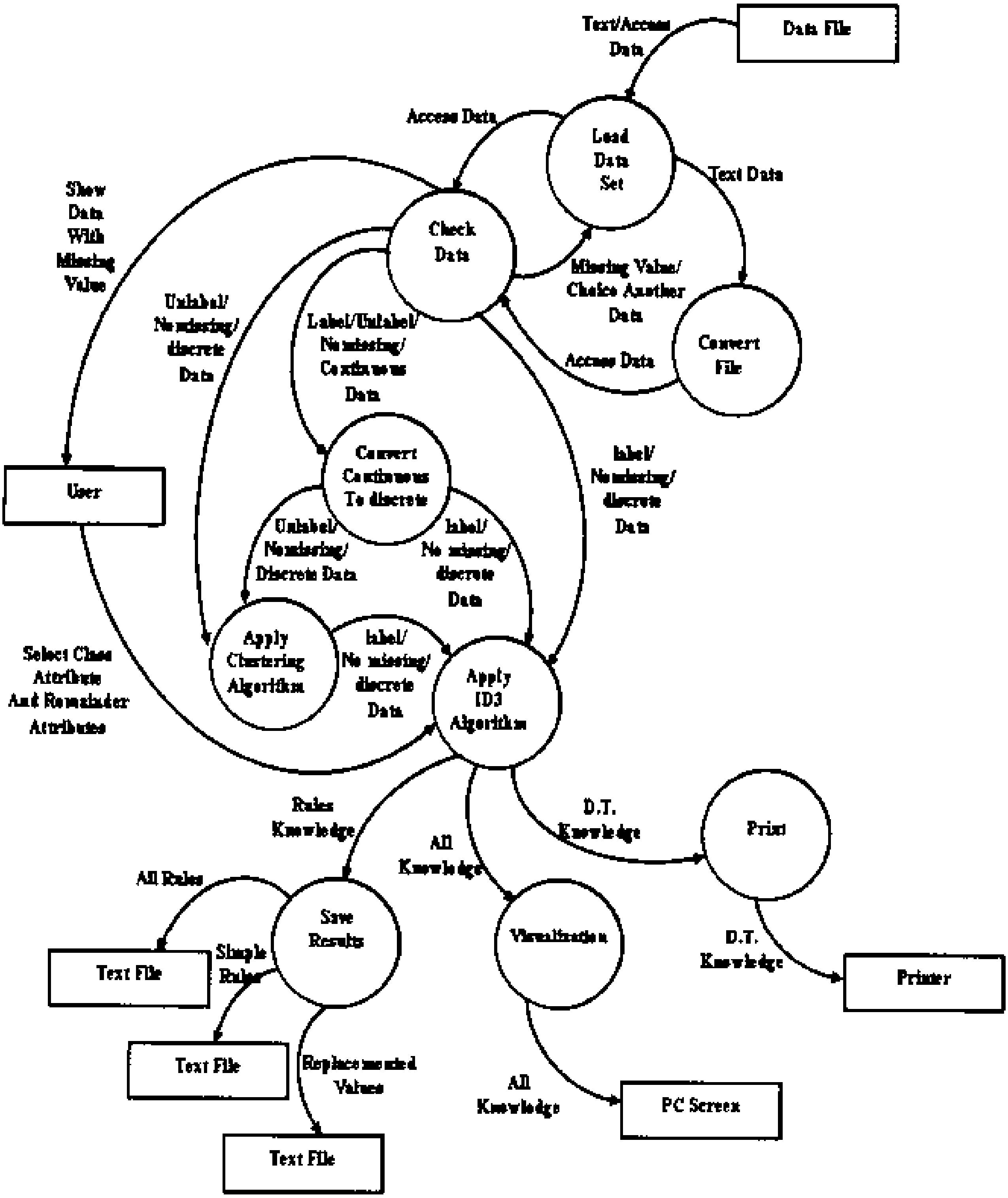

Data File
Text/Access Data
Load Data Set
Access Data
Text Data
Check Data
Missing Value/ Choice Another Data
Convert File
Access Data
Show Data With Missing Value
User
Unlabel/ Nonmissing/ discrete Data
Label/Unlabel/ Nonmissing/ Continuous Data
label/ Nonmissing/ discrete Data
Convert Continuous To discrete
Unlabel/ Nonmissing/ Discrete Data
label/ No missing/ discrete Data
Apply Clustering Algorithm
label/ No missing/ discrete Data
Apply ID3 Algorithm
Select Class Attribute And Remainder Attributes
Rules Knowledge
All Knowledge
D.T. Knowledge
Print
All Rules
Save Results
Simple Rules
Text File
Text File
Replacemented Values
Text File
Visualization
All Knowledge
PC Screen
D.T. Knowledge
Printer

Figure 3.4: DFD for the NIU-ID3 system.

Chapter 4

Clustering Front-End module

4.1 Introduction

The goal of NIU-ID3 system is to build a decision tree and to extract classification rules (decision rules) from the provided data set. Such rules can be used for prediction. The classification module of our system (ID3 algorithm) needs a labeled data set to train the classifier. Such data set consists of a number of records, each of which consists of several attributes. Attributes values will be dealt with accordingly. There is one distinguished attribute called the dependent (class) attribute.

In the case of un-labeled data, the clustering module will be used to carry out the labeling process. In this chapter we will focused or concerned on an one algorithm of clustering techniques, which called fuzzy k-means algorithm (extension of the normal k-means clustering method), and its software package, which called "FuzME program" to make it as a Front-End module to our system.

4.2 Clustering methods

In general, clustering is the process of grouping data objects into groups or clusters such that:

- Each group or cluster is homogeneous or compact with respect to certain characteristics. That is, objects in each group are similar to each other.
- Each group should be different from other groups with respect to the same characteristics; that is, objects of one group should be different from the objects of other groups.

Clustering is an unsupervised learning technique used to divide data sets into groups or clusters. These clusters can be viewed as a group of elements which are more similar to each other than elements belonging to other groups. An alternative definition of a cluster is a region with a relatively high density of points, separated from other clusters by a region with a relatively low density of points. "Clustering is a useful technique for the discovery of some knowledge from a dataset. It maps a data item into one of several clusters, where clusters are natural groupings of data items based on similarity metrics or probability density models. Clustering pertains to unsupervised learning, when data with class labels are not available." [29].

In general, data clustering algorithms can be categorized as hierarchical or partitioning. Hierarchical algorithms find successive clusters using previously established clusters, whereas partitioning algorithms finds all clusters at once. According to [30], some clustering algorithms can be categorized as following:

1. Partitioning methods:
    - Relocation Algorithms.
    - Probabilistic Clustering.
    - K-medoids Methods.
    - K-means Methods.
    - Density-Based Algorithms:
        - Density-Based Connectivity Clustering
        - Density Functions Clustering
2. Hierarchical Methods
    - Agglomerative Algorithms.
    - Divisive Algorithms.

4.2.1 Partitioning methods

This type of clustering algorithms requires the specification of the number of clusters. The division of the data objects is done according to the similarity of the data objects to the cluster centers (representative object). The division must be to mutually exclusive subsets (clusters) such that each data object is exactly in one cluster and each cluster must contain at least one object. "Partition the data set into ***k*** clusters. An immediate problem is how to determine the "best" value for ***k***. This is done either by guesswork, by application requirements, or by trying running the clustering algorithm several times with different values for ***k***, and then selecting one of the solutions based on a suitable evaluation criterion." [31].

4.2.2 Hierarchical methods

The hierarchical clustering can be agglomerative or divisive. The agglomerative type starts with ***n*** single tone clusters and at each step two of the clusters are joined until all ***n*** data objects are in one cluster. The divisive type is the opposite of the agglomerative type so it starts with one cluster containing all ***n*** data objects and at each step one of the clusters is split into two clusters until each data object is in a single cluster. The divisive and the agglomerative types can be seen as top-down and bottom-up methodologies in building the hieratical tree respectively.

4.3 Our system clustering algorithm

As it has been mentioned previously, our system works in unsupervised fashion which needs to label unlabeled data before it can generate the knowledge in the form of decision tree. To label unlabeled data, our system uses a program called FuzME which based on the clustering algorithm called fuzzy k-means algorithm.

4.3.1 K-means algorithm

The ***k***-means algorithm is a partitioning algorithm where the ***k*** is being the number of clusters. The algorithm starts by choosing ***k*** data objects to be used as centroids of the ***k*** clusters. Then each of the other data objects is assigned to the nearest cluster. "In general, the k-means method will produce exactly ***k*** different clusters of greatest possible distinction. K-means clustering algorithm uses an interchange (or switching) method to partition a data set into clusters. An initial partition is given, and new partitions are obtained by switching an object from one cluster to another. MacQueen's (1967) method randomly picks ***K*** points initially, where each point stands for a cluster to be made. A set of points is taken from the data set, and each point is added to the closest cluster. The 'closeness' to the cluster is determined by calculating the distance between a point and the centroid of a cluster-the mean distance of points that belong to that cluster. Then each point is visited to recalculate the distance to the updated clusters. If the closest cluster of the point is not the one it currently belongs, the point will switch to the new cluster. When switching occurs, centroids of both modified clusters have to be recalculated. This procedure is repeated until no more switching takes place. Although the initial points may not generate an optimal solution, Mac Queen's method converge the sum of squared distances (population variance) within the clusters to local optima." [32]. The over all algorithm is governed by the following steps:

- Choose the number of clusters, ***K***.
- Randomly generate ***K*** clusters and determine the cluster centers, or directly generate ***K*** random points as cluster centers.
- Assign each point to the nearest cluster center.
- Recompute the new cluster centers.

- Repeat the two previous steps until some convergence criterion is met (usually that the assignment hasn't changed).

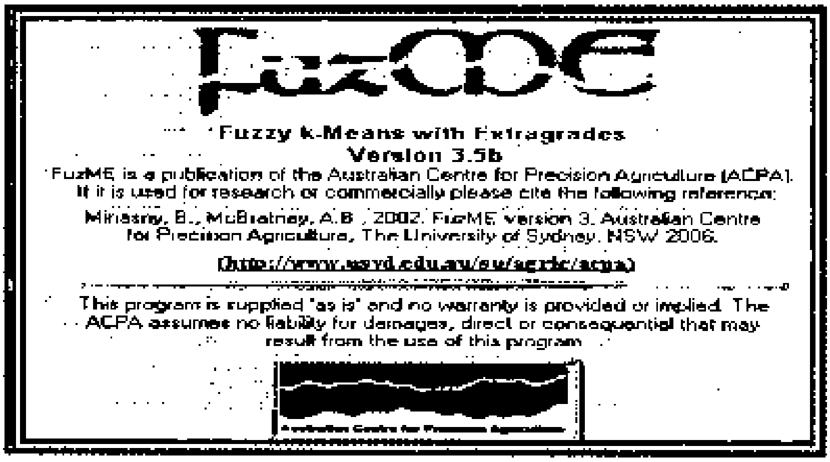
4.3.2 Fuzzy k-means algorithm

In fuzzy clustering, each data object has a degree of belonging in each cluster. So each data object belongs to all clusters with varying degree of membership. Thus, data objects on the edge of a cluster belong to the cluster with lesser degree than data objects that are in the center of the cluster.

In fuzzy K-means clustering, the centroid of a cluster is the average weight of the degree of belonging to the cluster. "Fuzzy-k-means clustering is an extension of the normal, crisp-k-means clustering method to account for uncertainties associated with class boundaries and class membership. As in k-means clustering, the iterative procedure minimizes the within-class sum of squares, but each object (or cell on a map) is assigned a continuous class membership value ranging from 0 to 1 in all classes, rather than a single class membership value of 0 or 1 used in the normal k-means clustering method (DeGruijter and McBratney, 1988). Fuzzy-k-means clustering was conducted using the FuzME program (Minasny and McBratney, 2002) with Mahalanobis distance and a fuzzy exponent of 1.2. Each cell was assigned to a single yield category based on the highest fuzzy membership value at this particular location."[33].

4.4 FuzME program

In our system ***NIU-ID3*** uses FuzME program (based on Fuzzy k-means algorithm) as front-end module to label unlabeled data objects. FuzME program, main widow is depicted in figure-4.1, was published and presented by Minasny B. and McBratney A. in the year of 2002 from the Australian Centre for Precision Agriculture (ACPA) at the University of Sydney, Australia. [34].



Fuzzy k-Means with Extragrades
Version 3.5b

FuzME is a publication of the Australian Centre for Precision Agriculture (ACPA). If it is used for research or commercially please cite the following reference:

Minasny, B., McBratney, A.B., 2002. FuzME version 3. Australian Centre for Precision Agriculture, The University of Sydney, NSW 2006.

(http://www.usyd.edu.au/su/agric/acpa)

This program is supplied "as is" and no warranty is provided or implied. The ACPA assumes no liability for damages, direct or consequential that may result from the use of this program.

Figure-4.1: Main window of FuzME Program.

Sub section 4.6 of this chapter, we will give more details on the FuzME program and its interfaces. According to [34], FuzME is a PC Windows program for calculation of Fuzzy k-means with/without extragrades. It is written in FORTRAN and compiled using Compaq Visual FORTRAN 6.6 under PC Windows's environment. The program needs a "control file" which details with the parameters for the fuzzy k-means algorithm and a data file that contains the data. The program works only under MS-windows environment. FuzME interface is implemented in Visual Basic to create the "control file" and execute the program.

