The Great Socialist People Libyan Arab Jamahiriya

University of Garyounis

Faculty of Information Technology

Department of computer science

Data Reduction by the Use of Decision Trees

A Dissertation Submitted to the Faculty of Information Technology in Partial Fulfillment of the Requirements of the Master Degree in Computer Science

Submitted by:

Salma I. Shiltiet

Under the Supervision of:

Dr. Faraj A. El-Mouadib

2010

Dedication

To my Family

Father, Mother, Brother, Sisters and my Husband

Also to my Friends

Salma

Acknowledgment

I would like to express my deep gratitude and thanks to my supervisor Dr. Faraj A. El-Mouadib, for having faith in my ability and for his assistance and guidance throughout my work. I have great respect for his knowledge and criticism that were essential to me and for his invaluable assistance patient guidance and constant encouragement during the preparation of thesis. I appreciate his help more than he known.

Abstract

Due to the availability of enormous amount of saved data in the databases and in the huge data warehouses; the need for developing and for inventing new devices characterized by the capability for analyzing data and for inferring information and knowledge from it has risen. From this need appeared a new field named "data mining" (DM). As a technique, it aims at inferring knowledge from a huge quantity of data.

Data reduction is one of the most practically important steps in the preprocessing process in the mining systems in the data due to the fact that data reduction is choosing a set of data smaller in size provided which maintains the original characteristics and the reliability and the generality of the data.

This work is concerned with the study of data reduction method by using "decision trees" and the production of a computer system in order to program one of the algorithms of the decision trees ID3 in data reduction. The objective of the system production is to prove that there is a big advantage in data reduction while obtaining the same results by using data collection.

The objective of this study is to apply an application that uses decision trees in the process of the reduction of data size as well as the study of the effect of using the concept of decision trees in the field of the data reduction. This work will compare the time of the treatment in constructing the same decision trees once when using the original data and another when using the reduced data. The time total in the first time includes the used time in applying the algorithm ID3 in addition to the time used in drawing the decision tree. The time total in the second time includes the used time in the reduction process and the time used in the application of the algorithm ID3 in addition to the time used in drawing the decision tree.

According to this work through 9 main experiments for different data with different sizes, the system was programmed with the visual basic 6.0. Finally, the obtained results from testing of this system with a detailed analysis of the results will be displayed.

Table of contents

Dedication ii

Acknowledgment iii

Abstract iv

Table of contents v

List of figures vii

List of tables viii

Chapter One: Introduction 1

1.1 Knowledge Discovery in Databases 1

1.2 Data Mining 3

1.3 Data mining functionalities 3

1.3.1 Classification and prediction 3

1.3.2 Cluster analysis 3

1.3.3 Evolution and deviation analysis 4

1.3.4 Association analysis 4

1.3.5 Characterization and discrimination 4

1.4 Preprocessing 4

1.5 Data reduction 5

1.6 Repositories of data 5

1.7 Decision trees 6

1.8 Research goals 7

Chapter Two: Literature review 8

2.1 Data Reduction 8

2.2 Decision trees 10

2.2.1 Building decision trees 12

2.2.2 Decision tree algorithms 12

Chapter Three: Design and implementation 14

3.1 The used programming language 14

3.2 Design of DRS 15
3.2.1 Data entry 15
3.2.2 ID3 algorithm applications 16
3.2.3 Result output 18
3.3 Implementation of DRS 19
Chapter four: Experiments and results 25
4.1 First experiment 25
4.2 Second experiment 31
4.3 Third experiment 32
4.4 Fourth experiment 33
4.5 Fifth experiment 34
4.6 Sixth experiment 35
4.7 Seventh experiment 36
4.8 Eighth experiment 37
4.9 Ninth experiment 38
Chapter Five: Conclusion and further directions 40
5.1 Conclusion 40
5.2 Further works 42

List of figures

1.1 The data mining step as part of KDD process 2

1.2 Major data preprocessing tasks 5

1.3 A decision tree for “buys computer” 7

3.1 Flowchart for DRS 18

4.1 System snap shot of the first version of the first experiment 27

4.2 Results histogram of the first version of the first experiment 28

4.3 System snap shot of the second version of the first experiment 28

4.4 Results histogram of the second version of the first experiment 29

4.5 System snap shot of the fourth version of the first experiment 29

4.6 Results histogram of the fourth version of the first experiment 30

4.7 System snap shot of the fifth version of the first experiment 30

4.8 Results histogram of the fifth version of the first experiment 31

List of tables

2.1 Applicability of data reduction techniques to different types of data 8
4.1 Data set range size 25
4.2 Final results of the first experiment 26
4.3 Final results of the second experiment 31
4.4 Final results of the third experiment 32
4.5 Final results of the fourth experiment 33
4.6 Final results of the fifth experiment 34
4.7 Final results of the sixth experiment 35
4.8 Final results of the seventh experiment 37
4.9 Final results of the eighth experiment 38
4.10 Final results of the ninth experiment 39
5.1 Obtained results from experiments 40

Chapter One

Introduction

In computer science theory there is a clear difference between data and information. According to [12], data are measurements of describing some objects and when these data are organized in some certain manner with the application of some functional or analytical processes, they can be information or knowledge.

Recently the world has seen the collection of massive amounts of data due to the availability and wide use of computers and the cheap cost of massive storage medias. This huge amount of data contains valuable implicit knowledge and the traditional methods of data analysis had lacked the ability to convert such sheer volumes of data into knowledge. The immanent need to convert the data into knowledge which have called for the emergence of a new field known as Knowledge Discovery in Databases (KDD).

1.1 Knowledge Discovery in Databases

Simply stated, the term KDD refers to the broad process of finding knowledge in data, and to emphasize "high level" application of Data Mining (DM). "The term *data mining* has mostly been used by statisticians, data analysts, and the management information systems (MIS) communities"[10].

In literature, many publications in the space of computer science use the term KDD as a synonym for Data Mining, while others consider data mining as one step in the KDD process, as mentioned in [6, 11, 26 and 30]. The field of knowledge discovery in databases, has received increasing attention during recent years as large organizations that have begun to realize the potential value of the information that is stored implicitly in their large amount of data, which are stored in databases. In fact there is no standard definition to what constitute a KDD, but according to Fayyad U. et al. [10]. KDD is the *non-trivial* process of identifying *valid*, *novel*, *potentially useful* and ultimately *understandable* patterns in data.

For a pattern to be valid for a given data, it must pass a certainty threshold requirement. Patterns are characterized as being novel if they are not known to the user or the system.

Patterns are potentially useful if they are beneficial to the system or the user. In order for patterns to constitute useful knowledge, these patterns must be easy to be understood by human. Knowledge discovery in databases as a process is depicted in figure-1.1 [10], which consists of the following steps

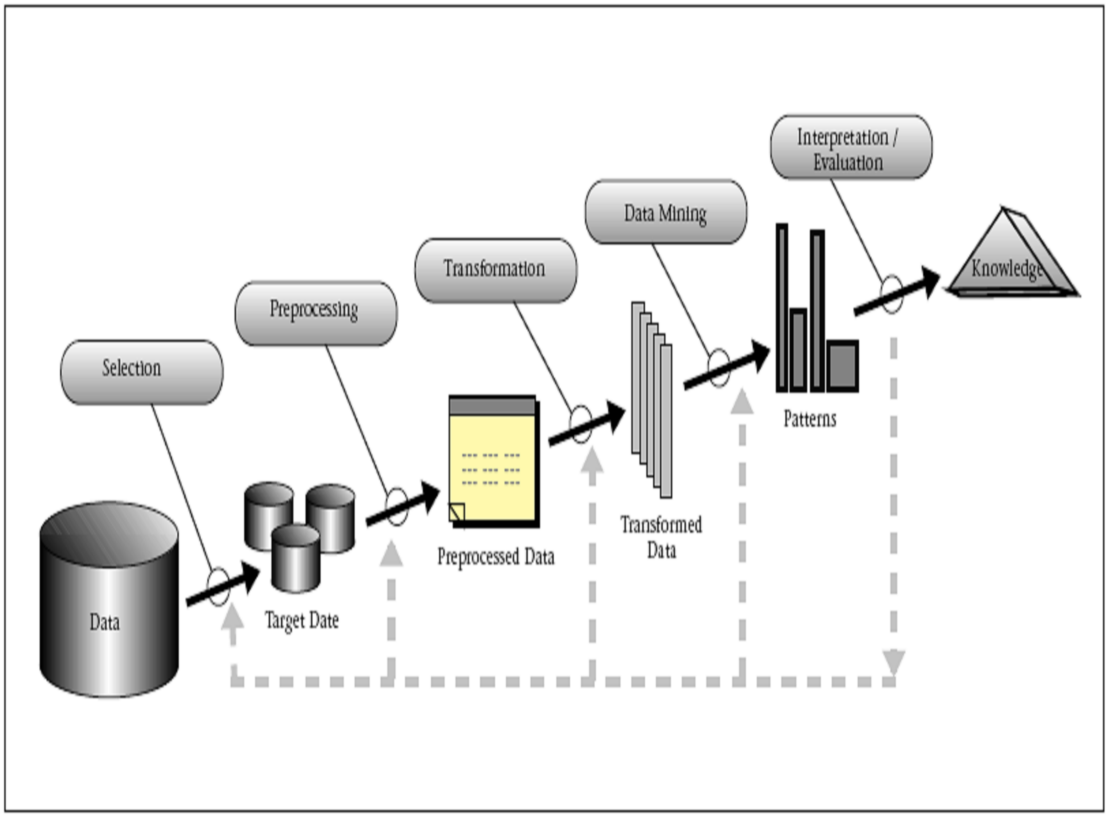


Figure-1.1: The data mining step as part of KDD process.

1. **Data cleaning** it consists of some basic operations, such as normalization, handling of missing data and noise removal, reduction of redundancy, etc. Data from various sources are often wrongful, incomplete, and inconsistent, so we need to clean this data prior to data mining.
2. **Data integration** it plays an important role in KDD, where multiple data sources may be combined together in one common schema.
3. **Data selection** where relevant data are recovered from the database.
4. **Data transformation** is to transform data into forms propitious for mining by applying some operations such as summarization or aggregation for the data.

5. **Data mining** is the essential process where intelligent methods (algorithms) are applied in order to extract useful patterns from the data.
6. **Pattern evaluation** is to identify the interesting patterns representing knowledge based on some interesting measures.
7. **Knowledge presentation** is the method or shape, in which minded knowledge is represented to the end user.

1.2 Data Mining

Data Mining is the process of extracting knowledge from large amounts of data by the use of one of the data mining functionalities. According to [10 and 24], the data mining step is considered as one step in the KDD process. The data mining step is one of the most essential steps in the knowledge discovery process because it constitutes the algorithm by which patterns can be extracted.

1.3 Data mining functionalities

The functionalities of data mining are used to specify the kind of patterns or regularities to be found via a mining task. In general terms, data mining tasks can be categorized as being predictive or descriptive.

According to [12 and 22], predictive mining tasks perform inference on the current data set in order to make predictions for missing or unavailable data values or for missing class labels. Predictive mining tasks are such as; prediction and classification are considered to be supervised learning tasks. Predictive mining tasks are also called inferential tasks. Descriptive mining tasks seek to characterize the general properties or features of the entire data set. Examples of descriptive mining tasks are; cluster analysis, outlier analysis, association analysis, characterization and discrimination. Descriptive mining tasks are considered to be unsupervised learning tasks.

1.3.1 Classification and prediction

According to [24], classification is the discovery of models or functions that governs the general properties of the data. While prediction refers to the use of these models or functions in order to make prediction. Prediction is used to predict class labels for unclassified objects or to predict missing attribute values.

1.3.2 Cluster analysis

Refers to the process of dividing objects into groups or clusters. Where objects in one cluster are similar to one another and objects in different clusters are dissimilar to the objects in other clusters.

1.3.3 Evolution and deviation analysis

Most of the data mining functionalities deal with the general trends of the data, the evolution and deviation analysis deals with the abnormalities or significant changes in the data within some time span.

1.3.4 Association analysis

Association analysis is the process of discovering association rules. Association rule is a statement on the form of IF $condition_1$ THEN $condition_2$ where $condition_1$ and $condition_2$ are attribute-value pairs that occur frequently together in the given set of data, for example if average >= 85 then result = pass.