4.5 Data files in FuzME program

4.5.1 The input data file format

According to [34], the data file that can be accepted as input to FuzME program must be in text format, where the first row must be start with the word "id" followed by attributes names. The second and consecutive rows start with the id as a number for each data object followed by the values of the attributes separated by a single space. As an example, figure-4.2 depicts an input to the FuzME program. The data file consists of 14 instances, each of which consists of four attributes.

```
Id Outlook Temperature Humidity Wind
1 Sunny Hot High Weak
2 Sunny Hot High Strong
3 Overcast Hot High Weak
4 Rain Mild High Weak
5 Rain Cool Normal Weak
6 Rain Cool Normal Strong
7 Overcast Cool Normal Strong
8 Sunny Mild High Weak
```

```
9 Sunny cool Normal Weak
10 Rain Mild Normal Weak
11 Sunny Mild Normal Strong
12 Overcast Mild High Strong
13 Overcast Hot Normal Weak
14 Rain Mild High Strong
```

Figure-4.2: Text file format present to FuzME program.

4.5.2 The output files format

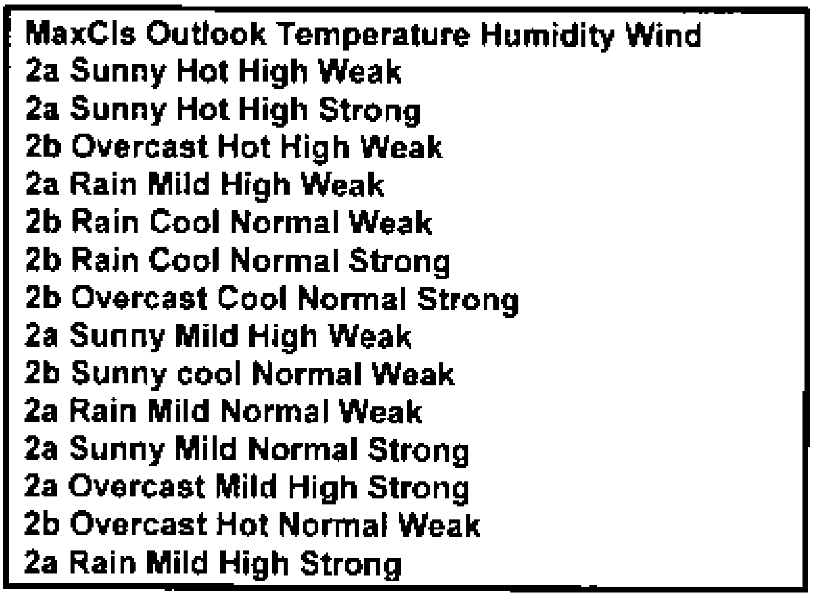
The execution of the FuzME program will generate in many text files as a result (output). The produced text files are: number of files each of them is named as *n_class*, where *n* is the number of produced (i.e. 2_class.txt, 3_class.txt, 4_class.txt, 5_class.txt, etc ...), number of files each of them is named as *n_dscr* that contains description of the produced clustering files (i.e. 2_dscr.txt, 3_dscr.txt, 4_dscr.txt, 5_dscr.txt, etc ...), Control, FuzMeout, pca, and summary. Our system needs only to use only one *n_class* depending on the number of classes the number of cluster that the user had specified.

The output file of the FuzME program is a text file that consists of each object and which cluster that the data object falls in. For example the output text file depicted in figure-4.3 is the result from executing the FuzME program on the data of figure-4.2 and the desired number of cluster is 2, which are coded as 2a and 2b respectively. So the output file name is 2_class.txt.

```
id MaxCls Cl 2a 2b
1 2a 0.09306 0.95347 0.04653
2 2a 0.07672 0.96164 0.03836
3 2b 0.38569 0.19285 0.80715
4 2a 0.00251 0.99875 0.00125
5 2b 0.00072 0.00036 0.99964
6 2b 0.00461 0.00230 0.99770
7 2b 0.00597 0.00299 0.99701
8 2a 0.00298 0.99851 0.00149
9 2b 0.15798 0.07899 0.92101
10 2a 0.04728 0.97636 0.02364
11 2a 0.02214 0.98893 0.01107
12 2a 0.20284 0.89858 0.10142
13 2b 0.04896 0.02448 0.97552
14 2a 0.00245 0.99878 0.00122
```

Figure-4.3: The output of FuzME program.

So, our system needs only to use the second (class) and add it to the original data text file as depicted in figure-4.4 to be used as input to the system.



```
MaxCls Outlook Temperature Humidity Wind
2a Sunny Hot High Weak
2a Sunny Hot High Strong
2b Overcast Hot High Weak
2a Rain Mild High Weak
2b Rain Cool Normal Weak
2b Rain Cool Normal Strong
2b Overcast Cool Normal Strong
2a Sunny Mild High Weak
2b Sunny cool Normal Weak
2a Rain Mild Normal Weak
2a Sunny Mild Normal Strong
2a Overcast Mild High Strong
2b Overcast Hot Normal Weak
2a Rain Mild High Strong
```

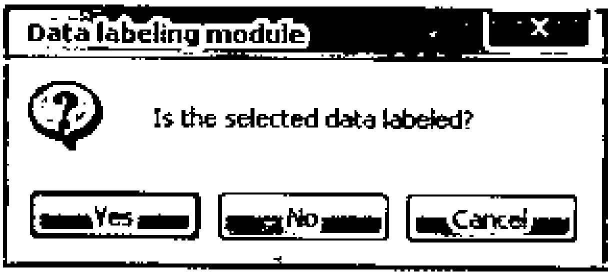
Figure-4.4: Merging of the class attribute with original data.

4.6 Front-End module Implementation

In our system, we link the FuzME program with the implementation of the ID3 algorithm by a shell function, which is considered as a technique of vb.net 2005 used to access to link external objects. "A Shell link is a data object that contains information used to access another object in the Shell's namespace that is, any object visible through Microsoft Windows Explorer. The types of objects that can be accessed through Shell links include files, folders, disk drives, and printers. A Shell link allows a user or an application to access an object from anywhere in the namespace."[35].

The external object that can be accessed or linked to must reside on the current computer disk drives.

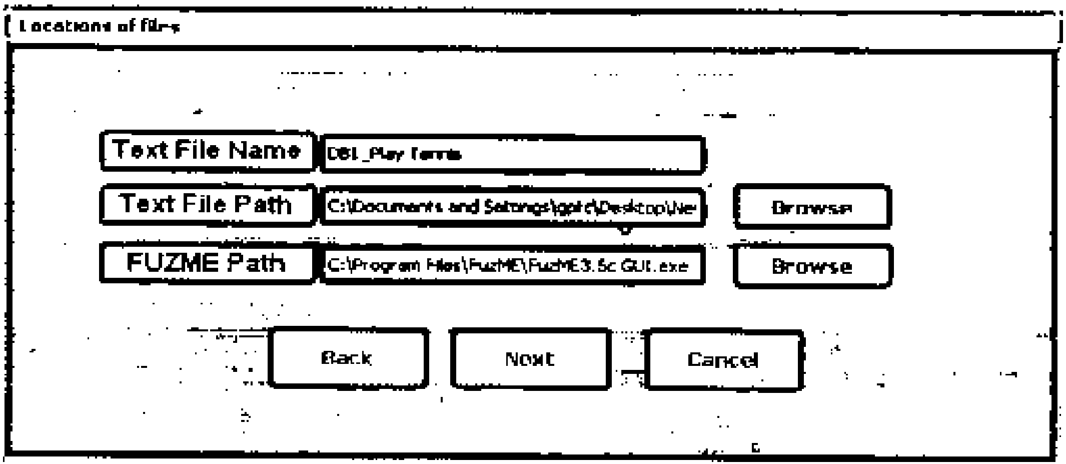
After loading the data file, our system will inquire from the user if the loaded data is labeled or not as depicted in figure-4.5.



Is the selected data labeled?

Figure-4.5: Data labeling dialog box.

In the case of unlabeled data, our system will continue with the execution of the FuzME program in order to label the data. The user must specify the new text file name to be used by FuzME program as input data file with special format, the saving path of new text file, and the locations of the FuzME program as depicted in figure-4.6. The default location of the FuzME program is "C:\Program Files\FuzME\FuzME3.5c GUI.exe" which is determined by the installation of the FuzME program.



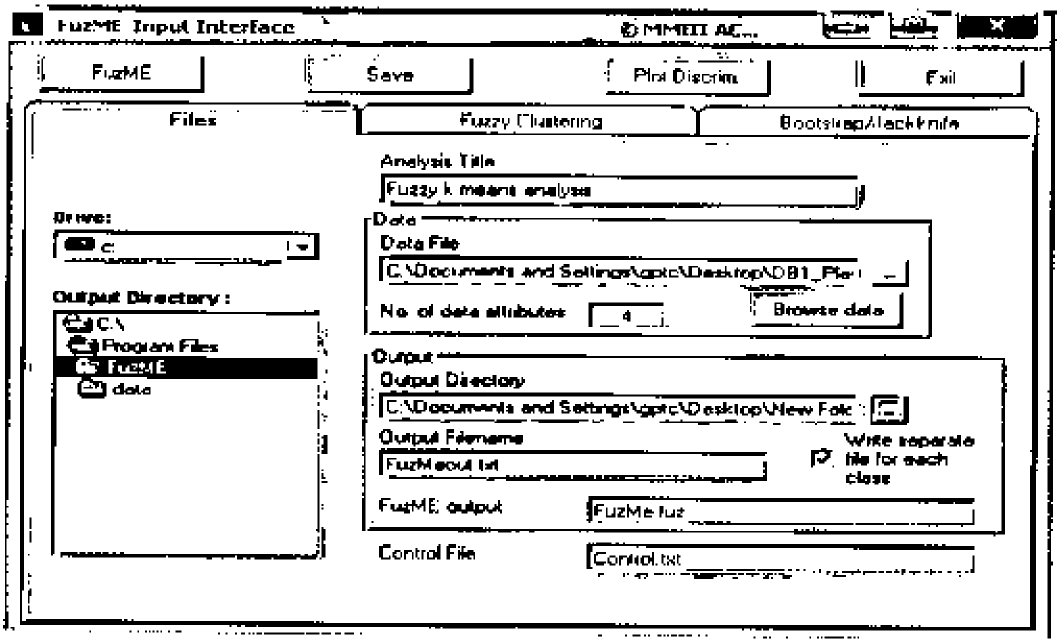
Text File Name: DB1_Play Tennis

Text File Path: C:\Documents and Settings\gptc\Desktop\Ne

FUZME Path: C:\Program Files\FuzME\FuzME3.5c GUI.exe

Figure-4.6: Locations of files.

The next step in our system execution is calling the FuzME program and specifying the needed files locations, the clustering characteristics as well as where to place the results from the program as depicted in figures-4.7(a) and (b).



Analysis Title: Fuzzy k means analysis

Drive: c:

Output Directory :

Data

Data File: C:\Documents and Settings\gptc\Desktop\DB1_Pla

No. of data attributes: 4

Output

Output Directory: C:\Documents and Settings\gptc\Desktop\New Fold

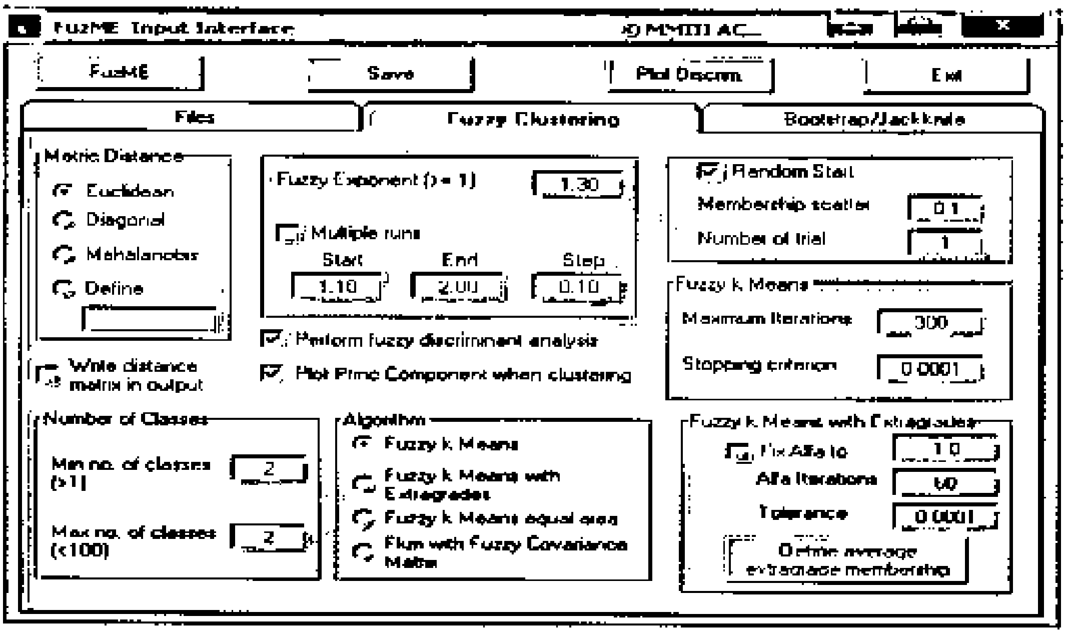
Output Filename: FuzMeout.txt

☑ Write separate file for each class

FuzME output: FuzMe.fuz

Control File: Control.txt

Figure-4.7(a): The locations of files to FuzME program.



Metric Distance: ● Euclidean ○ Diagonal ○ Mahalanobis ○ Define

□ Write distance matrix in output

Fuzzy Exponent (> 1): 1.30

□ Multiple runs — Start: 1.10 End: 2.00 Step: 0.10

☑ Perform fuzzy discriminant analysis

☑ Plot Princ Component when clustering

☑ Random Start — Membership scatter: 0.1 — Number of trial: 1

Fuzzy k Means — Maximum Iterations: 300 — Stopping criterion: 0.0001

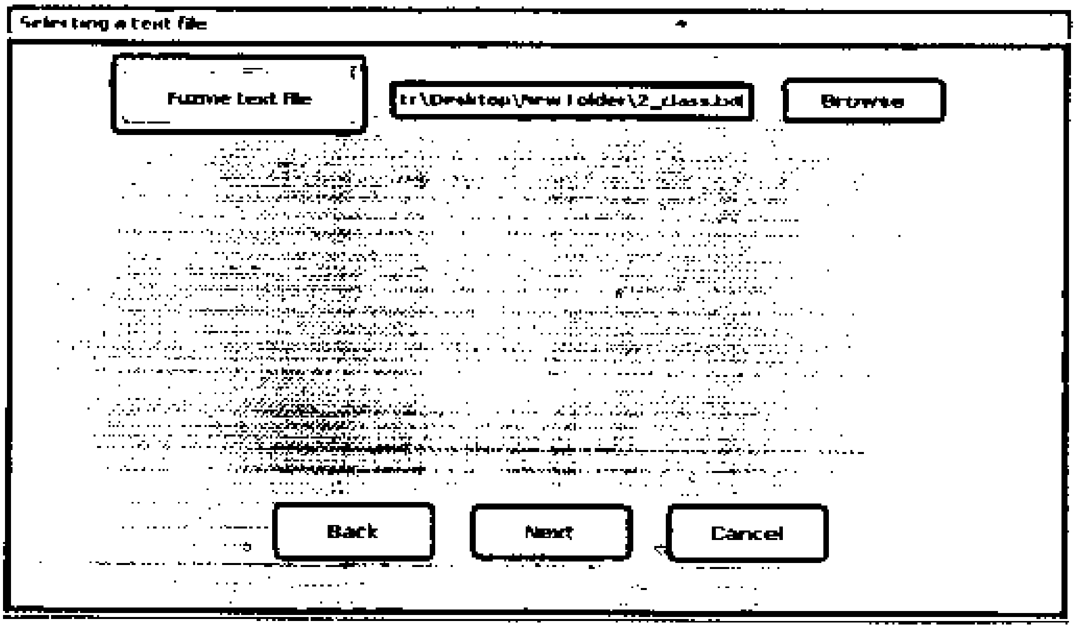
Number of Classes — Min no. of classes (>1): 2 — Max no. of classes (<100): 2

Algorithm: ● Fuzzy k Means ○ Fuzzy k Means with Extragrades ○ Fuzzy k Means equal area ○ Fkm with Fuzzy Covariance Matrix

Fuzzy k Means with Extragrades — □ Fix Alfa to: 1.0 — Alfa Iterations: 60 — Tolerance: 0.0001

Figure-4.7(b): The input characteristics of the clustering to FuzME program.

After the FuzME program had finished its task in labeling the data, then the labels of the data objects are merged with original data to be as input to the classification module (ID3 algorithm implementation) as depicted in figure-4.8.



tr\Desktop\New Folder\2_class.txt

Figure-4.8: Merging of the original data with class labels.

Chapter 5

System testing and results

5.1 Introduction

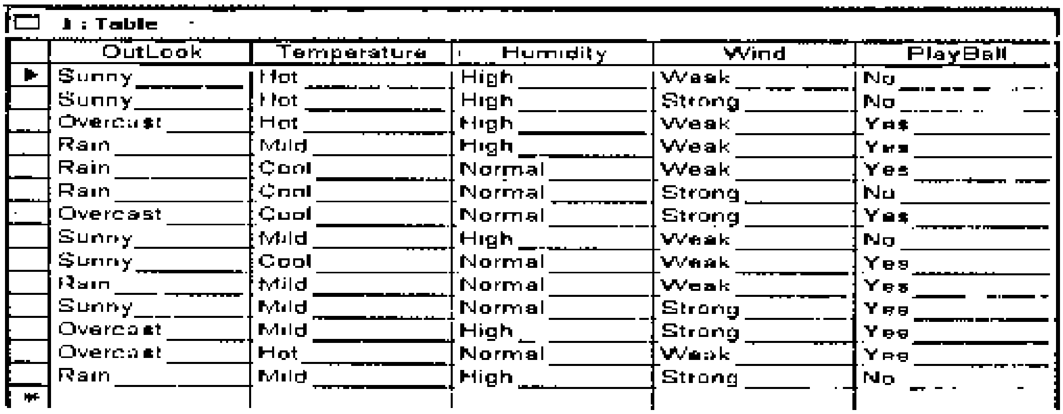
Model testing gives an estimate of any system's accuracy. Model testing computes the accuracy of the model when applied to a data set. This process is mainly concern with application of the system with the intent to find error and pitfalls. Testing software can never completely guaranty the correctness of the software and so testing can provide a criticism or comparison of the state of the software to some given specifications. The purpose of this study is to produce different forms of knowledge and to implement the ID3 algorithm in *supervised* and *unsupervised* fashion using Visual Basic.net 2005. In this chapter, we demonstrate the obtained results of applying our system to different types and sizes of data from a variety of domains. To test the effectiveness of our system, we have conducted some experiments using many real data sets (databases). We used real data in the experiments that available on public domain (the Internet). All of our experiments are performed on a PC with Microsoft Windows XP professional operating system (service pack 2) with a processor speed of 2.7 GHz, RAM size of 512MB and hard disk of size 80 GB. The PC computer is also equipped with Microsoft® Universal Data Access Component 2.5 or 2.6 (MDAC 2.5 or MDAC 2.6) also a reference to Microsoft® Service Component OLEDB Service Component 1.0 stored as; "C:\Program Files\Common Files\System\OLE DB\oledb32.dll.

5.2 Database format and files type that can be loaded to our system

Our system is works with Access database and it can also with text files after some preprocessing in converting the text files into Access database files. Then the actual loading of the data to the system can take place.

5.2.1 Access database file specifications

The Access database file name can be numerical or symbolic, and it must be consists of at least one relational table and the name of the relational table can be numerical or symbolic. The relational table, as depicted in figure-5.1, consists of number of tuples, each of which has number of attributes. The type of attributes can be numeric or symbolic discrete or continuous. Numerical attribute values can be real or integer. Each attribute must have a value, so no missing values are allowed. A symbolic attribute value can be symbols, numbers or both of them. Each tuple corresponds to one instance (example).



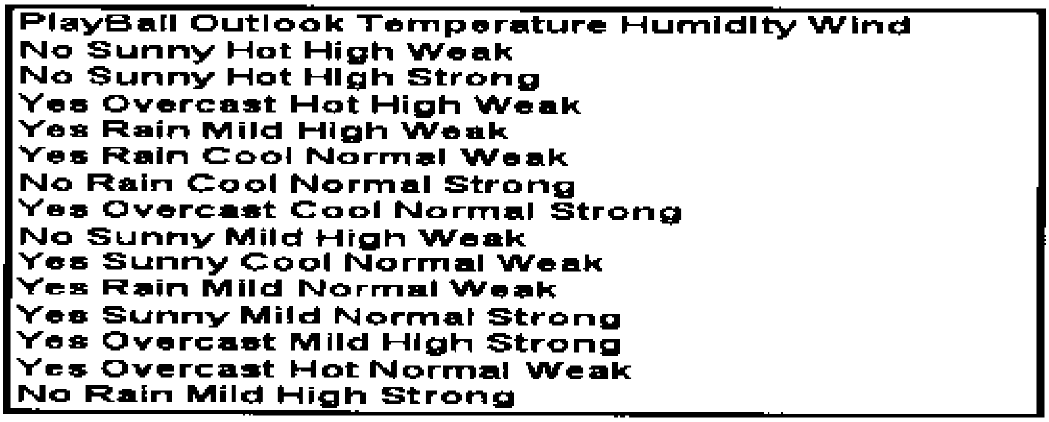
1 : Table

	OutLook	Temperature	Humidity	Wind	PlayBall
▶	Sunny	Hot	High	Weak	No
	Sunny	Hot	High	Strong	No
	Overcast	Hot	High	Weak	Yes
	Rain	Mild	High	Weak	Yes
	Rain	Cool	Normal	Weak	Yes
	Rain	Cool	Normal	Strong	No
	Overcast	Cool	Normal	Strong	Yes
	Sunny	Mild	High	Weak	No
	Sunny	Cool	Normal	Weak	Yes
	Rain	Mild	Normal	Weak	Yes
	Sunny	Mild	Normal	Strong	Yes
	Overcast	Mild	High	Strong	Yes
	Overcast	Hot	Normal	Weak	Yes
	Rain	Mild	High	Strong	No
*					

Figure-5.1: Database format.

5.2.2 Text file specifications

The text file name accepted by our system can be numerical or symbolic, and it can consist of any type of data (numerical or symbolic). Each text file consists of number of rows or lines as depicted in figure-5.2. The first line consists of the attribute names. The second line and subsequent ones consist of the data. The data values can be of any type (i.e. numerical or symbolic). A numerical attribute values can be real or integer. A symbolic attribute value can be symbols, numbers or both. The values in each row must be separated by one space. Each row corresponds to one example or instance and it consists of the values of the attributes. Also, here no missing values are allowed.



```
PlayBall Outlook Temperature Humidity Wind
No Sunny Hot High Weak
No Sunny Hot High Strong
Yes Overcast Hot High Weak
Yes Rain Mild High Weak
Yes Rain Cool Normal Weak
No Rain Cool Normal Strong
Yes Overcast Cool Normal Strong
No Sunny Mild High Weak
Yes Sunny Cool Normal Weak
Yes Rain Mild Normal Weak
Yes Sunny Mild Normal Strong
Yes Overcast Mild High Strong
Yes Overcast Hot Normal Weak
No Rain Mild High Strong
```

Figure-5.2: Text file format.

5.3 Labeled data experiments

Here, we have conducted number of experiments with data sets varying in size and number of attributes.

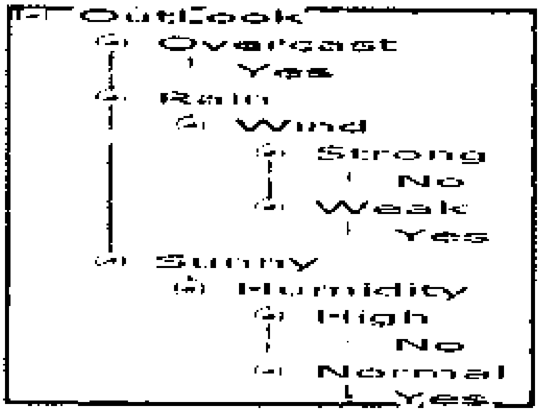
5.3.1 Play tennis data set experiment

This data set is used and tested by many researchers in many papers and researches. The aim of the task here is to learn if the weather conditions are suitable for playing tennis or not. The description of this data set is depicted in table-5.1.

File name is Play tennis.txt or Play tennis.mdb	
Number of instance (examples) is 14 records or instances	
Number of attributes is 5 (including class attribute)	
Attribute's name	**Attribute's values**
Outlook	Sunny, Overcast, Rain
Temperature	Hot, Mild, Cool
Humidity	High, Normal
Wind	Weak, Strong
PlayBall (Class)	Yes, No

Table-5.1: Description of the play tennis data set.

Testing our system with above data set has produced the decision tree that is depicted in figure-5.3(a) a redraw of it is depicted in figure-5.3(b). The results we obtained (all rules and simplified rules as depicted in figure-5.4 and figure-5.5) are exactly the same ones reported in [8, 15, 19, 21; 22, 23, 24 and 25].



```
Outlook
   Overcast
      Yes
   Rain
      Wind
         Strong
            No
         Weak
            Yes
   Sunny
      Humidity
         High
            No
         Normal
            Yes
```

Figure-5.3(a): Decision tree produced of play tennis data set.

For the sake of clarity, we have redrawn the tree of figure-5.3(a) as following:

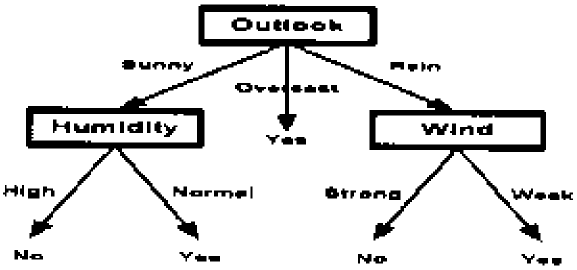
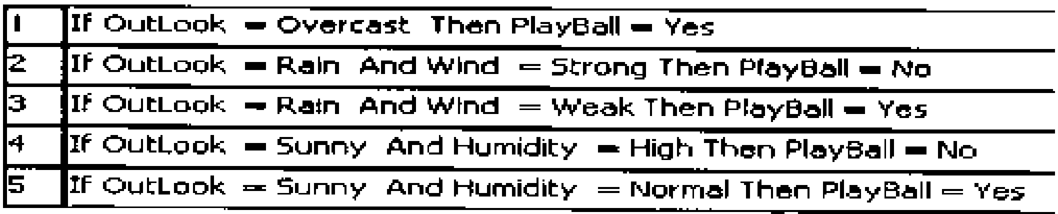
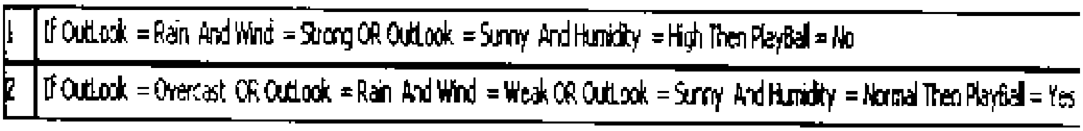


Figure-5.3(b): Redraw of produced play tennis decision tree.