1.3.5 Characterization and discrimination

Data characterization is the summarization of the general features of the current data set, which is called the target class while data discrimination is the comparison of the features or properties of the target class with one or more classes called contrasting class (es).

1.4 Preprocessing

Data preprocessing is an important step for successful data mining process because it can clean dirty data, smooth noise, correct inconsistent, fill in missing values, transform the data into suitable form, discard irrelevant data (data reduction) and integrate data from multiple different sources. Some of the different data preprocessing tasks are depicted in figure-1.2 [12]. The purpose of data preprocessing tasks are to qualify the data for the current miming task. These tasks are not mutual-exclusive.

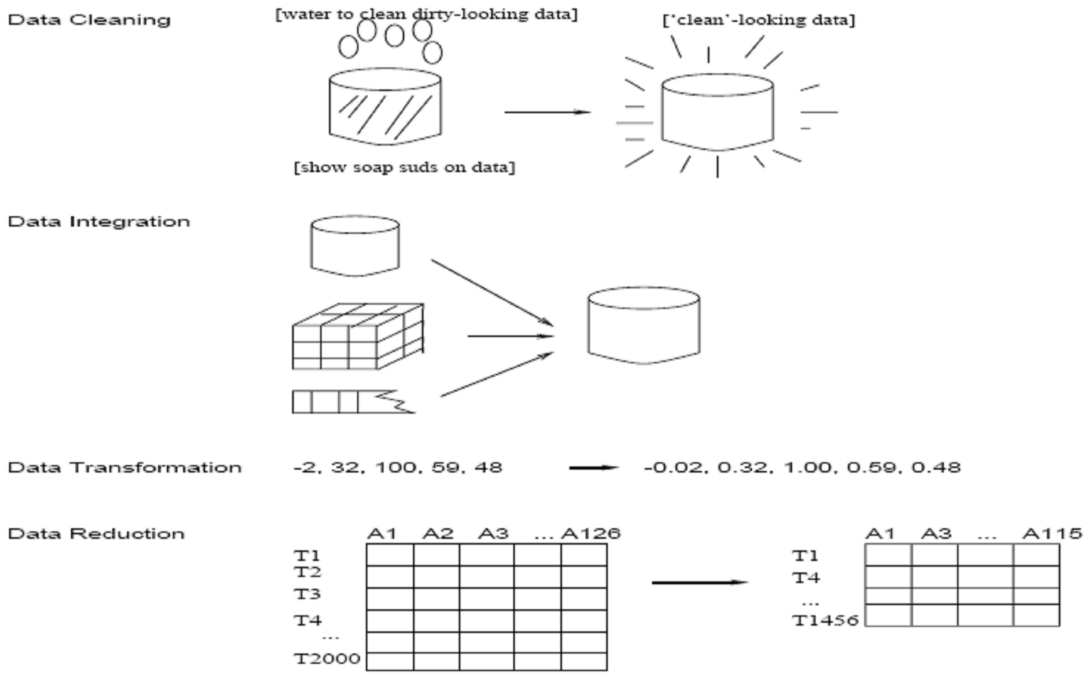


Figure-1.2: Major data preprocessing tasks.

1.5 Data reduction

As mentioned earlier, DM systems are characterized to deal with large data, which make the mining process slow. Many techniques have been developed to speed up the DM process, one of which is data reduction. According to [12 and 24], data reduction refers to the selection of a subset of the data, which is smaller in volume with the condition of maintaining the integrity of the original data. In other words; data reduction is an important preprocessing task in data mining system.

1.6 Repositories of data

There are many different repository systems that are used to house the large volumes of the data. Here, we review some of these repository systems.

- **Data warehouse** is a repository of information collected from multiple sources, organized under a unified schema at a single site in order to facilitate management decision-making. Data warehouses are constructed via processes of; data cleaning, data transformation, data integration, data loading, and periodic data refreshing.
- **Data cube** is a metaphor for multidimensional data storage. The actual physical storage of such data may differ from its logical representation. The important thing to

note is that data cubes are *n*-dimensional and do not confine data to 3-D. which allows data to be modeled and viewed in multiple dimensions

Data reduction techniques can be applied to obtain a reduced version of the data set which smaller in volume, but it represents all states in the data. Mining on reduced data set should be more efficient yet produce the same (or almost the same) analytical results.

There are many strategies for data reduction and according to [12], these strategies are:

- Data cube aggregations is the application of aggregation operations on data cubes in order to construct another data cubes with fewer entries.
- Dimension reduction is accomplished by the removal of irrelevant, weakly relevant or redundant attributes or dimensions.
- Data compression is the reduction of the data by the use of some encoding mechanisms.
- Discretization and concept hierarchies where actual data values are replaced by concepts in higher levels of the concept hierarchy.
- Numerosity reduction is to replace the actual data by an estimation model of it. This can be parametric models (which store only the model parameters rather than the actual data), or nonparametric model such as clustering, sampling, histograms or the use of decision trees.

1.7 Decision trees

"Decision tree induction constructs a flow-chart-like structure where each internal (non-leaf) node denotes a test on an attribute, each branch corresponds to an outcome of the test, and each external (leaf) node denotes a class prediction. At each node, the algorithm chooses the “best" attribute to partition the data into individual classes” [12].

According to [28], a decision tree is constructed in a breath-first fashion; beginning with the root and preceding down word to the leaves. This type of tree is called Top-Down Induction Decision Tree (TDIDT) due to the formation methodology. Figure-1.3 depicts a decision tree for buying a computer [17].

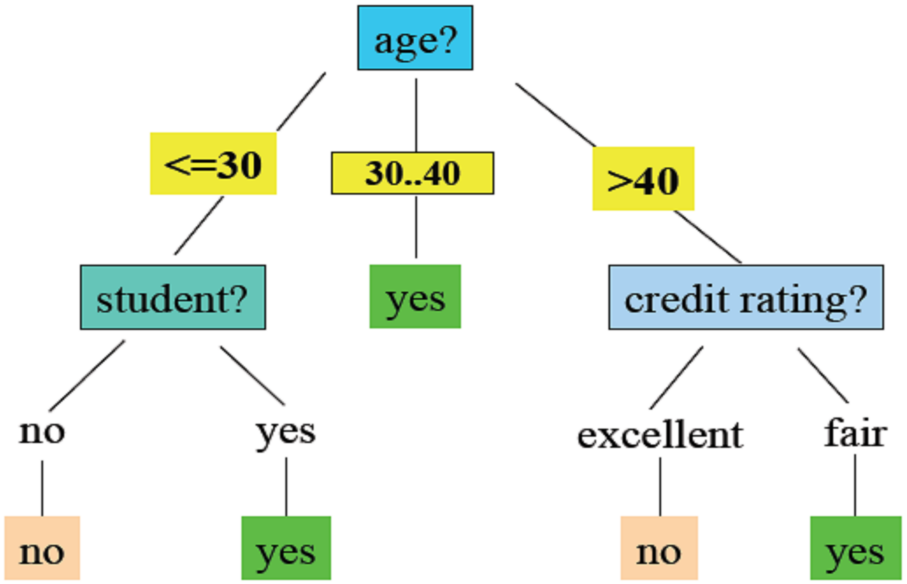


Figure-1.3: A decision tree for "buys computer".

1.8 Research goals

The aim of this study is to implement an application called Data Reduction System (DRS) that uses decision trees in the process of data reduction. The objective of this work is to study the effect of using the concept of decision trees in the area of data reduction. This work will compare the two processing times in building the same decision tree; first with the original data (t_1) the other with the reduced data (t_2). Given that t_1 consists of applying the algorithm and drawing the decision tree process and the t_2 consists of reduction process time and drawing the decision tree process.

Chapter Two

Literature review

This chapter serves as a background survey to the field of data reduction, decision trees and ID3 algorithm. Reduction techniques can be used to draw a minimized replica of the data set which is minimal in size, yet keeps the features of the original data. Mining on the reduced data set is more useful as far as the run time is concerned. However, the outcome result should resemble the analytical results. Nowadays, the world has witnessed a collection of mountains of data due to the presence and wide spread of computers usage and inexpensiveness of enormous storage medias. This enormous quantity of data contains priceless hidden knowledge. The traditional approach of data analysis lack the ability to extract knowledge from huge data bases; Data reduction as a preprocessing step can contribute in solving the scalability problem.

2.1 Data Reduction

Daniel et al. [1] says there is often a necessity to obtain fast approximate answers from large databases, this calls for a need for data reduction; therefore, there are many different approaches to this problem, some of them are not traditionally laid out as solutions to a data reduction problem. The paper describes and evaluates several widespread techniques for data reduction. Where there is an urgent need for fast approximate answers from very large data sets of data stored in databases or data warehouses.

Data reduction is invaluable in this context, and we believe that it is going to be widely used in data mining in the future. According to [1], there already exists a rich variety of data reduction techniques such as; Singular Value Decomposition (SVD), Wavelets, Regression, Log-Linear Models, Histograms, Clustering Techniques, Index Trees, and Sampling with different features and different data types such as Distance Only, Unordered Flat, Unordered Hierarchical, Sparse, Skewed, and High dimensional. Table-2.1, depict the applicability and evaluation of different data reduction techniques with different data types.

Table-2.1: Applicability of data reduction techniques to different types of data.

Data type	Reduction techniques

	SVD	Wavelets	Regression	Log-Linear	Histogram	Clustering	Index Tree	Sampling
Distance Only	N	N	N	N	D	Y	M	Y
Unordered Flat	Y	N	N	Y	D	M	N	Y
Unordered Hierarchical	Y	M	N	Y	M	M	M	Y
Sparse	B	F	F	F	F	B	F	D
Skewed	F	F	B	F	F	F	F	D
High dimensional	N	F	W	W	M	D	W	W

Y= Yes; N= No; M= Maybe;

F= Fine; B= Better; W= Worse;

D = Depends (on further specification, could be better or worse).

The data reduction issue has been studied extensively in the literature on data mining and knowledge discovery. The book by Han and Kamber [12], gives a review of some of the techniques and methods that are used in data reduction. Strategies for data reduction include: Data cube aggregation, Dimension reduction, Data compression, Numerosity reduction, Discrimination and concept hierarchy generation. However, in this thesis we will focus on dimension reduction, where it reduces the data set size by removing attributes (or dimensions) from it.

Bukhman [5], demonstrates a comparison of three deterministic data sampling algorithms; (Epsilon Approximation Sampling Enabled (EASE) Algorithm, Biased EA (Epsilon Approximation) Algorithm and MinSupport_Biased-L2)) and is applied towards generating a data sample that can be used for approximating a data cube of the original dataset. Data reduction techniques have often been used to ease the scalability problem of processing large datasets. Scalability becomes an even greater worry when managing multidimensional datasets and data reduction plays a crucial role as part of the solution to this problem. This thesis explores the use of deterministic sampling algorithms to create a data sample that is usable as an alternative for a multidimensional dataset. The results

indicate that the MinSupport_Biased-L2 algorithm is most suitable for the approximation of iceberg-cubes.

Bronnimann et al. [4], says in his paper, two algorithms are proposed and compared for sampling-based association mining, these algorithms called FAST (Finding Association rules from Sampled Transactions) and EA (Epsilon Approximation), are designed to "count" data applications such as association-rule mining.

The algorithms are similar in that both of them attempt to produce a sample of data that represents the whole database. However, in the way that they greedily search through the exponential number of possible samples FAST is uses random sampling together with trimming of "outlier" transactions. On the other hand, the EA algorithm repeatedly and deterministically halves the data to obtain the final sample. Unlike FAST, the EA algorithm provides a undertaken level of accuracy. The experiments show that EA is more expensive to run than FAST, but yields more accurate results for a given sample size.

2.2 Decision trees

Due to the scientists' interest to facilitate the representation and presentation of data and understanding, there is what is so called trees, which help many people to understand and to simplify the data. It is very easy to obtain useful information and make decisions based on the classifications resulting from those trees; especially what is called decision trees. A decision tree is a tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a classification or decision. Decision tree can easily be converted to classification rules to be used in future classification and it has been used in many application fields such as medicine, business and various theories. They are the basis for several rule induction systems. Decision tree induction algorithms have been used for classification in a wide range of application fields. Besides the learning and classification steps of decision tree induction are normally fast. "The advantages of decision trees include reasonable training time, fast application, easy interpretation, easy implementation, and ability to handle large number of features. Since they do not make any assumptions about the underlying data distribution, they are specially suited for exploratory knowledge discovery." [24].