1	If OutLook = Overcast Then PlayBall = Yes
2	If OutLook = Rain And Wind = Strong Then PlayBall = No
3	If OutLook = Rain And Wind = Weak Then PlayBall = Yes
4	If OutLook = Sunny And Humidity = High Then PlayBall = No
5	If OutLook = Sunny And Humidity = Normal Then PlayBall = Yes

Figure-5.4: All rules extracted from play tennis decision tree.

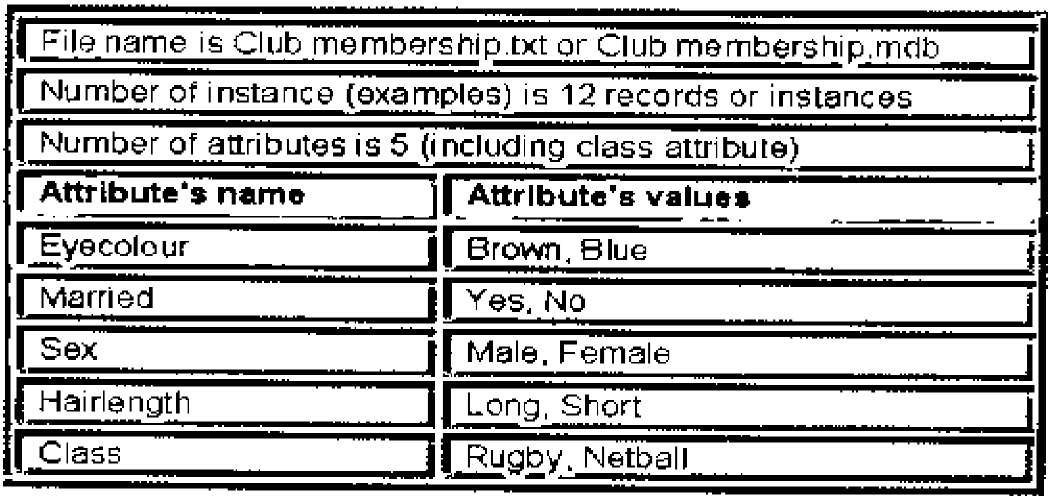


1	If OutLook = Rain And Wind = Strong OR OutLook = Sunny And Humidity = High Then PlayBall = No
2	If OutLook = Overcast OR OutLook = Rain And Wind = Weak OR OutLook = Sunny And Humidity = Normal Then PlayBall = Yes

Figure-5.5: Simplified rules extracted from all rules of play tennis.

5.3.2 Club membership data set experiment

This data set is concerned with 12 University students together with the 'class' to which each belongs - in this case whether each is a member of the University's Rugby club or its Netball club [18]. The data description is depicted in table-5.2.



File name is Club membership.txt or Club membership.mdb

Number of instance (examples) is 12 records or instances

Number of attributes is 5 (including class attribute)

Attribute's name	Attribute's values
Eyecolour	Brown, Blue
Married	Yes, No
Sex	Male, Female
Hairlength	Long, Short
Class	Rugby, Netball

Table-5.2: Description of club membership data set.

We have tried this data in our system and we obtained the results as depicted in figures-5.6(a), 5.6(b) and 5.7. All the obtained results are the same results that had been published in [18].

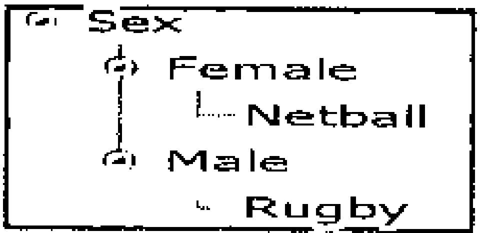


Figure-5.6(a): Decision tree produced of club membership data set.

For the sake of clarity, we have redrawn the tree of figure-5.6(a) as following:

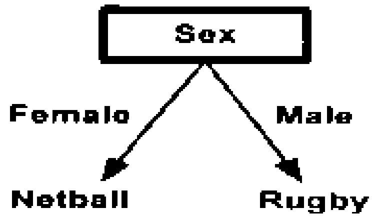
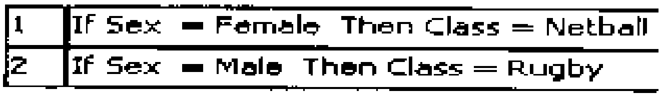


Figure-5.6(b): Redraw of produced club membership decision tree.

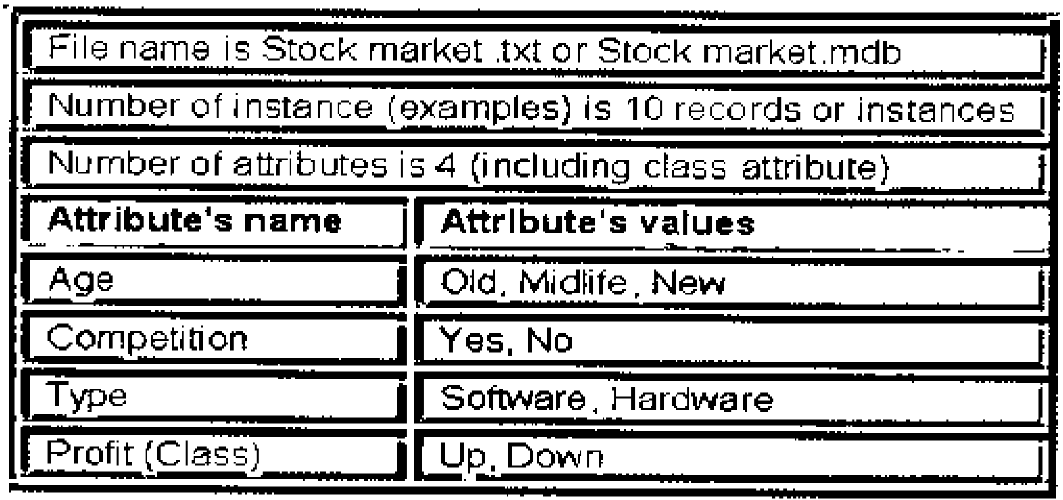


1	If Sex = Female Then Class = Netball
2	If Sex = Male Then Class = Rugby

Figure-5.7: All rules and simplified rules extracted from club membership decision tree.

5.3.3 Stock market data set experiment

This data set aims to predict the increase or decrease a company profits given its profile. This data set had been studied in [36 and 37] and the description of the data set is depicted in table-5.3.



File name is Stock market .txt or Stock market.mdb

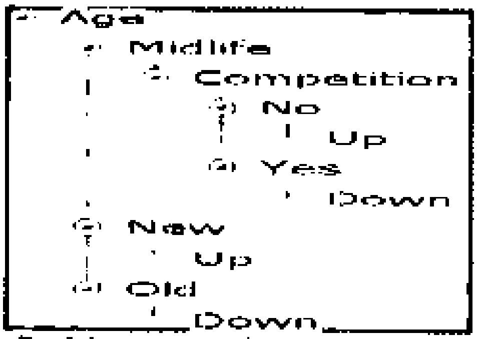
Number of instance (examples) is 10 records or instances

Number of attributes is 4 (including class attribute)

Attribute's name	Attribute's values
Age	Old, Midlife, New
Competition	Yes, No
Type	Software, Hardware
Profit (Class)	Up, Down

Table-5.3: Description of the stock market data set.

We have tried this data in our system and we obtained the results as depicted in figures-5.8(a), 5.8(b), 5.9 and 5.10. All the obtained results are the same results that had been published in [36 and 37].



```
Age
  Midlife
     Competition
        No
           Up
        Yes
           Down
  New
     Up
  Old
     Down
```

Figure-5.8(a): Decision tree produced of stock market data set.

For the sake of clarity, we have redraw the tree of figure-5.8(a) as following:

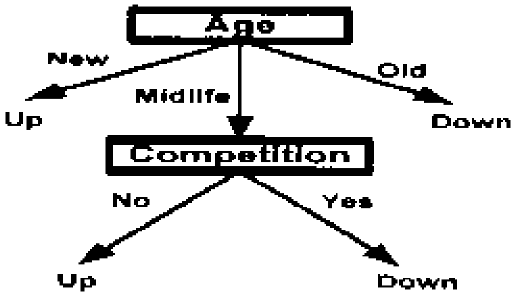


Figure-5.8(b): Redraw of produced stock market decision tree.

1	If Age = Midlife And Competition = No Then Profit = Up
2	If Age = Midlife And Competition = Yes Then Profit = Down
3	If Age = New Then Profit = Up
4	If Age = Old Then Profit = Down

Figure-5.9: All rules extracted from stock market decision tree.

1	If Age = Midlife And Competition = Yes OR Age = Old Then Profit = Down
2	If Age = Midlife And Competition = No OR Age = New Then Profit = Up

Figure-5.10: Simplified rules extracted from all rules of stock market.

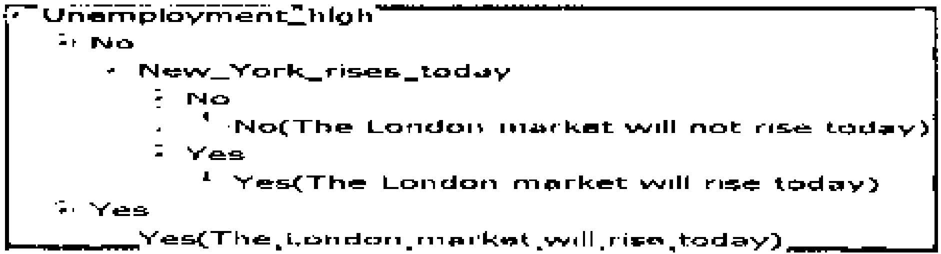
5.3.4 London stock market data set experiment

This data set is about the stock market of London that has been studied in [25]. The aim of this data set is to predict the fall or increase of the London stock market price. The description of the data set is depicted in table-5.4.

File name is London stock market txt or London stock market.mdb	
Number of instance (examples) is 6 records or instances	
Number of attributes is 6 (including class attribute)	
Attribute's name	**Attribute's values**
It_rose_yesterday	Yes, No
New_York_rises_today	Yes, No
Bank_rate_high	Yes, No
Unemployment_high	Yes, No
England_is_losing	Yes, No
It_rises_today (Class)	Yes(The London market will rise today). No(The London market will not rise today)

Table-5.4: Description of the London stock market data set.

We have tried this data in our system and we obtained the results as depicted in figures-5.11(a), 5.11(b), 5.12 and 5.13. All the obtained results are the same results that had been published in [25].



```
Unemployment_high
  No
    New_York_rises_today
      No
        No(The London market will not rise today)
      Yes
        Yes(The London market will rise today)
  Yes
    Yes(The London market will rise today)
```

Figure-5.11(a): Decision tree produced of London stock market data set.

For the sake of clarity, we have redraw the tree of figure-5.11(a) as following:

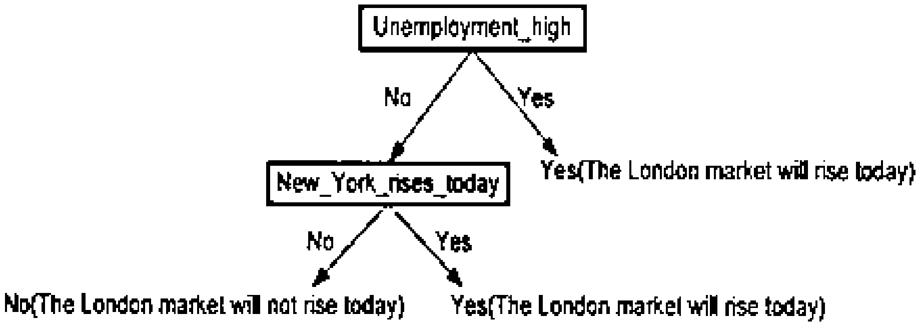
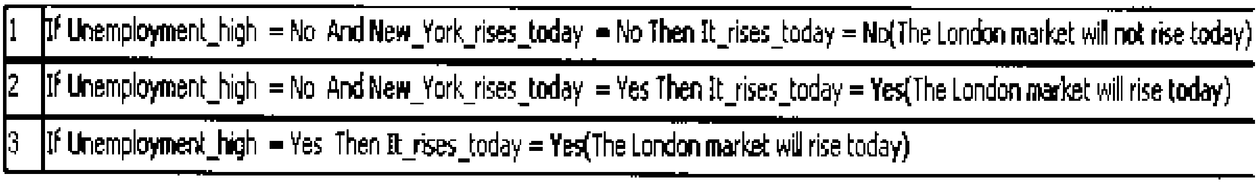
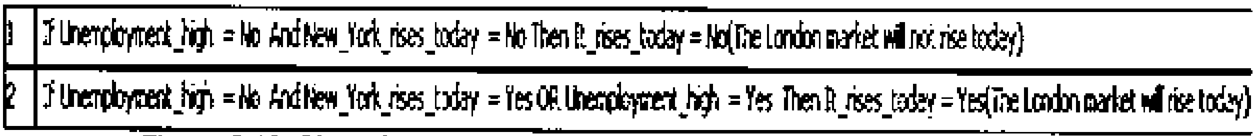

Unemployment_high
No
Yes
Yes(The London market will rise today)
New_York_rises_today
No
Yes
No(The London market will not rise today)
Yes(The London market will rise today)

Figure-5.11(b): Redraw of produced London stock market decision tree.



1	If Unemployment_high = No And New_York_rises_today = No Then It_rises_today = No(The London market will not rise today)
2	If Unemployment_high = No And New_York_rises_today = Yes Then It_rises_today = Yes(The London market will rise today)
3	If Unemployment_high = Yes Then It_rises_today = Yes(The London market will rise today)

Figure-5.12: All rules extracted from London stock market decision tree.