Quinlan [28], summarizes an approach of synthesizing decision trees that has been implemented in various systems. It describes the ID3 system in details. The outcomes from recent studies indicate ways in which the methodology can be adjusted to deal with information that is noisy and/or incomplete. The paper draws conclusions with illustrations of updated research directions.

Corston-Oliver and Gamon [8], presents a hybrid machine learning approach to correct the features in transfer of linguistic symbols in translation machines. The hybrid method combines decision trees and transformation-based learning. Decision trees serve as a filter to large search space of possible interrelations between characteristics of the data. Transformation-based learning results in a simple set of arranged rules that can be compiled and executed after transfer and prior to sentence realization in the target language. The set of transformations that have been obtained from the decision trees is then used as the set of candidate rules for transformation-based learning. This technique implemented in the domain of machine translation, in order to filter errors in transferred linguistic symbols which are complex and contain large numbers of interdependent characteristics.

Pješivac–Grbović et al. [27], presents an essential step in realizing good performance of Massage Passing Interface (MPI) applications is selecting the close-to-optimal collective algorithm based on the parameters of the collective call at run time. In this paper, the applicability of C4.5 decision tree algorithm to the MPI collective algorithm selection problem was explored where construct C4.5 algorithm from the measured algorithm performance data and analyze the features of a decision tree and anticipated run time performance penalty. This research displays that the C4.5 decision trees can probably be used to generate a small and accurate decision function.

Han and Kamber [12], reviews in there book the concept of decision tree induction, tree pruning, extracting classification rules from decision trees, enhancements to basic decision tree induction, scalability and decision tree induction, and integrating data warehousing techniques and decision trees induction.

Berry and Linoff [2], gives some detailed ideas about decision trees such as; definition of decision trees and how they work. The book also gives some ideas about Neural Networks and how they can be applied to classification and prediction. It describes the

core algorithm that is used to construct decision trees and discusses some of the most outstanding variants of that core algorithm. Practical examples are used to illustrate the utility and general applicability of the decision tree models and are used to demonstrate practical suggestion that have to be accounted for.

Furthermore, in the book by Larose [22], contains classification methods for data mining. One attractive classification method includes the construction of a decision trees, and designates two of the many methods for measuring leaf node purity, which lead to the two leading algorithms for constructing decision trees (Classification And Regression Trees (CART) algorithm and C4.5 algorithm). In addition, it also shows the decision rules, as well as a comparison of the C5.0 and CART algorithms applied to real data.

2.2.1 Building decision trees

According to [3, 12 and 28], there are two approaches in dealing with the construction of decision trees; the first is for building the decision tree and the second is for pruning it. The approach for building the decision tree is as follows:

- Top-down tree construction

  Step1: At the start all data objects are at the root node.

  Step2: Pick the most appropriate attribute to partition the data.

  Step3: Exclude the previously chosen attribute.

  Step4: Pick the most appropriate attribute note chosen before.

  Step5: Partition the data.

  Step6: Recursively perform step2 to step5 until there are no more attribute or all objects are in the same class.

Pruning a decision tree is to remove any sub trees that had occurred in the decision tree. Usually this process is performed in a bottom-up approach.

Here we are concerned with the construction of decision tree using ID3, which does not require pruning process.

2.2.2 Decision tree algorithms

According to [12, 22, 24 and 28], there are many algorithms to construct decision trees, some of them are:

- ID3 (Quinlan 79)
- CART (Brieman et al. 84)

- Assistant (Cestnik et al. 87)
- C4.5 (Quinlan 93)
- See5 (Quinlan 97)
- Orange (Demsar, Zupan 98-03)
- etc …

According to [24], decision trees are claimed to be more efficient due to the fact that the classification is reached faster than Artificial Neural Networks (ANNs) in training phase. This point was tackled by Quinlan's ID3 (Iterative Dichotomiser3) algorithm which is considered as an improvement of the Concept Learning System (CLS). The basic algorithm for decision tree induction is greedy algorithms which construct decision trees in a top-down recursive fashion.

Chapter Three

Design and implementation

In the previous chapters, we have focused on the concepts of data reduction. as well as reviewed in details a very well-known algorithm ID3 that is used to build decision trees. Using this algorithm, the decision trees are constructed in a breath-first fashion. Such methodology is known as TDIDT where building the tree start at the root and proceeds downwards to the leaves. The ID3 algorithm works in a recursive fashion to generate knowledge in the form of a decision tree. The decision trees can easily converted into decision rules.

The aim of this study is to implement an application that uses decision trees in the process of data reduction. Data reduction refers to the selection of a subset of the data, which is smaller in volume with the condition of maintaining the integrity of the original data and the data reduction techniques can be applied to obtain a reduced version of the data set. Mining on a reduced data set should be more efficient, yet it should produce the same (or almost the same) analytical results.

The objective of this work is to study the effect of using the concept of decision trees in the area of data reduction. This work will compare the two processing times in building the same decision tree; first with the original data (t_1) and the other with the reduced data (t_2). Given that t_1 consists of applying the ID3 algorithm and drawing the decision tree process and the t_2 consists of reduction process time, applying the ID3 algorithm and drawing the decision tree process.

The analysis and design of our application will be demonstrated via the concepts of flowcharting.

3.1 The used programming language

Microsoft Visual Studio 6.0 is considers one of the most important programming languages that proceeded Visual Basic .NET that is a language for creating .NET applications. It used to product an application (DRS).

The features of Visual Basic Studio 6.0

According to [9 and 29], there are many features of Visual Basic Studio 6.0 language some of them are:

1. Faster compiler.
2. Lots of icons and pictures to choose from, which allows database integration with wide variety of applications.
3. Fast response to mouse and keyboard actions.
4. Easy access to clipboard and printer.
5. Full array of mathematical, string handling and graphics functions.
6. Ease of handling different types of variables (fixed and dynamic) and control arrays.
7. Support the access to different of types of files, such as; sequential and random access.
8. Support useful facilities for debugging and error-handling.
9. The wizard of package and deployment simplifies the distribution of applications.
10. Provides facilities to design data reports.
11. Provide additional internet capabilities.

3.2 Design of DRS

DRS is designed into a number of processors; each of them has a special purpose. In the follows, we will explain briefly each of these processes.

3.2.1 Data entry

The purpose of the data entry is to provide the system with the data. In our system the entered data must comply with the following conditions:

1. Extension file must be (.DBF)
2. Data must be categorical data (supervised data).
3. Type of data must be nominal.
4. The data must not contain missing values.
5. Last attribute in the table must be the class attribute.

6. Class attribute value must be binary in nature (i.e. yes/no, play/not play, fly/not fly, accept/reject etc …).

Due to the fact that DRS works with clean data and some of the data attributes do not contribute to the process of the system. There are some attributes that must be removed from the provided data set such as:

- The single attribute values.
- The multi attribute values.
- The serial number attributes.

3.2.2 ID3 algorithm applications

For DRS to meat its objectives each run of DRS will apply the ID3 algorithm twice. The first is with the original data and the second is with the reduced data.

The application of ID3 algorithm can be demonstrated in the following steps [15].

```
ID3(in T : table; C : classification attribute)
    return decision tree

{ if (T is empty) then return(null); /* Base case 0 */
  N := a new node;
  if (there are no predictive attributes in T) /* Base case 1 */
    then label N with most common value of C in T (deterministic tree)
                 or with frequencies of C in T (probabilistic tree)
  else if (all instances in T have the same value V of C) /* Base case 2 */
    then label N, "X.C=V with probability 1"
  else { for each attribute A in T compute AVG_ENTROPY(A,C,T);
         AS := the attribute for which AVG_ENTROPY(AS,C,T) is minimal;
         if (AVG_ENTROPY(AS,C,T) is not substantially smaller than ENTROPY(C,T)) /* Base case 3 */
           then label N with most common value of C in T (deterministic tree)
                        or with frequencies of C in T (probabilistic tree).
         else {
              label N with AS;
              for each value V of AS do {
                  N1 := ID3(SUBTABLE(T,A,V),C) /* Recursive call */
                  if (N1 != null) then make an arc from N to N1 labelled V;
                  }
              } }
  return N;
}


SUBTABLE(in T : table; A : predictive attribute; V : value) return table;
{ T1 := the set of instance X in T such that X.A = V;
  T1 := delete column A from T1;
  return T1
}


/* Note: in the textbook this is called I(p(v_1) ... p(v_k)) */
ENTROPY(in C : classification attribute; T : table) return real number;
{ for each value V of C, let p(V) := FREQUENCY(C,V,T);
  return -sum_V p(V) log_2(p(V)) /* By convention, we consider 0 . log_2(0) to be 0. */
}


/* Note; In the textbook this is called "Remainder(A)" */
AVG_ENTROPY(in A: predictive attribute; C : classification attribute; T : table)
  return real number;
{ return sum_V FREQUENCY(A,V,T) . ENTROPY(C,SUBTABLE(T,A,V)) }


FREQUENCY(in B : attribute; V : value; T : table) return real number;
{ return #{ X in T | X.B=V } / size(T); }
```

3.2.3 Result output

The purpose of result output is to display the final result, which consists of:

- The decision tree before data reduction and the processing time.
- The decision tree after data reduction and the processing time.

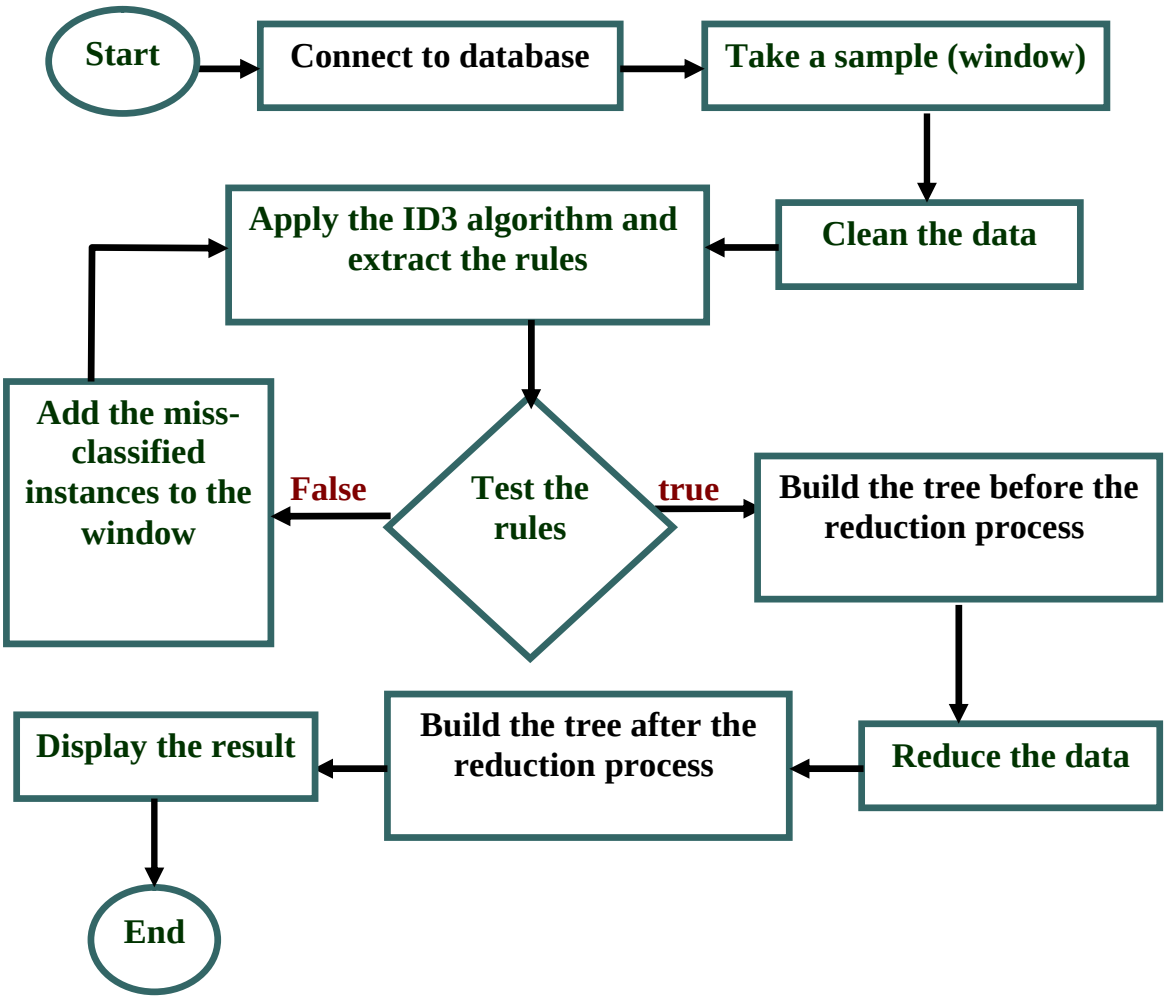


Figure-3.1: Flowchart for DRS.