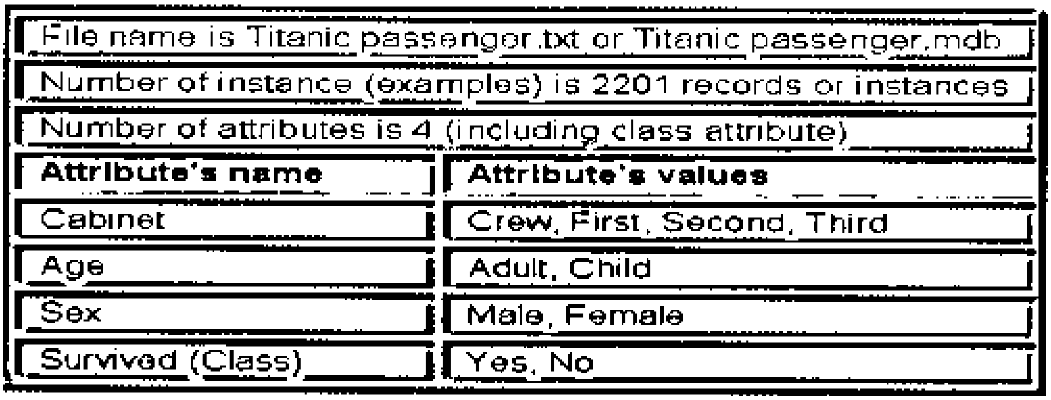


1	If Unemployment_high = No And New_York_rises_today = No Then It_rises_today = No(The London market will not rise today)
2	If Unemployment_high = No And New_York_rises_today = Yes OR Unemployment_high = Yes Then It_rises_today = Yes(The London market will rise today)

Figure-5.13: Simplified rules extracted from all rules of London stock market.

5.3.5 Titanic passenger data set experiment

This data set is about the people who survived the Titanic crash [38] and their location on the ship during the capsized. The description of the data set is depicted in table-5.4.



File name is Titanic passengor.txt or Titanic passenger.mdb	
Number of instance (examples) is 2201 records or instances	
Number of attributes is 4 (including class attribute)	
Attribute's name	**Attribute's values**
Cabinet	Crew, First, Second, Third
Age	Adult, Child
Sex	Male, Female
Survived (Class)	Yes, No

Table-5.5: Description of titanic passenger data set.

We have run our system with this data set and we obtained the results as depicted in figures-5.14(a), 5.14(b), 5.15 and 5.16.

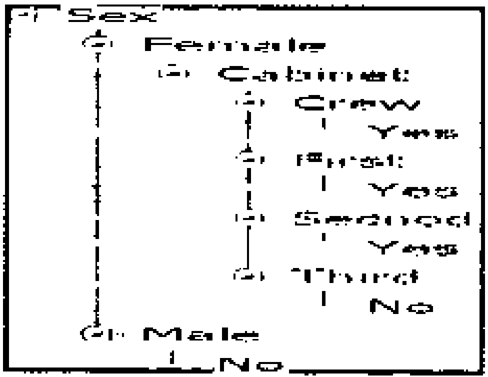


Figure-5.14(a): Decision tree produced of titanic passenger data set.

For the sake of clarity, we have redraw the tree of figure-5.14(a) as following:

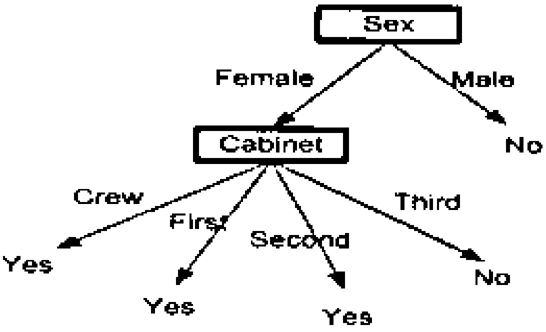
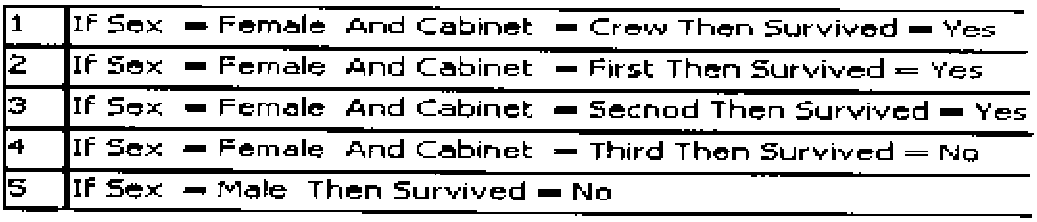
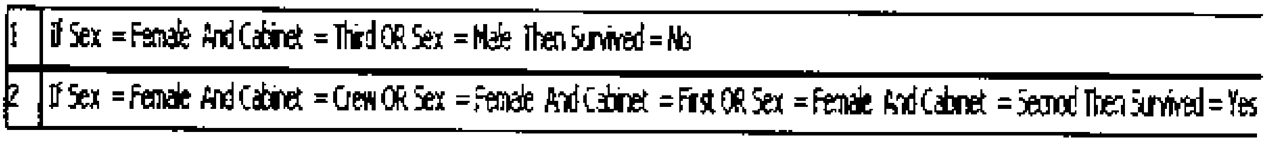


Figure-5.14(b): Redraw of produced titanic passenger decision tree.



1	If Sex = Female And Cabinet = Crew Then Survived = Yes
2	If Sex = Female And Cabinet = First Then Survived = Yes
3	If Sex = Female And Cabinet = Secnod Then Survived = Yes
4	If Sex = Female And Cabinet = Third Then Survived = No
5	If Sex = Male Then Survived = No

Figure-5.15: All rules extracted from titanic passenger decision tree.

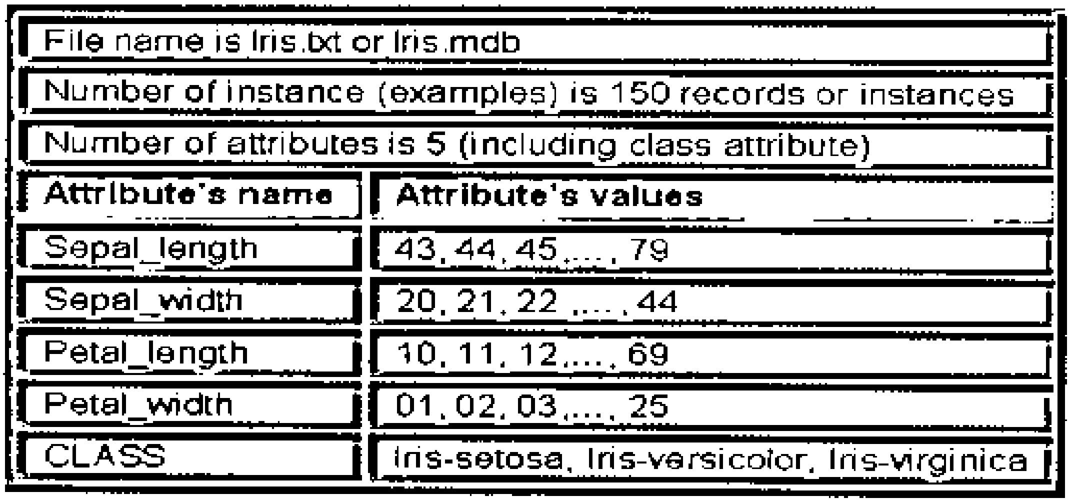


1	If Sex = Female And Cabinet = Third OR Sex = Male Then Survived = No
2	If Sex = Female And Cabinet = Crew OR Sex = Female And Cabinet = First OR Sex = Female And Cabinet = Second Then Survived = Yes

Figure-5.16: Simplified rules extracted from all rules of titanic passenger.

5.3.6 Iris data set experiment

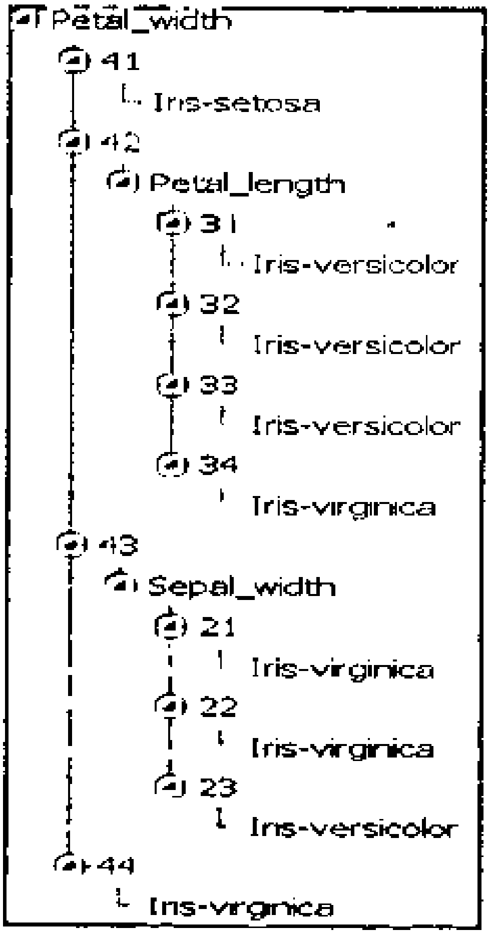
This is a very well known and studied data set [38, 39, 40 and 41]. The data set is about the characteristic of the Iris plant and their diseases. The description of this data set is depicted in table-5.6.



File name is Iris.txt or Iris.mdb	
Number of instance (examples) is 150 records or instances	
Number of attributes is 5 (including class attribute)	
Attribute's name	**Attribute's values**
Sepal_length	43, 44, 45,..., 79
Sepal_width	20, 21, 22 ,..., 44
Petal_length	10, 11, 12,..., 69
Petal_width	01, 02, 03,..., 25
CLASS	Iris-setosa, Iris-versicolor, Iris-virginica

Table-5.6: Description of iris data set.

We have run our system with this data set and we obtained the results as depicted in figures-5.17(a), 5.17(b), 5.18 and 5.19.



```
Petal_width
  41
    Iris-setosa
  42
    Petal_length
      31
        Iris-versicolor
      32
        Iris-versicolor
      33
        Iris-versicolor
      34
        Iris-virginica
  43
    Sepal_width
      21
        Iris-virginica
      22
        Iris-virginica
      23
        Iris-versicolor
  44
    Iris-virginica
```

Figure-5.17(a): Decision tree produced of iris data set.

For the sake of clarity, we have redraw the tree of figure-5.17(a) as following:

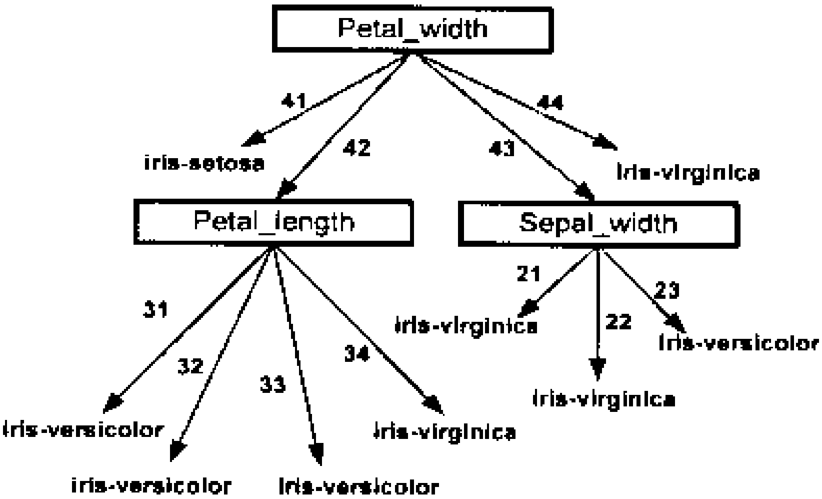
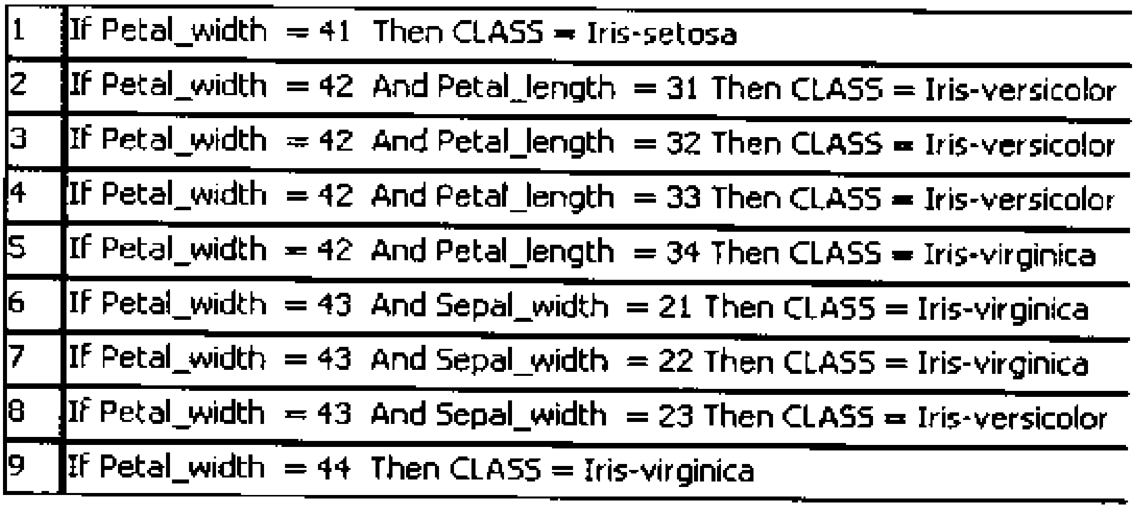
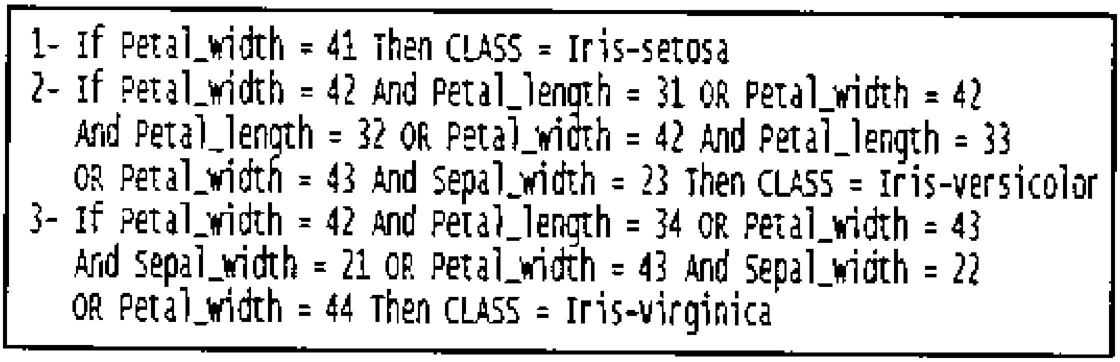


Figure-5.17(b): Redraw of produced iris decision tree.



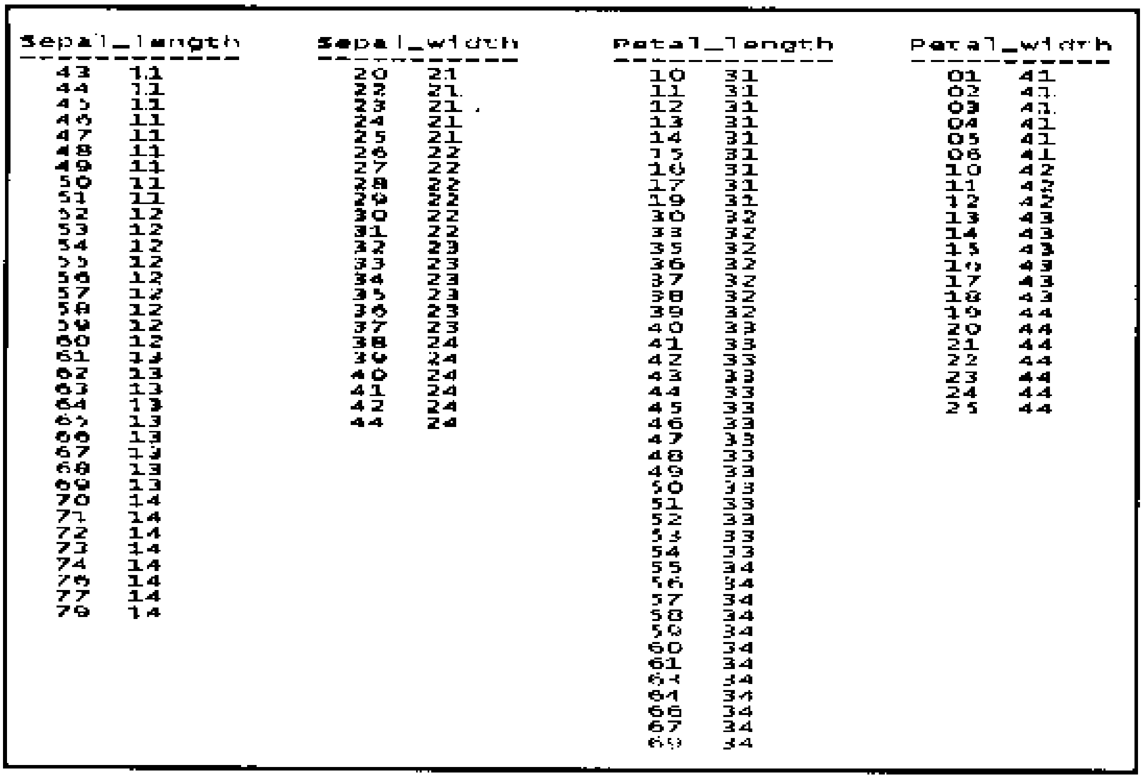
1	If Petal_width = 41 Then CLASS = Iris-setosa
2	If Petal_width = 42 And Petal_length = 31 Then CLASS = Iris-versicolor
3	If Petal_width = 42 And Petal_length = 32 Then CLASS = Iris-versicolor
4	If Petal_width = 42 And Petal_length = 33 Then CLASS = Iris-versicolor
5	If Petal_width = 42 And Petal_length = 34 Then CLASS = Iris-virginica
6	If Petal_width = 43 And Sepal_width = 21 Then CLASS = Iris-virginica
7	If Petal_width = 43 And Sepal_width = 22 Then CLASS = Iris-virginica
8	If Petal_width = 43 And Sepal_width = 23 Then CLASS = Iris-versicolor
9	If Petal_width = 44 Then CLASS = Iris-virginica

Figure-5.18: All rules extracted from iris decision tree.



1- If Petal_width = 41 Then CLASS = Iris-setosa
2- If Petal_width = 42 And Petal_length = 31 OR Petal_width = 42
And Petal_length = 32 OR Petal_width = 42 And Petal_length = 33
OR Petal_width = 43 And Sepal_width = 23 Then CLASS = Iris-versicolor
3- If Petal_width = 42 And Petal_length = 34 OR Petal_width = 43
And Sepal_width = 21 OR Petal_width = 43 And Sepal_width = 22
OR Petal_width = 44 Then CLASS = Iris-virginica

Figure-5.19: Simplified rules extracted from all rules of iris.



```
Sepal_length        Sepal_width        Petal_length        Petal_width
------------        -----------        ------------        -----------
  43   11             20   21            10   31             01   41
  44   11             22   21            11   31             02   41
  45   11             23   21            12   31             03   41
  46   11             24   21            13   31             04   41
  47   11             25   21            14   31             05   41
  48   11             26   22            15   31             06   41
  49   11             27   22            16   31             10   42
  50   11             28   22            17   31             11   42
  51   11             29   22            19   31             12   42
  52   12             30   22            30   32             13   43
  53   12             31   22            33   32             14   43
  54   12             32   23            35   32             15   43
  55   12             33   23            36   32             16   43
  56   12             34   23            37   32             17   43
  57   12             35   23            38   32             18   43
  58   12             36   23            39   32             19   44
  59   12             37   23            40   33             20   44
  60   12             38   24            41   33             21   44
  61   13             39   24            42   33             22   44
  62   13             40   24            43   33             23   44
  63   13             41   24            44   33             24   44
  64   13             42   24            45   33             25   44
  65   13             44   24            46   33
  66   13                                47   33
  67   13                                48   33
  68   13                                49   33
  69   13                                50   33
  70   14                                51   33
  71   14                                52   33
  72   14                                53   33
  73   14                                54   33
  74   14                                55   34
  76   14                                56   34
  77   14                                57   34
  79   14                                58   34
                                         59   34
                                         60   34
                                         61   34
                                         63   34
                                         64   34
                                         66   34
                                         67   34
                                         69   34
```

Figure-5.20: Replacement values of iris data set.

All the obtained results were not the same as the ones in [39, 40 and 41]. Our comments about the differences in the results will be explained in the next chapter.

5.4 Unlabeled data experiments

In the previous subsection, we had used sets of data that are labeled. Here we will use unlabeled data sets in order to examine our system more fully.

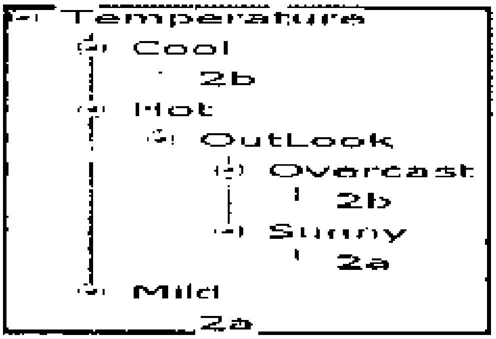
5.4.1 Play tennis data set experiment

Actually, this data set is labeled but we have removed the class attribute, so the data for this experiment became unlabeled. The description of the data set is depicted in table-5.7.

File name is Unlabeled_Play tennis.txt or Unlabeled_Play tennis.mdb	
Number of instance (examples) is 14 records or instances	
Number of attributes is 4	
Attribute's name	**Attribute's values**
Outlook	Sunny, Overcast, Rain
Temperature	Hot, Mild, Cool
Humidity	High, Normal
Wind	Weak, Strong

Table-5.7: Description of the unlabeled play tennis data set.

We have run our system with this data set and we obtained the results as depicted in figures-5.21(a), 5.21(b), 5.22 and 5.23. All the obtained results were not the same as the ones in subsection (5.3.1). Our comments about the differences in the results will be explained in the next chapter.



```
Temperature
  Cool
     2b
  Hot
     OutLook
        Overcast
           2b
        Sunny
           2a
  Mild
     2a
```

Figure-5.21(a): Decision tree produced of unlabeled play tennis data set.

For the sake of clarity, we have redraw the tree of figure-5.21(a) as following:

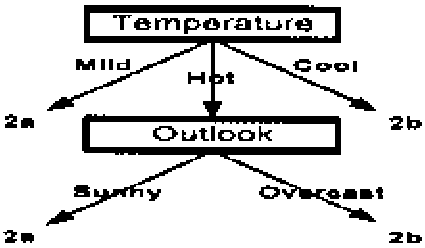
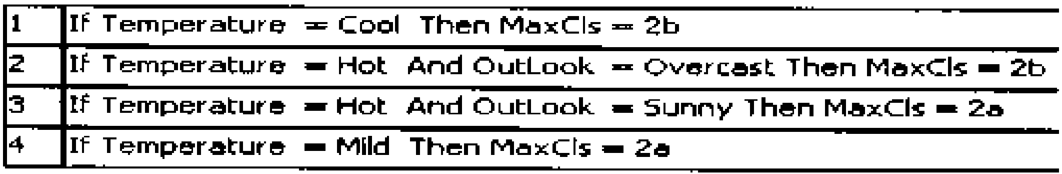
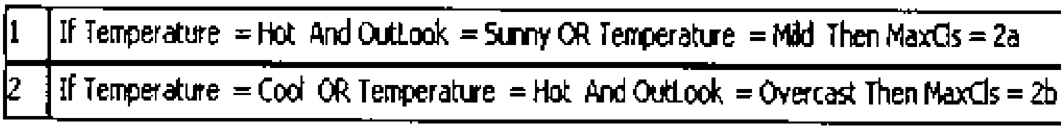


Figure-5.21(b): Redraw of produced unlabeled play tennis decision tree.



1	If Temperature = Cool Then MaxCls = 2b
2	If Temperature = Hot And OutLook = Overcast Then MaxCls = 2b
3	If Temperature = Hot And OutLook = Sunny Then MaxCls = 2a
4	If Temperature = Mild Then MaxCls = 2a

Figure-5.22: All rules extracted from unlabeled play tennis decision tree.

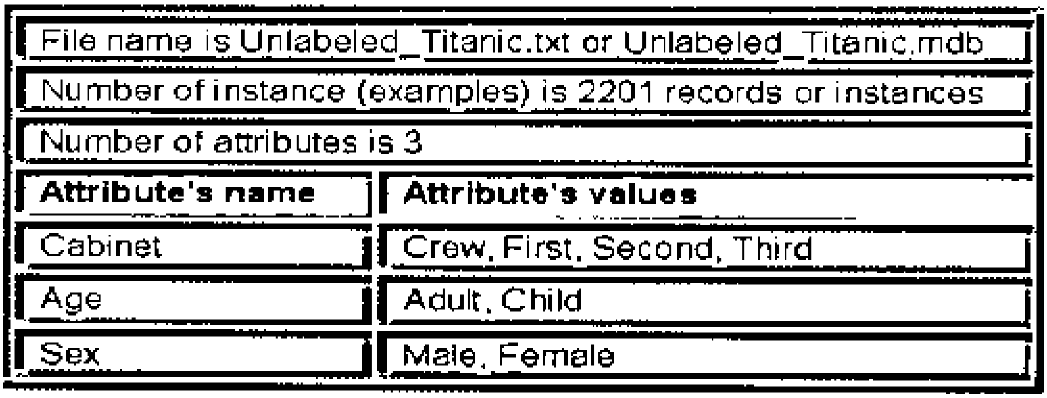


1	If Temperature = Hot And OutLook = Sunny OR Temperature = Mild Then MaxCls = 2a
2	If Temperature = Cool OR Temperature = Hot And OutLook = Overcast Then MaxCls = 2b

Figure-5.23: Simplified rules extracted from all rules of unlabeled play tennis.

5.4.2 Titanic passenger data set experiment

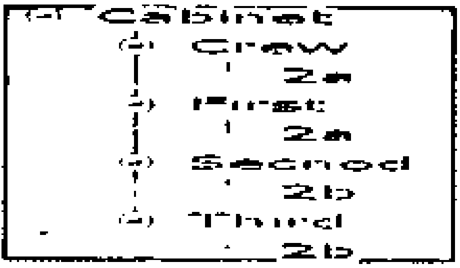
For this data set "the Titanic data set", we have removed the class attribute so it would be unlabeled. The description of the data set is depicted in table-5.8.



File name is Unlabeled_Titanic.txt or Unlabeled_Titanic.mdb	
Number of instance (examples) is 2201 records or instances	
Number of attributes is 3	
Attribute's name	**Attribute's values**
Cabinet	Crew, First, Second, Third
Age	Adult, Child
Sex	Male, Female

Table-5.8: Description of unlabeled titanic passenger data set.