3.3 Implementation of DRS

In the previous subsections, we have given details of the design of DRS, and here we explain the implementation. DRS is implemented in Microsoft Visual Studio 6.0.

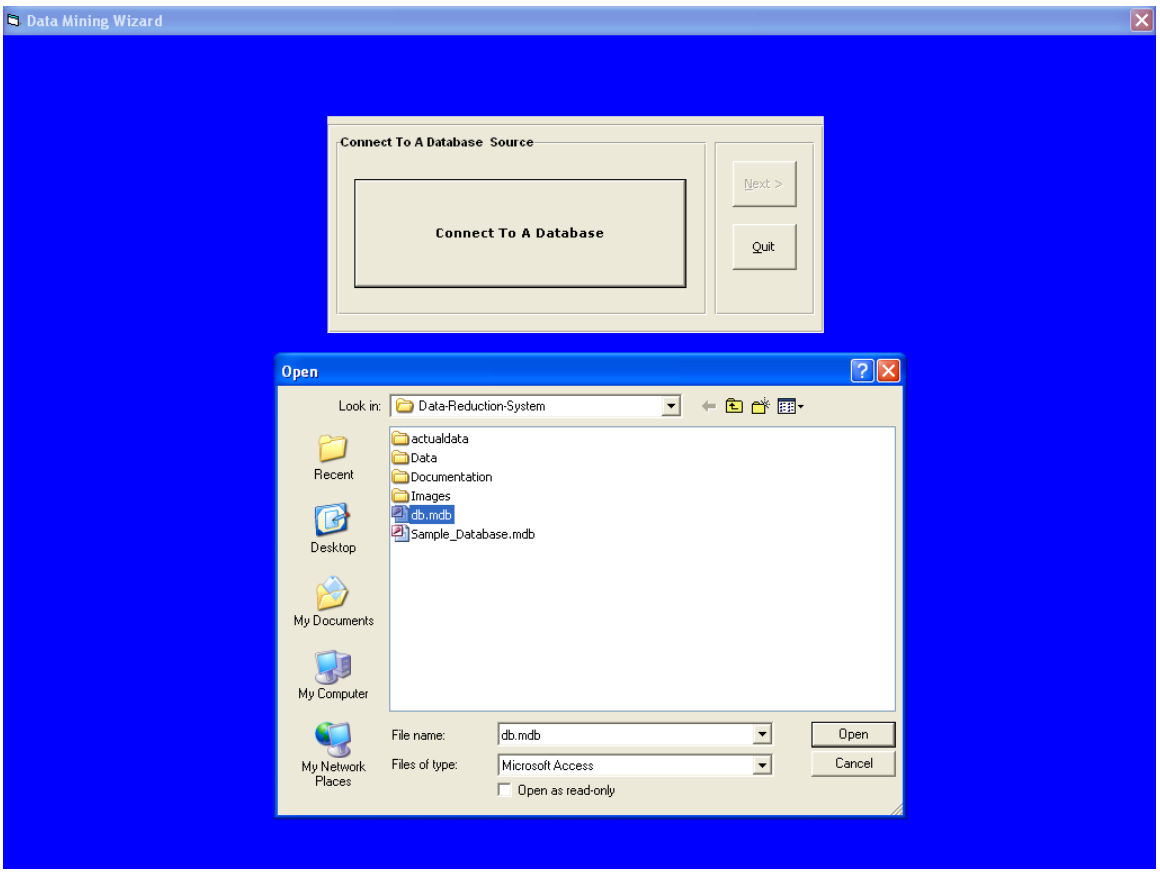
The system starts by taking a sample (window) from the selected data and ends up with the results obtained.

In the first stage of the system, we take the window and deal with it as an input to the system, and then we apply the ID3 to extract rules from the data, and then we build the decision tree with the chosen window and final calculated runtime.

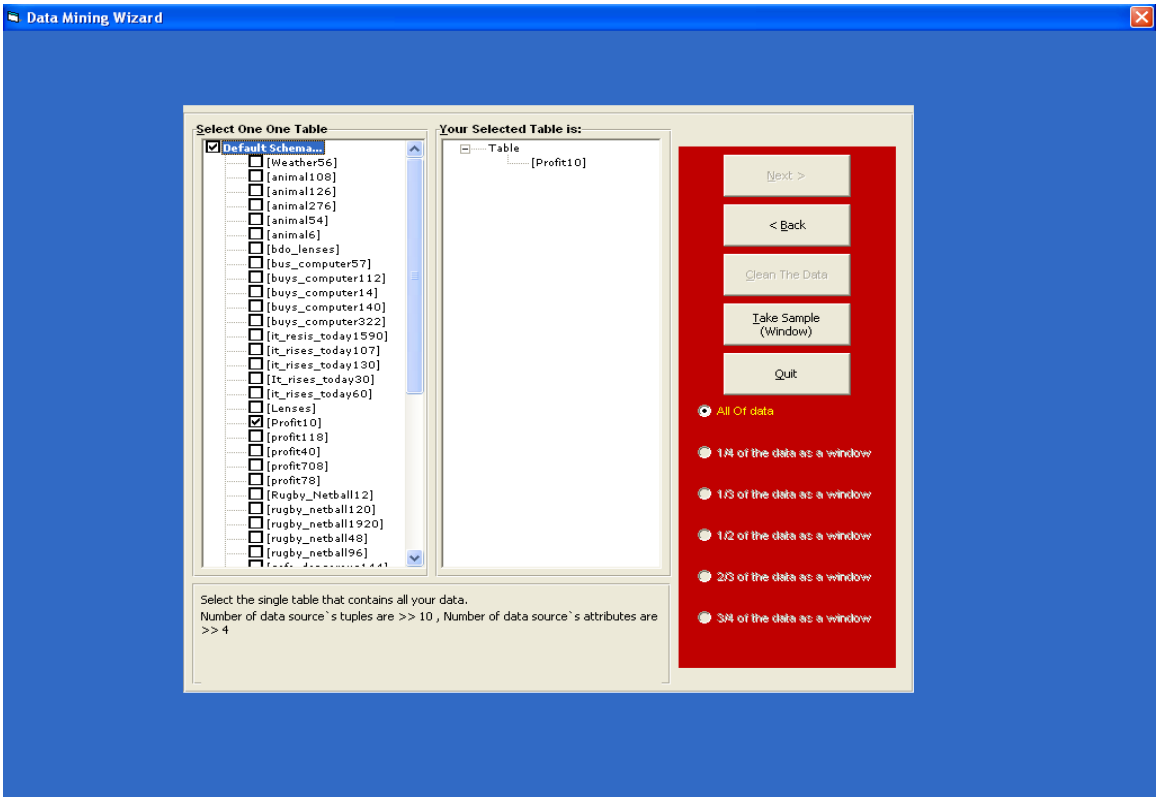
In the second stage, we do the same process as in the first but with reduced set of data. The unused attributes of the data are determined in the first stage.

The implementation of our system consists of the following steps:

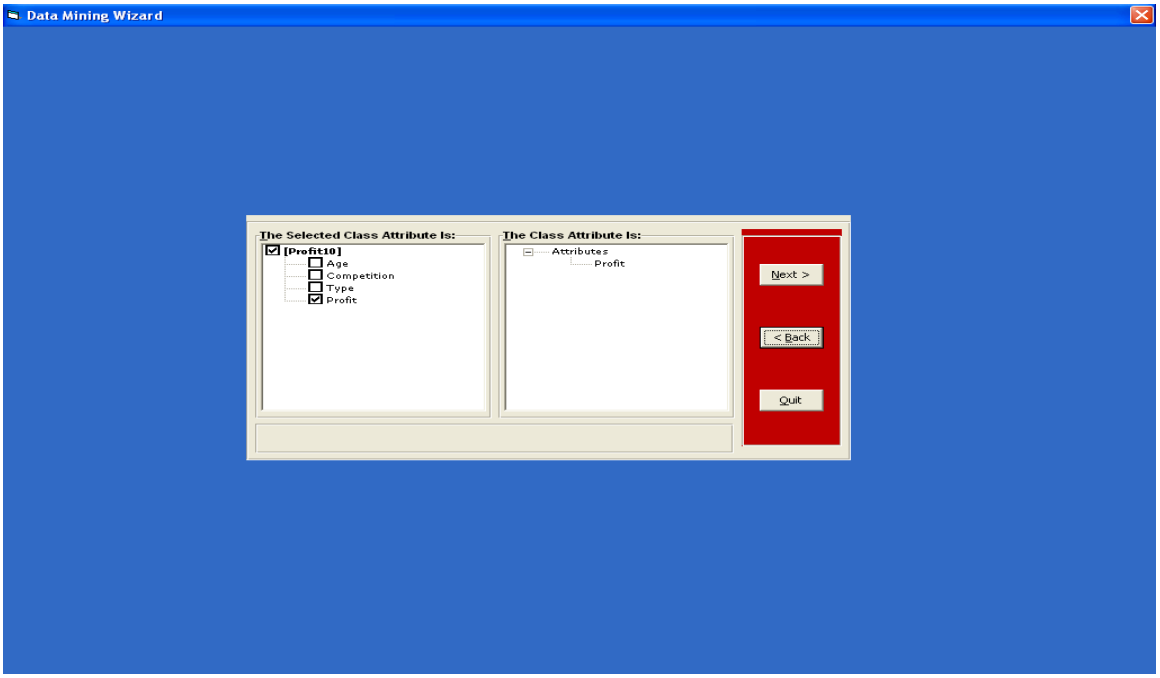
1. The system starts by allowing the user to select the database as follows:



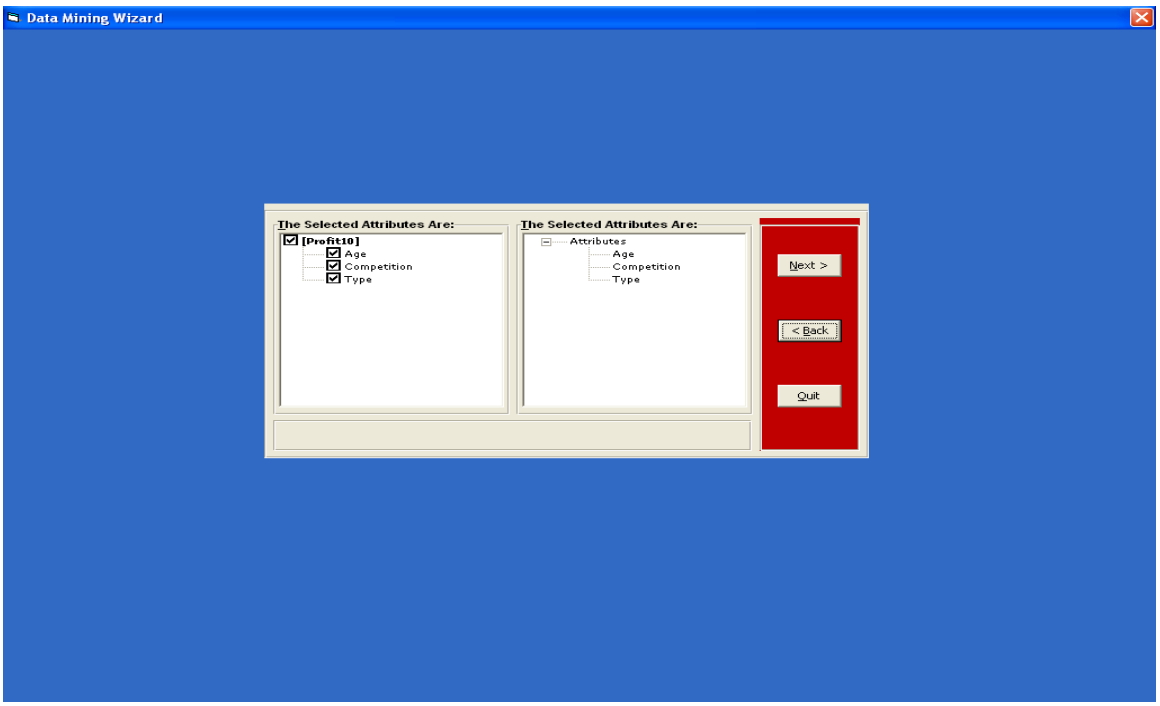
2. The user selects only one table from tables that are saved in the database to deal with it as an input of the system as follows:



3. The system takes a window from the selected data.
4. The system must clean the data from the attributes that have the single values, multi attribute values, and counter attributes.
5. The system considers the last attribute as a class attribute as follows:

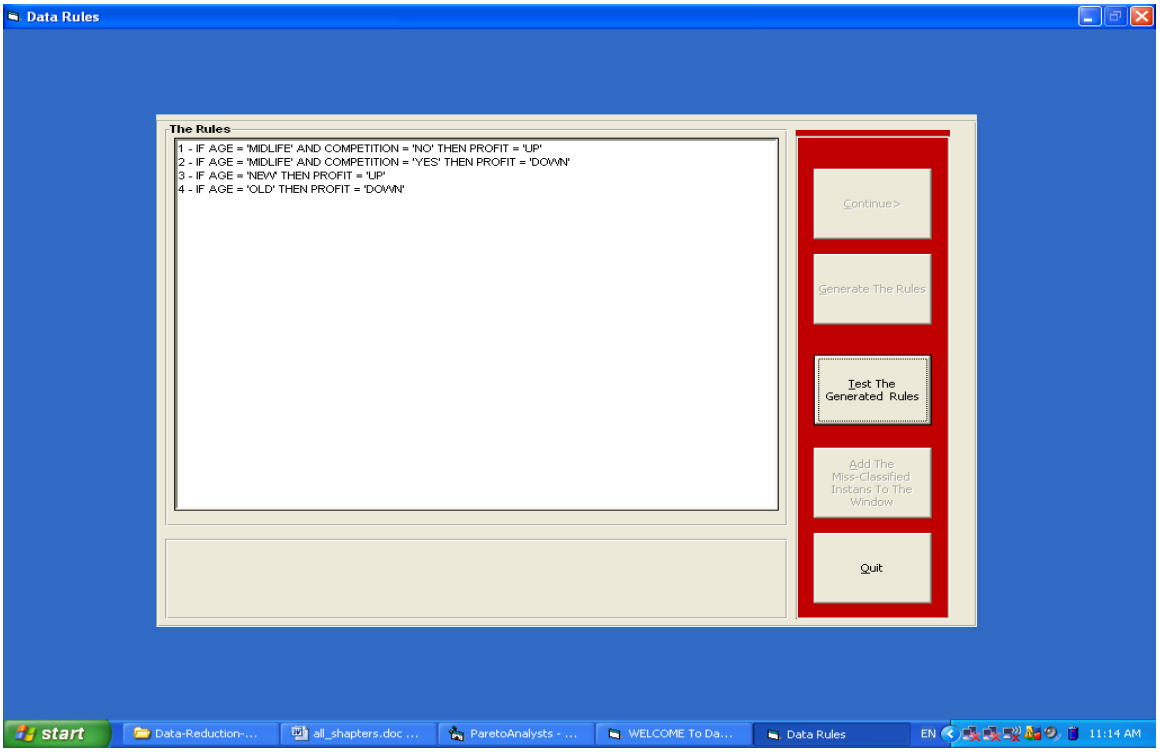


6. The system considers the remnant attributes without removed any of them from the database as follows:

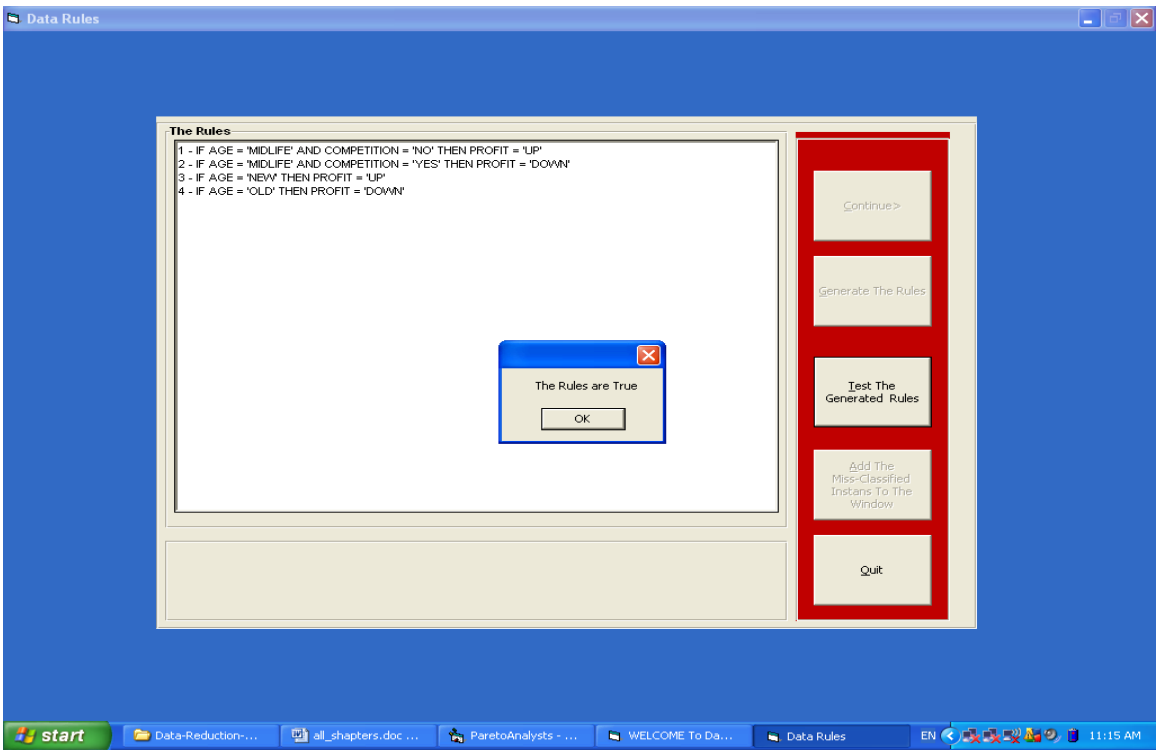


7. The system applies the ID3 algorithm to build the decision tree.

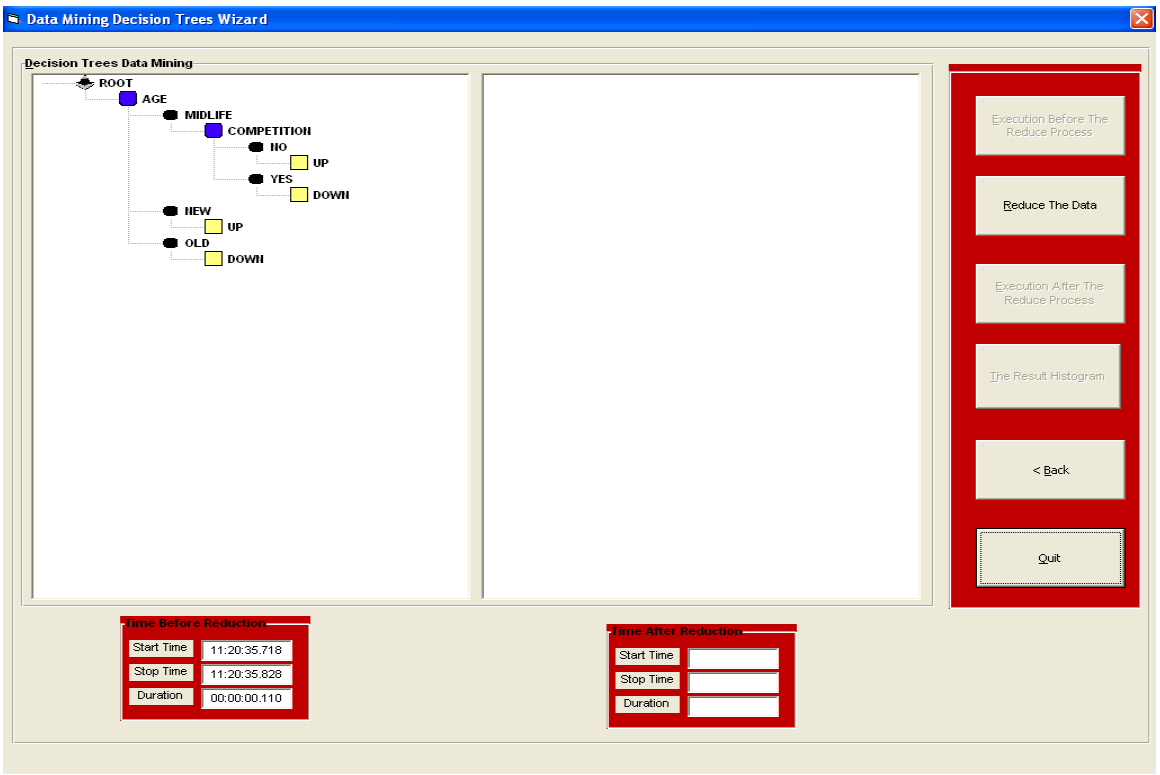
8. The system generates the rules as follows:



9. The system validates the produced rules. If any of the rules is false then the miss classified instances is added to the window and step7 is initiated again as follows:



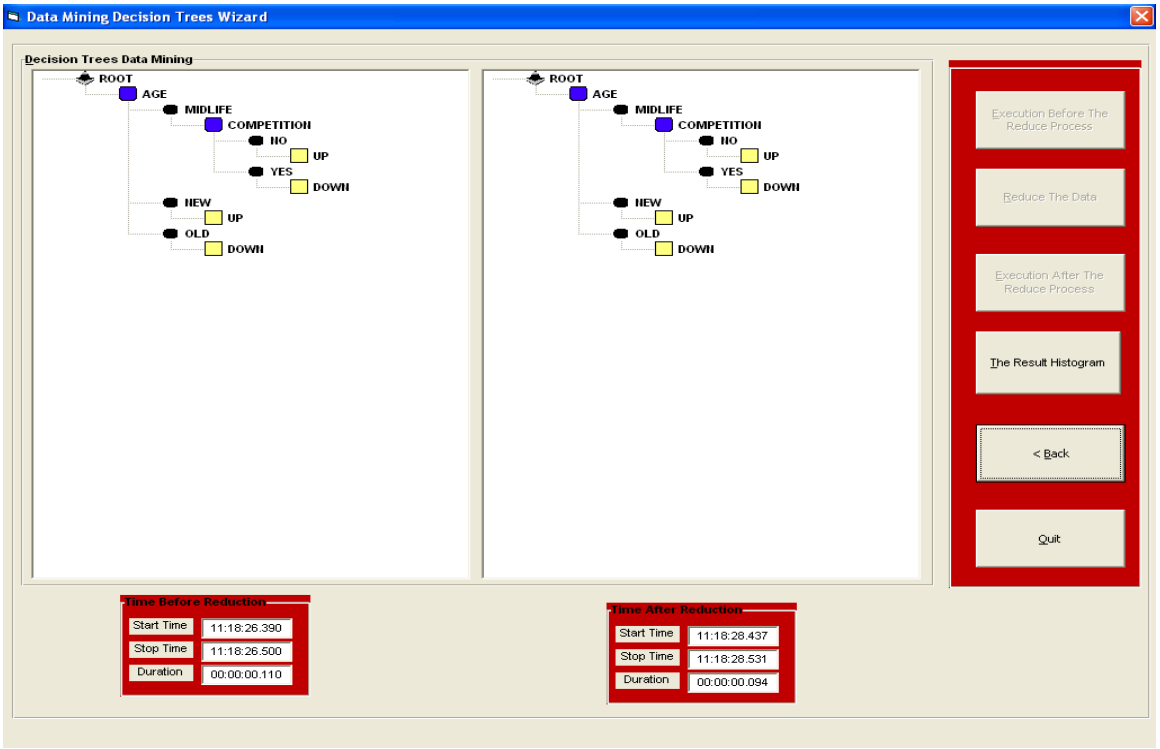
10. The system builds the decision tree after displaying the used attributes and the unused attributes and calculates the runtime (the time spent in applying the ID3 algorithm on the original data and drawing the decision tree process) as follows:



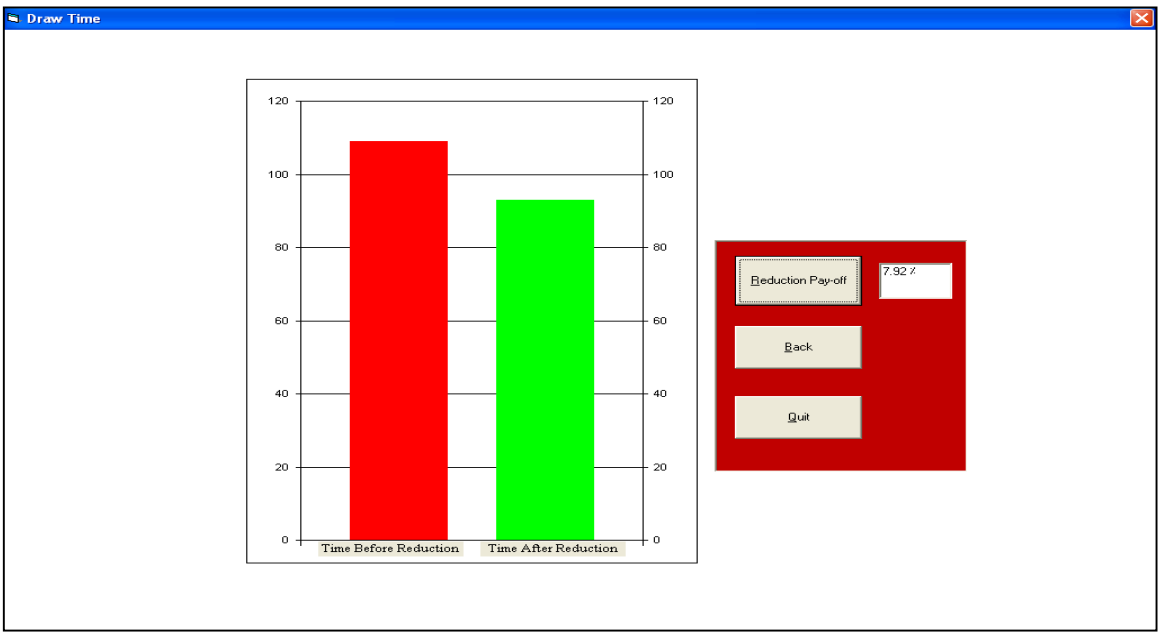
11. The system reduces the data by remove unused attributes from the data.

12. The system repeats the same process (apply the algorithm on the reduced data).

13. The system builds the decision tree and calculates the runtime (the time spent in reducing the data, applying the ID3 algorithm on the reduced data and drawing the decision tree process) as follows:



14. The system calculate the pay-off for the time and draw the histogram for the two times. Where pay-off = ((*t1- t2*)/(*t1 + t2*) * 100) as follows:



Chapter Four

Experiments and Results

In this chapter, we demonstrate the obtained results from our system (data reduction by the use of decision trees) using different sizes of data. The interpretation of the results and comparison will also be presented. Our system performance is tested by number of categorical data sets.

All the experiments are conducted on a Laptop computer with 1.73GHz Intel(R) CPU, 504 MB of RAM memory, 250 MB Hard disk and is running on Microsoft Windows XP professional operating system.

As it has been mentioned in the previous chapter, our system consists of two sub-systems. The first is for building the decision tree before the reduction process takes place and the second is to build the decision tree after the reduction for each data set used to test our system. Our system is tested with a total of nine experiments, each of which consist of 5 different versions depending on the data size (very small, small, medium, large, very large).The comparison of the two sub systems is based on the criterion of the run time.

In accordance with [7], we take a sample from each data set when we conduct our experiments. The sample sizes are taken by the ratio of 382 to 1000. For this reason we divided the data as depicted in table-4.1.

Table-4.1: Data set range size.

Sample number	Data set range size		Experiment data sets size
	From	To	
1	1	38	all
2	39	76	1/2, all
3	77	114	1/3, 2/3, all
4	115	152	1/4, 1/2, 3/4, all
5	153	more	1/4, 1/2, 3/4, all

Here we demonstrate the obtained results from performing our system on different data sets via different versions of the same data taken in account the data samples.

4.1 First experiment

In this experiment, we use the animal data set [19], which consists of 6 objects and 4 attributes in addition to the class attribute. The attributes are; warm-blooded, feathers, fur and swims. The class attribute values are being yes and no for laying eggs. The size of the original data set is 112KB. For the sake of examining our system with large data sets, we have increased this data set randomly to contain 276 instances with size of 116KB. The final results of this experiment are depicted in table-4.2 where the reduction payoff is highlighted with bold.

Table-4.2: Final results of the first experiment.

Experiment version	Number of objects	Sample size	Time before reduction(*t1*)	Time after reduction(*t2*)	Reduction payoff
1	6	all	63	47	**14.55%**
2	54	all	63	47	**14.55%**
		1/2	63	63	0%
3	108	all	63	63	0%
		1/3	62	62	0%
		2/3	62	63	-0.8%
4	126	all	78	62	**11.43%**
		1/4	78	62	**11.43%**
		1/2	78	63	10.64%
		3/4	78	63	10.64%
5	276	all	125	109	6.84%
		1/4	125	94	**14.16%**
		1/2	140	110	12%
		3/4	125	109	6.84%

From table-4.2, we can make the following comments on the results:

1. For the first version of the experiment, there was 14.55% payoff even though the sample size was very small. Figure-4.1, depicts a snap shot of the result from our system and figure-4.2 depicts a histogram of the result of this version of the experiment.
2. For the second version of the experiment, the best payoff was 14.55% when we used the entire data sample that consists of 54 instances. Figure-4.3, depicts a snap shot of the result from our system and figure-4.4 depicts a histogram of the result of this version of the experiment.
3. For the third version of the experiment, there was no payoff in fact there was a drawback of 0.8% in time.

4. For the fourth version of the experiment, the best payoff was 11.43% when we used the entire data and 1/4 sample of data. Figure-4.5, depicts a snap shot of the result from our system and figure-4.6 depicts a histogram of the result of this version of the experiment.
5. For the fifth version of the experiment, there was 14.16% payoff even though the sample size was 1/4 of the data size. Figure-4.7, depicts a snap shot of the result from our system and figure-4.8 depicts a histogram of the result of this version of the experiment.

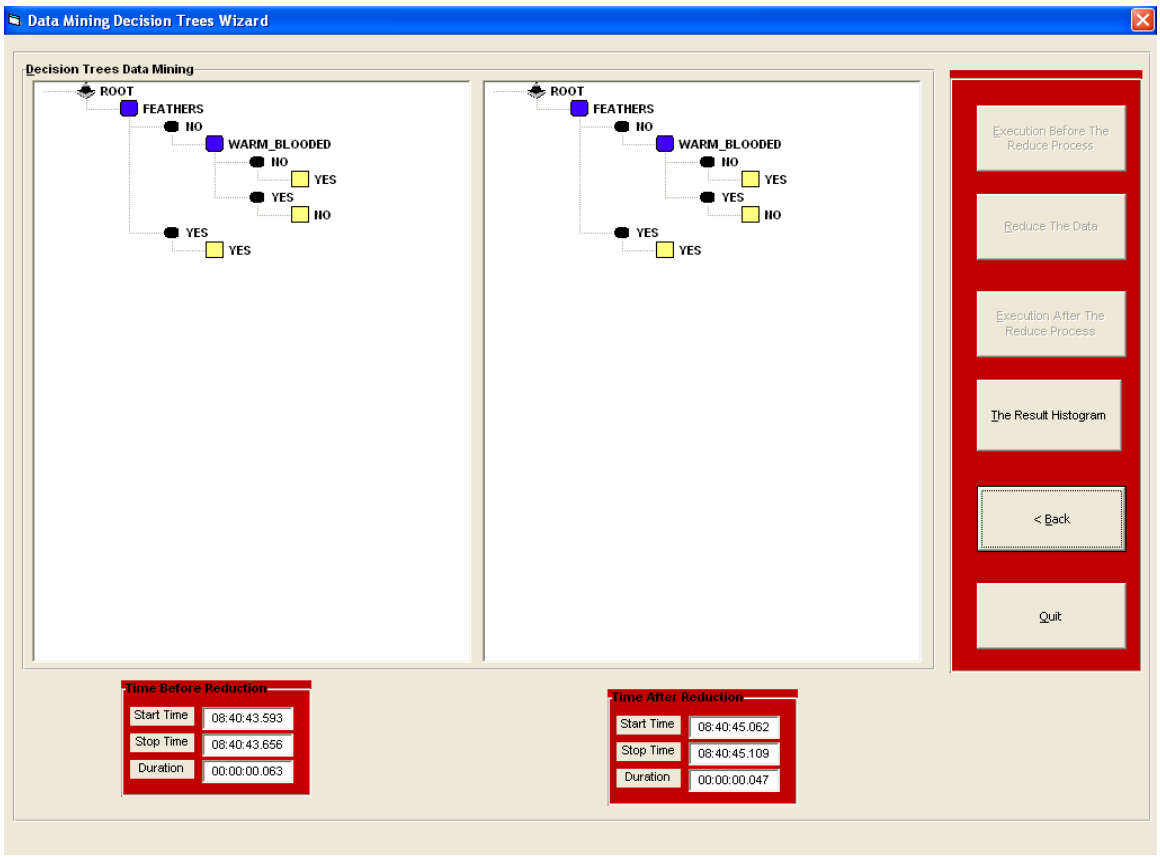


Figure-4.1: System snap shot of the first version of the first experiment.

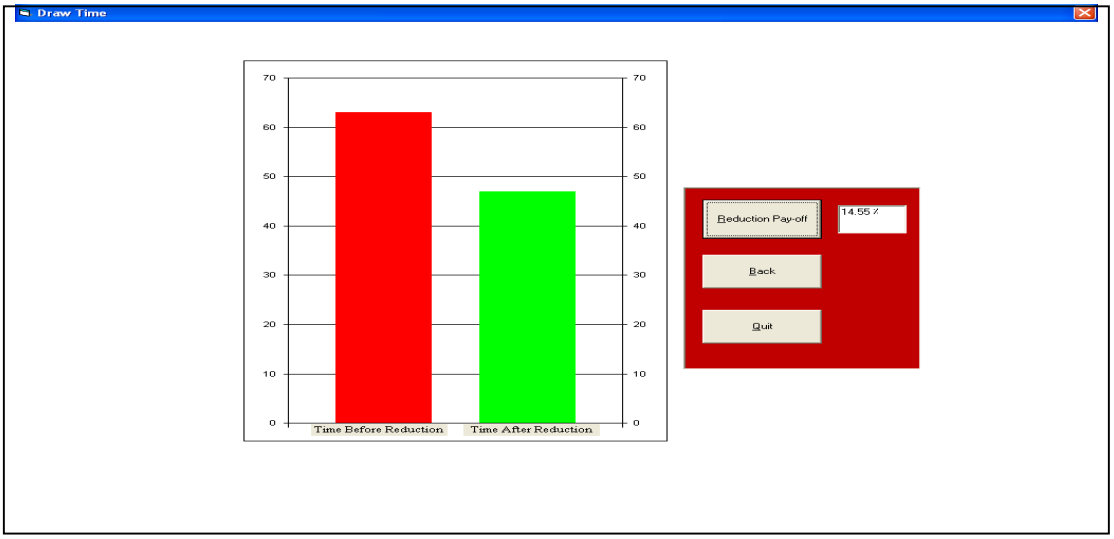


Figure-4.2: Results histogram of the first version of the first experiment.