We have run our system with this data set and we obtained the results as depicted in figures-5.24(a), 5.24(b), 5.25 and 5.26. All the obtained results were not the same as the ones in subsection (5.3.5). Our comments about the differences in the results will be explained in the next chapter.



```
(-) Cabinet
     (-) Crew
      |     2a
     (-) First
      |     2a
     (-) Secnod
      |     2b
     (-) Third
            2b
```

Figure-5.24(a): Decision tree produced of unlabeled titanic passenger data set.

For the sake of clarity, we have redraw the tree of figure-5.24(a) as following:

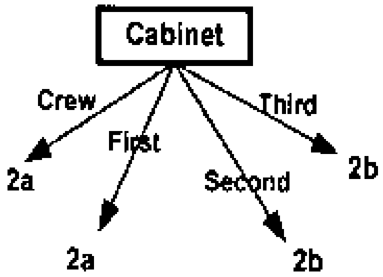
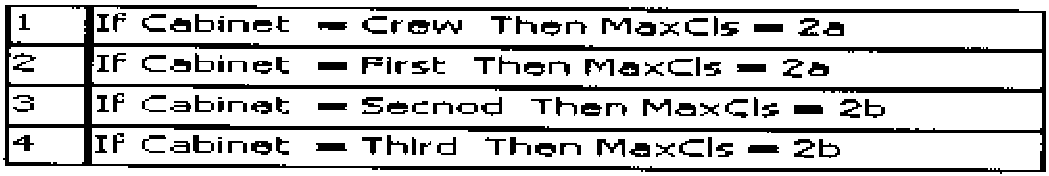
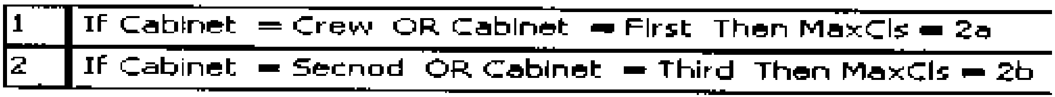


Figure-5.24(b): Redraw of produced unlabeled titanic passenger decision tree.



1	If Cabinet = Crew Then MaxCls = 2a
2	If Cabinet = First Then MaxCls = 2a
3	If Cabinet = Secnod Then MaxCls = 2b
4	If Cabinet = Third Then MaxCls = 2b

Figure-5.25: All rules extracted from unlabeled titanic passenger decision tree.



1	If Cabinet = Crew OR Cabinet = First Then MaxCls = 2a
2	If Cabinet = Secnod OR Cabinet = Third Then MaxCls = 2b

Figure-5.26: Simplified rules extracted from all rules of unlabeled titanic passenger.

Chapter 6

Discussion, conclusion and future work

The objectives of this study are to implement the ID3 algorithm so that it works in unsupervised fashion by the addition of a front-end module to it. The implementation was carried out in Visual Basic.net programming language. In this chapter we will summarize the results of the experiments that were carried out on the implemented system and review the system advantage. At the end of the chapter the discussion and conclusion of this study will be given as well as future research work.

6.1 System advantages

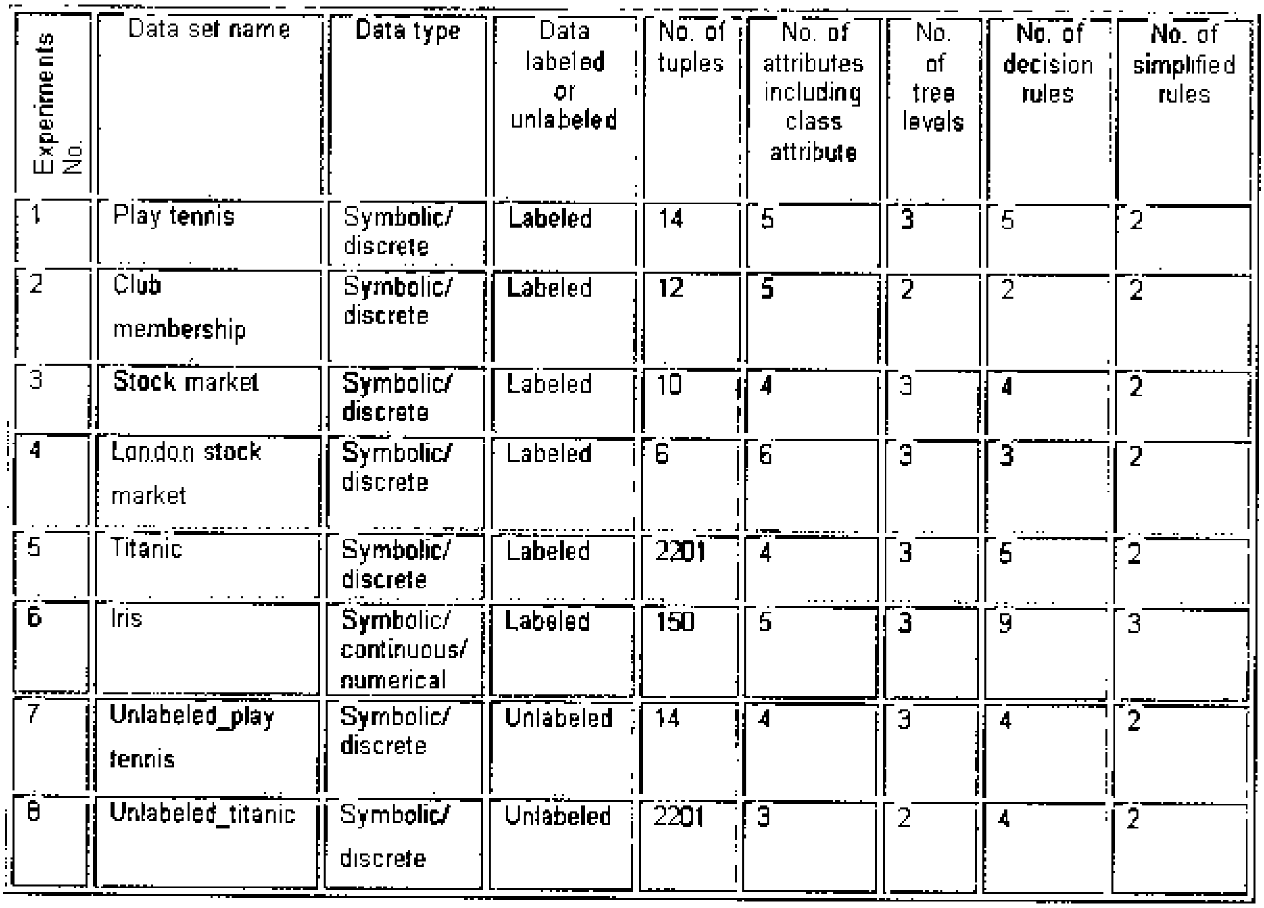
From the short experience with the implemented system (NIU-ID3) and the experiments that were carried out, the author would like to point out the following advantages of the system:

1. The NIU-ID3 system deals with two types of data files; text data files as well as Access database files.
2. The NIU-ID3 system converts the text data file into Access database file very easily.
3. The NIU-ID3 system accepts labeled and unlabeled data.
4. The NIU-ID3 system deals with continuous as well as discrete valued attributes.
5. The NIU-ID3 system discovers missing values in the loaded data and alerts the user to that.
6. The NIU-ID3 system can deal with any number of attributes in the database (data set).
7. The NIU-ID3 system can deal with any number of attribute values in the database (data set).
8. The NIU-ID3 system can deal with any types of attribute values in the database (data set).
9. The NIU-ID3 system displays the end result in several forms (i.e. decision tree, decision rules or general simplified rules).
10. The NIU-ID3 system enables the user to print the decision tree.
11. The NIU-ID3 system enables the user to zoom in and zoom out of the decision tree.
12. The NIU-ID3 system enables the user to save the results of rules in text files.

13. The NIU-ID3 system can be used and useful in several fields of our life, such as; medicines, banks, markets, companies, etc ...

6.2 Discussion

As it has been mentioned in chapter 5, we have conducted a total of 8 experiments with different data sets. The differences in the data sets are in data types and sizes. The results of these experiments are summarized in table-6.1.



Experiments No.	Data set name	Data type	Data labeled or unlabeled	No. of tuples	No. of attributes including class attribute	No. of tree levels	No. of decision rules	No. of simplified rules
1	Play tennis	Symbolic/ discrete	Labeled	14	5	3	5	2
2	Club membership	Symbolic/ discrete	Labeled	12	5	2	2	2
3	Stock market	Symbolic/ discrete	Labeled	10	4	3	4	2
4	London stock market	Symbolic/ discrete	Labeled	6	6	3	3	2
5	Titanic	Symbolic/ discrete	Labeled	2201	4	3	5	2
6	Iris	Symbolic/ continuous/ numerical	Labeled	150	5	3	9	3
7	Unlabeled_play tennis	Symbolic/ discrete	Unlabeled	14	4	3	4	2
8	Unlabeled_titanic	Symbolic/ discrete	Unlabeled	2201	3	2	4	2

Table-6.1: Summary of the experiments results.

Depending on the obtained results from our system, the author would like to make the following remarks:

1. The results obtained from all experiments giving us decision trees with three levels, because we used discretization techniques to reduce the number of values per attribute to 4.
2. The results obtained from experiments no. 1, 2, 3 and 4 are the same results as in [8, 15, 18, 19, 21, 22, 23, 24, 25, 36 and 37].

3. We had no previous results for experiment no. 5, so we could not compare it with previous ones and we think that this result is satisfactory depending on the accurate results that we have obtained in experiments 1 to 4.
4. For experiment no. 6, there were some differences between our results and the results published in [39, 40 and 41]. The differences in results could be due to:
   - The discretization (normalization) technique used in [39].
   - C4.5 (classification) algorithm and the discretization (normalization) technique used in [40 and 41].
5. The results obtained from experiments no. 7 and 8 are different from the ones published in experiments no. 1 and 5, this could be due to the labeling process via the use of FuzMe Program.

6.3 Conclusion

In this research work, we have added a front-end to the ID3 algorithm, so it works in unsupervised mode. Generally, our system consisted of two parts: the first part is the implementation of the ID3 algorithm to be used in classifying labeled data sets; the second part is used to label the unlabeled data sets using FuzMe Program. Our system, NIU-ID3 has been tested with a number of different data sets (labeled, unlabeled and different data types and sizes). We believe that our system will enable decision makers such as; managers, analysts, engineers, physicians, etc... to take the correct decisions.

From our system's results, we can conclude that:

1. Our system has produced very accurate results such as the ones in experiments 1, 2, 3, and 4.
2. The decision trees produced by our system were very clear to visualize.
3. The rules produced by our system were simple to understand and clear to visualize.
4. We think that the results we obtained for experiment no. 5 is satisfactory.
5. The differences in results of experiment 6, with the original results from [39, 40 and 41] could be due to the discretization techniques used and the difference in the classification algorithm or the labeling process such as experiment 7 and 8.

6.4 Future work

The author would like to make a number of suggestions for future research work:

- To test our system with more data in the form of experiments.
- In incorporation of subsystem to deal with pruning techniques.
- In incorporation of subsystem to deal with missing data values and noisy data.
- In incorporation of subsystem to deal with the training data that may outlier instances.
- To develop our system to deal with types of database such as; Oracle database, SQL Server database, etc...
- The study of the possibility to apply this system in Libya in difference fields such as; medicines, banks, markets, companies, etc ...

References

1. C. Apte, T.J., and S.M. Weiss. Data Mining with Decision Trees and Decision Rules, *Paper*, Future generation computer systems, November- 1997.

2. Indira Guzman. Data Mining: A Strategic Decision Support Tool for Organizations, *Paper*, School of information studies, Syracuse University, May-2002.

3. Tsymbal Alexey. Dynamic Integration of Data Mining Methods in Knowledge Discovery Systems, *M.Sc. Thesis*, University of Jyväskylä, December-2002.

4. Claire J. Kennedy and Christophe Giraud Carrier. An Evolutionary Approach To Concept Learning With Structured Data, *Paper*, Department of Computer Science, University of Bristol, Bristol, UK, http://www.cs.bris.ac.uk/Publications/Papers/1000348.pdf, [Web Accessed 14th November 2007].

5. Joakim Nivre. Machine Learning: Basic Concepts, Uppsala University and Vaxjo University, Sweden, http://w3.msi.vxu.se/~nivre/teaching/gslt/basic07ho.pdf, [Web Accessed 25th December 2007].

6. Wen ke Tseng. A Heuristic Partition Method Of Numerical Attributes In Classification Tree Construction, *M.Sc. Thesis*, Department of Information Management, Tatung University, June-2004.

7. Qiong Liu, Stephen Levinson, Ying Wu and Thomas Huang. Interactive And Incremental Learning Via A Mixture Of Supervised And Unsupervised Learning Strategies, *Paper*, Beckman Institute for Advanced Science and Technology, University of illinois at Urbana-Champaign, http://www.ece.northwestern.edu/~yingwu/papers/conference/2000/jcis00.pdf, [Web Accessed 5th June 2006].

8. Erik W. Sinsel. Ensemble Learning For Ranking Interesting Attributes, *M.Sc. Thesis*, College of Engineering and Mineral Resources, West Virginia University, Morgantown, West Virginia, 2005.

9. Gilbert Saporta. Data Mining and Official Statistics, *Paper*, Chaire de Statistique Appliquée, Conservatoire National des Arts et Métiers. 292 rue Saint Martin, Paris, 15 novembre 2000.