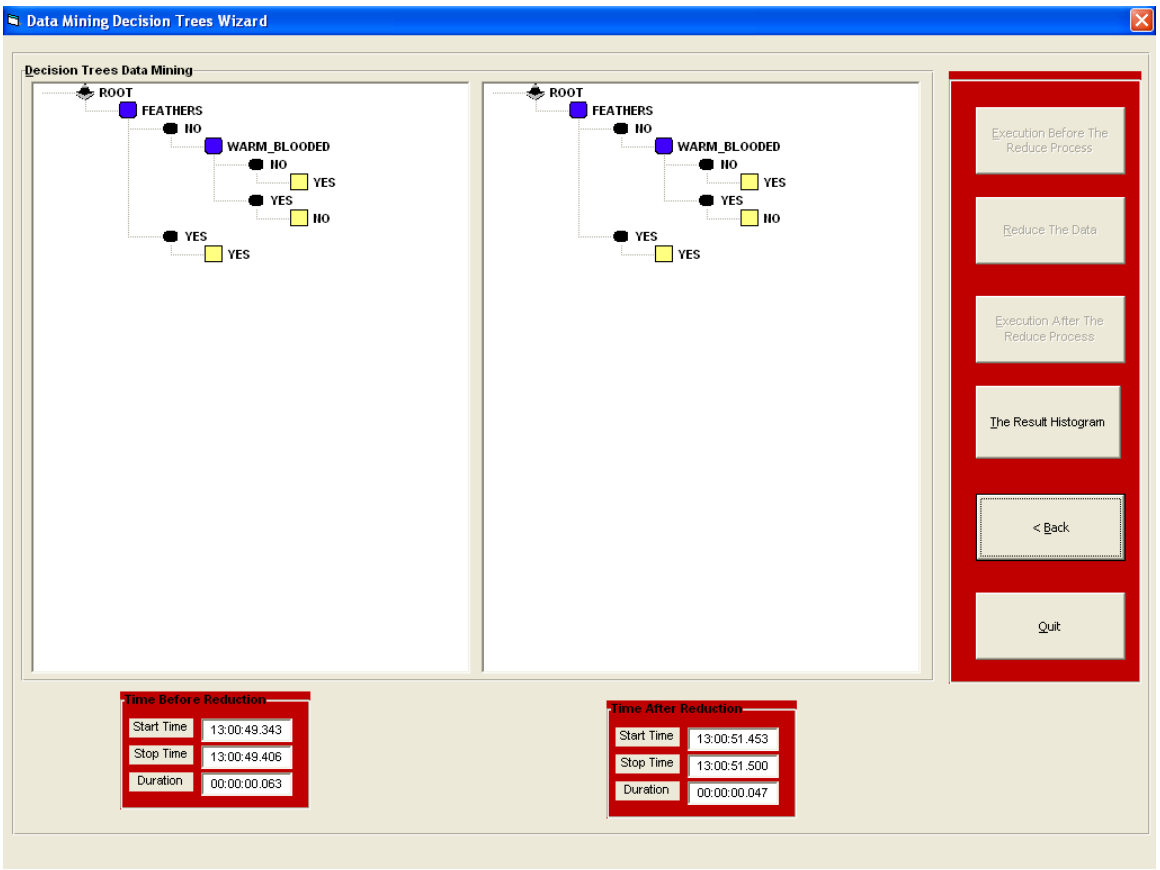


Figure-4.3: System snap shot of the second version of the first experiment.

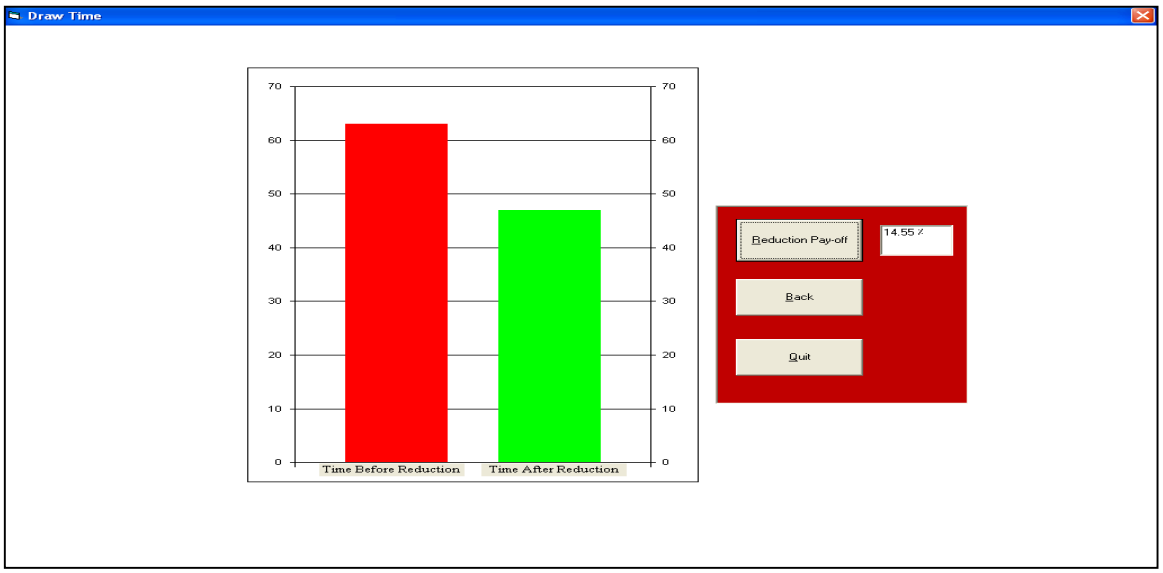


Figure-4.4: Results histogram of the second version of the first experiment.

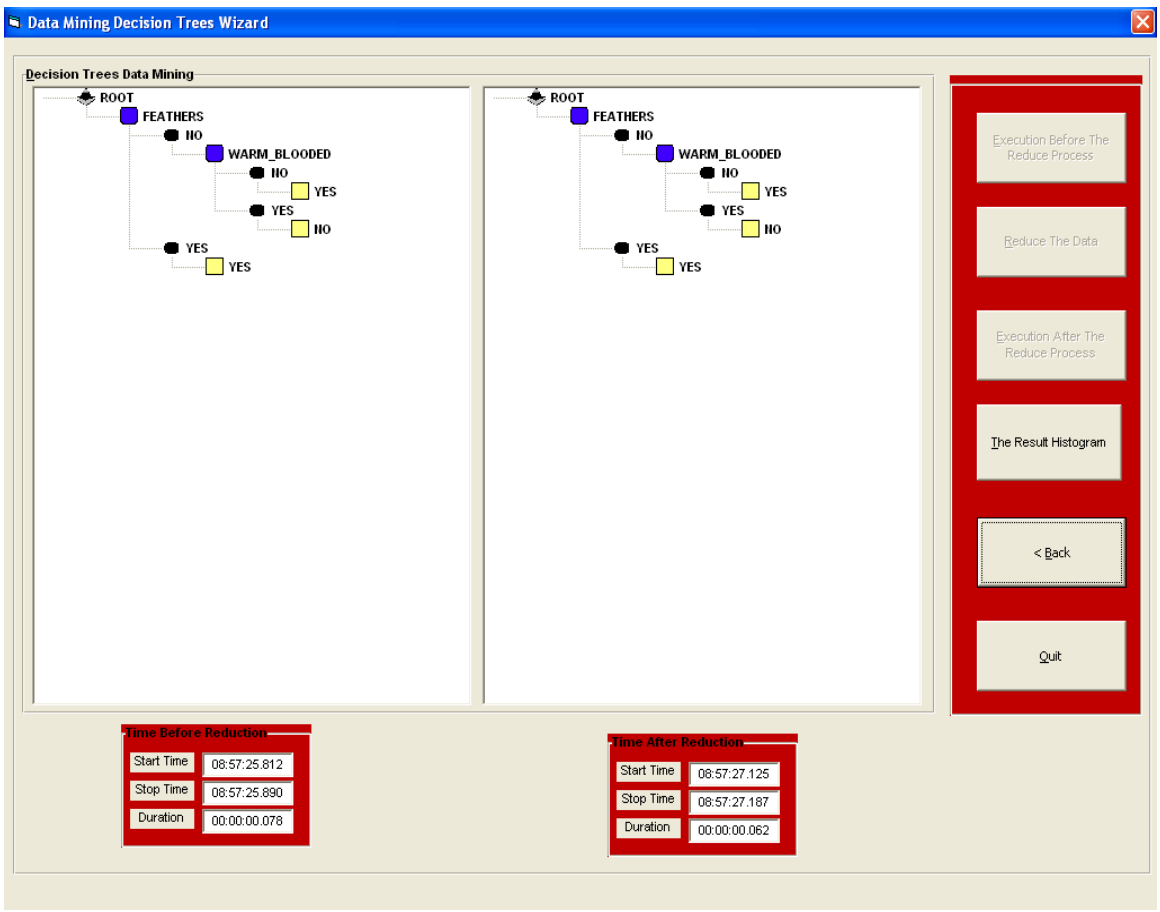


Figure-4.5: System snap shot of the fourth version of the first experiment.

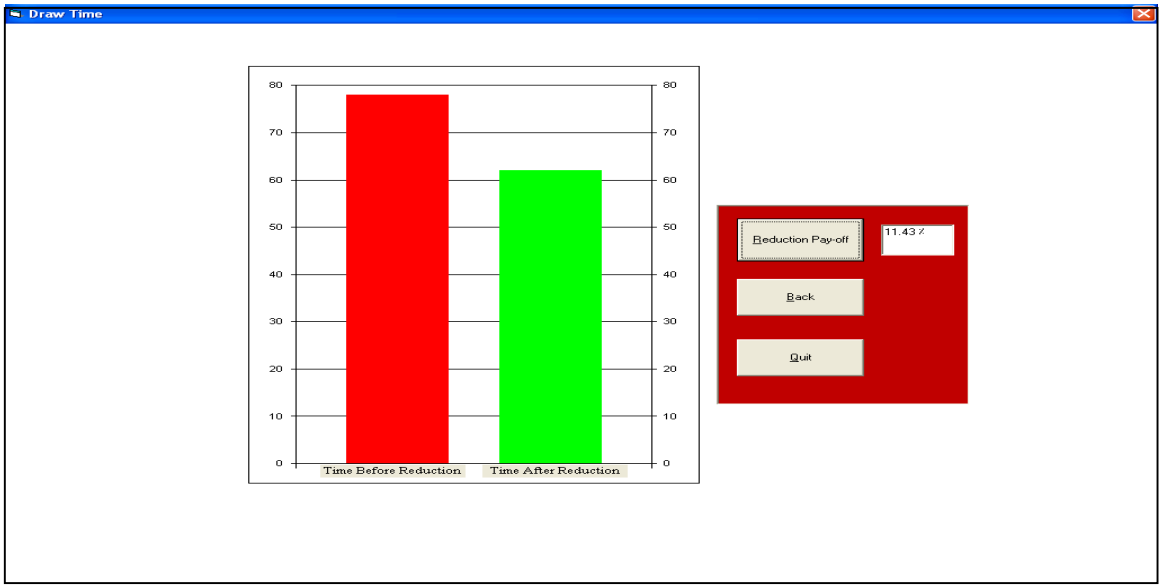


Figure-4.6: Results histogram of the fourth version of the first experiment.