10. Usama Fayyad, Gregory Piatetsky Shapiro and Padhraic Smyth. From Data Mining To Knowledge Discovery In Databases, Copyright ©1996, American Association for Artificial Intelligence. 0738-4602-1996.

11. Daniel T. Larose. Discovering Knowledge In Data: An Introduction To Data Mining, Copyright ©2005 by John Wiley & Sons, Inc. Published by John Wiley & Sons, Inc., Hoboken, New Jersey. ISBN 0-471-66657-2 (cloth).

12. George H. John. Enhancements to the Data Mining Process, *PhD. Thesis*, department of computer science, Stanford University, March-1997.

13. Ina Naydenova and Kalinka Kaloyanova. Basic Approaches Of Integration Between Data Warehouse And Data Mining, *Paper*, Faculty of Mathematics and Informatics, University of Sofia, Bulgaria, http://is.fmi.uni-sofia.bg/isdg/reports/DMDWIntegrationLast.doc, [Web Accessed 24th August 2007]

14. Johannes Gehrke, Venkatesh Ganti, Raghu Ramakrishnany and Wei-Yin Lohz. Boat: Optimistic Decision Tree Construction, Department of Computer Sciences and Department of Statistics, University of Wisconsin-Madison, http://www.cs.cornell.edu/johannes/papers/1999/sigmod1999-boat.pdf, [Web Accessed 29th October 2008].

15. J.R, Quinlan. Induction of Decision Trees, Centre for Advanced Computing Sciences, New South Wales Institute of Technology, August-1985, Sydney, Australia.

16. Zhongli Ding. Decision Tree Learning For Negotiation Rules, *M.Sc. Thesis*, Computer Science Electrical Engineering Department, University of Maryland Baltimore County, January-2001.

17. S. D. Cochrane. Knowledge Sharing Between Design And Manufacture, *research project*, Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, UK, April-2004..

18. Professor M.A.Bramer. Induction Of Classification Rules From Examples, Artificial Intelligence Research Group, Department of Information Science, University of Portsmouth, Milton, Southsea, UK, http://www.maxbramer.org.uk/papers/mining.doc, [Web Accessed 29th October 2007].

19. T. Mitchell, Mcgraw Hill. Decision Tree Learning Based On Machine Learning, ch.3, 1997,http://www4.stat.ncsu.edu/~dickey/Analytics/Datamine/Reference%20Papers/machine%20learning.pdf, [Web Accessed 3rd September 2006].

20. Pieter De Leenheer and Mohamed Aabi. Support Vector Machines: Analysis Of Its Behavior & Extension for Large-Scale Problems, *M.Sc. Thesis*, Department of Computer Science, Faculty of Sciences, Vrije Universiteit Brussel, Academic year 2001-2002.

21. Wei Peng, Juhua Chen and Haiping Zhou. An Implementation Of ID3-Decision Tree Learning Algorithm, School of Computer Science & Engineering, University of New South Wales, Sydney, Australia, http://web.arch.usyd.edu.au/~wpeng/DecisionTree2.pdf, [Web Accessed 10th july 2006].

22. Jonas Thulin. Machine Learning: Based Classifiers in the Direkt Profil Grammatical Profiling System, *M.Sc. Thesis*, Department of Computer Science, Lund Institute of Technology, Lund University, Sweden, 11th January-2007.

23. Deniz Yuret and Basak Mutlum. Machine Learning: *Lecture* 10, November-2003, http://www.ai.rug.nl/nl/vakinformatie/KI2/2003/AdvisedReading/ecoe554-10.pdf, [Web Accessed 2nd January 2007].

24. Amos Storkey. Learning From Data: Decision Trees, School of Informatics University of Edinburgh, 2004, http://homepages.inf.ed.ac.uk/amos/lfd/lectures/decisiontree.pdf, [Web Accessed 14th November 2007].

25. HO Tu Bao. Introduction To Knowledge Discovery And Data Mining, *course*, Institute of Information Technology, National Center for Natural Science and Technology, http://www.ebook.edu.vn/?page=1.9&view=1694, [Web Accessed 22th January 2008].

26. Hamad I. Odhabi, Ray J. Paul and Robert D. Macredie. The Four Phase Method For Modelling Complex Systems, Centre for Applied Simulation Modelling (CASM), Department of Information Systems and Computing, Brunel University, Uxbridge, Uk, 1997.

27. http://www.edrawsoft.com/Data-Flow-Diagrams.php, [Web Accessed 15th August 2008].

28. http://www.smartdraw.com/examples/view/index.aspx?catID=.Examples,SmartDraw. Software_Design. [Web Accessed 13th October 2008].

29. Sushmita Mitra and Tinku Acharya. Data Mining: Multimedia, Soft Computing, And Bioinformatics, Copyright ©2003 by John Wiley & Sons, Inc. Published by John Wiley & Sons, Inc., Hoboken, New Jersey. ISBN 0-471-46054-0.

30. Pavel Berkhin. Survey of Clustering Data Mining Techniques, Accrue Software, 1045 Forest Knoll Dr., San Jose, CA, 95129, USA, http://www.ee.ucr.edu/~barth/EE242/clustering_survey.pdf, [Web Accessed 19th October 2007].

31. Nong Ye. The Handbook of Data Mining, Arizona State University, Copyright ©2003 by Lawrence Erlbaum Associates, Inc. ISBN 0-8058-4081-8.

32. Yaoyao Zhu. Unsupervised Database Discovery Based On Artificial Intelligence Techniques, *M.Sc. Thesis*, Department of Electrical & Computer Engineering And Computer Science of the College of Engineering, University of Cincinnati, May-2002.

33. A. Dobermann, J. L. Ping, V. I. Adamchuk, G. C. Simbahan, and R. B. Ferguson. Classification of Crop Yield Variability in Irrigated Production Fields, *Paper*, American Society of Agronomy, 677 S. Segoe Rd., Madison, WI 53711 USA, 2003.

34. http://www.usyd.edu.au/su/agric/acpa/fkme. [Web Accessed 15th July 2007]

35. MSDN Library for Visual Studio 2005.

36. Building Classification Models: ID3 And C4.5, Http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html, [Web Accessed 26th June 2007].

37. Dr Xue Li. A Tutorial on Induction of Decision Trees, School of information technology and electrical engineering, *tutorial*, University Of Queensland, 2002.

38. http://www.ics.uci.edu/~mlearn/MLRepository.html, [Web Accessed 3rd October 2007].

39. Ronny Kohavi, (1996). http://www.sgi.com/tech/mlc/util/util/util.html, [Web Accessed 4th July 2008].

40. Hung Son Nguyena and Sinh Hoa Nguyenb. Fast Split Selection Method And Its Application In Decision Tree Construction From Large Databases, *paper*, Institute of Mathematics, Polish-Japanese Institute of Information Technology, Warsaw University, Poland, 2005, http://logic.mimuw.edu.pl/publikacje/IJHIS_son.pdf, [Web Accessed 12th July 2008].

41. Lawrence O. Hall, Nitesh Chawla and Kevin W. Bowyer. Decision Tree Learning on Very Large Data Sets, *Paper*, Department of Computer Science and Engineering, University of South Florida, http://www.cse.iitb.ac.in/dbms/Data/Papers-Other/Mining/class/smc98.pdf, [Web Accessed 27th April 2008].

Appendix: (list of abbreviations used in the study)

C4.5	C4.5 is an extension of quinlan's earlier ID3 algorithm
DFD	Data Flow Diagram
ID3	Iterative Dichtomizer 3rd
KNN	K-Nearest Neighbors
NIU-ID3	New Implementation Of Unsupervised Id3
NN	Neural Networks
SLIQ	Supervised Learning In Quest
SPRINT	Scalable Parallelizable Induction Of Decision Trees
SQL	Structured Query Language
VB.NET 2005	Visual Basic.Net 2005

الملخص

أن أحجام وكميات البيانات ازدادت بشكل ملحوظ في السنوات القليلة الماضية ، لهذا السبب بعض الباحثين يعتقدون بأن حجم البيانات سيتضاعف كل سنة. لذا التنقيب في البيانات يبدو أنه يكون الحل الأكثر وعودا لمعضلة التعامل مع البيانات الكثيرة جدا والمعرفة القليلة جدا.

تطورت تقنيات قواعد البيانات بشكل مثير منذ السبعينات و التنقيب في البيانات وأصبحتا مساحة جذب كما أنها تعد بتحويل تلك البيانات الخام إلى معرفة ذات مغزى للاعمال التجارية لزيادة ربحيتها.

التنقيب في البيانات عبارة عن مجموعة من التقنيات الحاسوبية المساعدة مصممة للتنقيب الذاتي في أحجام كبيرة من البيانات المتكاملة لاكتشاف معلومات أو نمطيات جديدة غير متوقعة أو مخفية.

أن تقدم تكنولوجيا مجموعة البيانات، مثل نواسخ الباركود الضوئية في الحالات التجارية وأجهزة التحسس في المجالات العلمية والصناعية، ولد كميات ضخمة من البيانات. هذا النمو المتفجر في البيانات وقواعد البيانات ولدا الاحتياج لتقنيات وأدوات جديدة تلك التي من الممكن أن تحول آليا وبشكل ذكي البيانات إلى معلومات ومعرفة مفيدة.

باستعمال تكنولوجيات الإدراك النمطي والتقنيات الرياضية والإحصائية للتدقيق في مخازن المعلومات و التنقيب في البيانات ساعد المحللين في أدراك الحقائق الهامة والعلاقات والنزعات والأنماط والاستثناءات والأشياء الشاذة.

مخازن البيانات و التنقيب في البيانات عبارة عن مجموعة من الأدوات لإدارة وتحليل مجموعات البيانات الكبيرة واكتشاف الأنماط الغير مالوفة. التنقيب في البيانات كان كثير الاستعمال من قبل الخبراء الإحصائيين ومحللو البيانات وطائفة إدارة نظم المعلومات ومحترفون آخرون.

التنقيب في البيانات يمكن أن يستعمل في العديد من الأنواع المختلفة لقواعد البيانات (قواعد البيانات العلائقية و قواعد البيانات الإجرائية و قواعد البيانات الشيئية ومخازن البيانات) أو أنواع أخرى من مخازن المعلومات (قواعد البيانات الحيزية وقواعد بيانات المتوالية الزمنية وقواعد البيانات النصية ومتعددة الوسائط وقواعد البيانات المتوارثة والشبكة العالمية للإنترنت).

بصوره عامة، التنقيب في البيانات هي عملية تحليل البيانات من مناظير مختلفة وملخصة في المعلومات المفيدة. تبدأ بالبيانات الخام وتحصل على النتائج التي من الممكن أن تكون نماذج تنبوية أو نماذج قاعدية أو نماذج تبصريه.

أن نظم التنقيب في البيانات ممكن أن تصنف على أساس فئة معينة من المعايير كالآتي:

- تصنف وفقاً لأنواع قواعد البيانات المنقبة.
- تصنف وفقا لأنواع المعرفة المنقبة.
- تصنف وفقا لأنواع التقنيات المستخدمة.
- تصنف وفق التطبيقات المهيأة.

هذا التصنيف ممكن أيضا أن يكون مفيد للمستخدمين المحتملين لتمييز نظم التنقيب في البيانات ويحدد أفضل نظائر احتياجاتهم المحددة.

أن الغرض من هذا البحث هو تطبيق أحدى تقنيات التنقيب في البيانات (التصنيف) للتعامل مع مجموعات البيانات المصنفة مسبقاً ودمجها مع تقنية أخرى للتنقيب في البيانات (العنقدة) للتعامل مع مجموعات البيانات الغير مصنفة في نظام حاسوب موحد باستخدام لغة البيسك المرئي دوت نت 2005.

نظامنا (NIU-ID3) يمكنه التعامل مع نوعين من ملفات البيانات وهي الملفات النصية و ملفات قواعد البيانات من النوع أكسس. وايضا يمكنه إعادة معالجة البيانات الغير مصنفة ومعالجة البيانات المصنفة.

NIU-ID3 يستطيع اكتشاف المعرفة في شكلين مختلفين و هما: شجرة القرارات و قواعد القرارات (قواعد التصنيفات)، هذه الطريقة تم تنفيذها بواسطة لغة فيجوال بيسك المرئي دوت نت مع لغة الاستفسارات المهيكلة. النظام تم اختباره بعدد من قواعد البيانات من النوع أكسس، وبيانات نصية وتمثيل النتائج على شكل شجرة قرارات، قواعد القرارات أو القواعد المبسطة.

فبراير، 2009

الرقم الإشاري ج.ع/02/1148/2009م

التاريخ :......................
الموافق : 2009 - 7 - 29

كليــة العلــوم

قســم الحاســوب

عنـوان البحـث

تطبيق جـديد لخوارزمية ID3 الغير موجهة باستخدام Visual Basic.Net

مقـدمة مـن الطالـب

أحمـد علـي محمـد الهـوني

** لجنـــة المناقشـــة :

1 – د.فــرج عبــد القــادر المــزوذب ....
(مشــرفاً)

2 – د.عبـد الـرحيم نصـر الصغـير ....
(ممتحنــاً داخليــاً)

3 – د.احــد محمــد أبــوشعــالة ....
(ممتحنــاً خارجيــاً)

د.أحمـد فـرج مسـ...
أمين اللجنة الشعبية لكلية العلوم

29
02
09

جامعة التحدى

كلية العلوم

قسم الحاسوب

تطبيق جديد لخوارزمية ID3 الغير موجهه باستخدام Visual Basic.net

إعداد

أحمد علي محمد الهوني

إشراف

د. فرج عبدالقادر المؤدب

قدمت هذه الرسالة لإستكمال متطلبات الحصول على شهادة
الدرجة العليا الماجستير فى علم الحاسوب

سرت
العام الجامعي 2008-2009