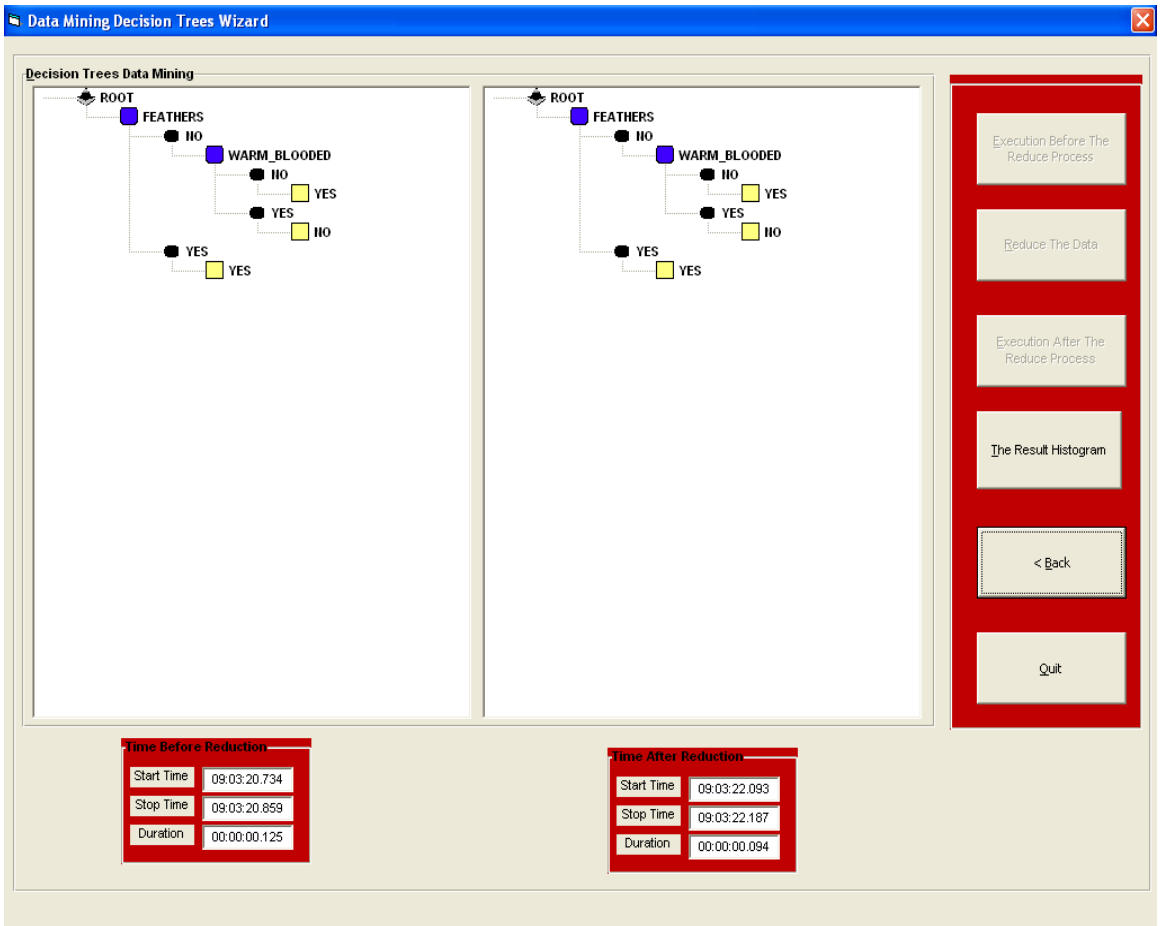


Figure-4.7: System snap shot of the fifth version of the first experiment.

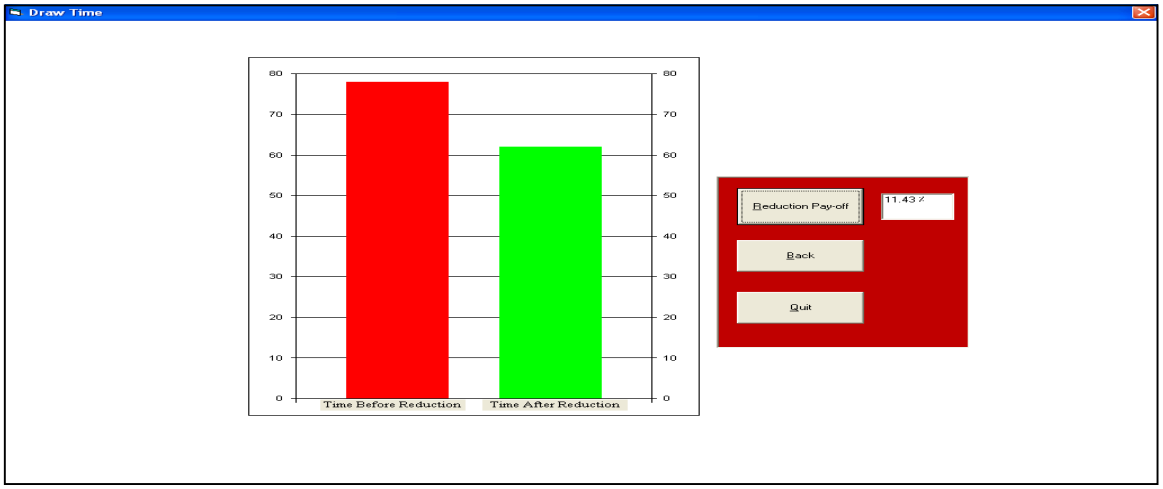


Figure-4.8: Results histogram of the fifth version of the first experiment.

4.2 Second experiment

In the second experiment, we use the buys computer data set [17], which consists of 14 objects and 4 attributes which are (age, income, student and credit rating) and the class attribute (+/-) and the size of the original data set is 108KB. For the sake to examine our system, we increased ,randomly, this data set to contain 322 instances. The final results of this experiment are depicted table4-3.

Table-4.3: final results of the second experiment.

Experiment version	Number of objects	Sample size	Time before reduction(*t1*)	Time after reduction(*t2*)	Reduction payoff
1	14	all	109	94	**7.39%**
2	57	all	141	125	**6.02%**
		1/2	140	125	5.66%
3	112	all	156	157	-0.32%
		1/3	172	157	4.56%
		2/3	172	141	**9.9%**
4	140	all	172	156	4.88%
		1/4	172	140	**10.26%**
		1/2	188	172	4.44%
		3/4	172	156	4.88%
5	322	all	297	250	8.59%
		1/4	281	219	**12.4%**
		1/2	281	250	5.84%
		3/4	281	250	5.84%

From table-4.3, we can make the following comments on the results:

1. The first version of the experiment resulted in 7.39% payoff despite the fact that the sample size was very small.
2. The second version of the experiment concluded that the payoff amounted to 6.02% when the entire data sample that consists of 57 instances was used.
3. The third version of the experiment presented that the best payoff reached 9.9% when 2/3s of sample of data was used.
4. The fourth version of the experiment, the best payoff was 10.26% when 1/4 sample of data was used.
5. The fifth version of the experiment, there was 12.4% payoff despite the fact that the sample size was 1/4 of the data size.

4.3 Third experiment

In the third experiment, we use the London stock market data set [16], which consists of 6 objects and 5 attributes which are (It_rose_yesterday, New_York_rises_today, Bank_rate_high, Unemployment_high and England_is_losing) and the class attribute (No(The London market will not rise today)/ Yes(The London market will rise today)) and the size of the original data set is 116KB. For the sake to examine our system, we increased ,randomly, this data set to contain 1590 instances. The final results of this experiment are depicted table-4.4.

Table-4.4: final results of the third experiment.

Experiment version	Number of objects	Sample size	Time before reduction(*t1*)	Time after reduction(*t2*)	Reduction payoff
1	30	all	141	125	**6.02%**
2	60	all	157	125	**11.35%**
		1/2	157	177	-5.99%
3	107	all	172	156	4.88%
		1/3	172	156	4.88%
		2/3	172	78	**37.6%**
4	130	all	188	156	9.3%
		1/4	187	78	**41.13%**
		1/2	187	156	9.04%

Experiment version	Number of objects	Sample size	Time before reduction(*t1*)	Time after reduction(*t2*)	Reduction payoff
		3/4	203	156	13.09%
5	1590	all	1.032	703	**18.96%**
		1/4	1.031	703	18.92%
		1/2	1.032	719	17.88%
		3/4	1.031	718	17.9%

From table-4.4, we can make the following comments on the results:

1. For the first version of the experiment, there was 6.02% payoff although the sample size was very small.
2. For the second version of the experiment, the best payoff was 11.35% when the entire data was used from the sample that consists of 60 instances.
3. For the third version of the experiment, the best payoff was 37.6% when 2/3s of the data was used.
4. For the fourth version of the experiment, the best payoff was 41.13% when a 1/4 of the sample of data was used.
5. For the fifth version of the experiment, there was 18.96% payoff though the sample size included the entire data sample.

4.4 Fourth experiment

In the fourth experiment, we use the stock market data set [14], which consists of 10 objects and 3 attributes which are (Age, Competition and Type) and the class attribute (Down/Up) and the size of the original data set is 120KB. For the sake to examine our system, we increased ,randomly, this data set to contain 708 instances. The final results of this experiment are depicted table-4.5.

Table-4.5: final results of the fourth experiment.

Experiment version	Number of objects	Sample size	Time before reduction(*t1*)	Time after reduction(*t2*)	Reduction payoff
1	10	all	47	47	0%
2	40	all	63	47	**14.55%**
		1/2	63	47	**14.55%**
3	78	all	78	62	**11.43%**
		1/3	63	63	0%
		2/3	78	62	**11.43%**

Experiment version	Number of objects	Sample size	Time before reduction(*t1*)	Time after reduction(*t2*)	Reduction payoff
4	118	all	78	78	0%
		1/4	94	47	**33.33%**
		1/2	94	78	9.3%
		3/4	94	62	20.51%
5	708	all	313	250	11.19%
		1/4	296	156	**30.97%**
		1/2	296	250	8.59%
		3/4	312	235	14.08%

From table-4.5, we can make the following comments on the results:

1. For the first version of the experiment, no payoff resulted when the size of data was very small size.
2. For the second version of the experiment, the best payoff was 14.55% when the entire and half of the data sample that consists of 40 instances was used.
3. For the third version of the experiment, the best payoff was 11.43% when the whole and 2/3s of data sample was used.
4. For the fourth version of the experiment, the best payoff was 33.33% when a 1/4 of the data sample was used.
5. For the fifth version of the experiment, there was 30.97% payoff when a 1/4 of the data sample was used.

4.5 Fifth experiment

In the fifth experiment, we use the club member ship data set [18], which consists of 12 objects and 4 attributes which are (Eyecolour, Married, Sex and Hairlength) and the class attribute (Rugby/Netball) and the size of the original data set is 114KB. For the sake to examine our system, we increased ,randomly, this data set to contain 1920 instances. The final results of this experiment are depicted table-4.6.

Table-4.6: final results of the fifth experiment.

Experiment version	Number of objects	Sample size	Time before reduction(*t1*)	Time after reduction(*t2*)	Reduction payoff
1	12	all	47	31	**20.51%**
2	48	all	47	16	**49.21%**
		1/2	47	31	20.51%

Experiment version	Number of objects	Sample size	Time before reduction(*t1*)	Time after reduction(*t2*)	Reduction payoff
3	96	all	63	47	14.55%
		1/3	62	47	13.76%
		2/3	63	31	**34.04%**
4	120	all	78	31	**43.12%**
		1/4	78	47	24.8%
		1/2	78	31	**43.12%**
		3/4	78	47	24.8%
5	1920	all	640	297	36.61%
		1/4	641	297	**36.67%**
		1/2	641	297	**36.67%**
		3/4	640	313	34.31%

From table-4.6, we can make the following comments on the results:

1. As for the first version of the experiment, there was 20.15% payoff in spite of the fact that the sample size was very small.
2. As for the second version of the experiment, the best payoff was 49.21% when the entire data sample that consists of 48 instances was used.
3. As for the third version of the experiment, the best payoff was 34.04% when 2/3s of data sample that consists of 96 instances was used.
4. As for the fourth version of the experiment, the best payoff was 43.12% when all and a 1/2 of data sample was used.
5. As for the fifth version of the experiment, there was 36.67% payoff even though a 1/4 and a1/2 of data sample was used.

4.6 Sixth experiment

In the sixth experiment, we use the fruit data set [20], which consists of 16 objects and 4 attributes which are (Skin, Colour, Size and Flesh) and the class attribute (safe/dangerous to eat) and the size of the original data set is 116KB. For the sake to examine our system, we increased ,randomly, this data set to contain 2560 instances. The final results of this experiment are depicted table-4.7.

Table-4.7: final results of the sixth experiment.

Experiment version	Number of objects	Sample size	Time before reduction(*t1*)	Time after reduction(*t2*)	Reduction payoff

1	16	all	156	141	**5.05%**
2	48	all	172	172	0%
		1/2	172	157	**4.56%**
3	96	all	203	187	**4.1%**
		1/3	203	187	**4.1%**
		2/3	203	188	3.84%
4	144	all	219	203	3.79%
		1/4	235	203	**7.31%**
		1/2	218	203	3.56%
		3/4	219	203	3.79%
5	2560	all	1.594	1.328	**9.1%**
		1/4	1.593	1.344	8.48%
		1/2	1.594	1.344	8.51%
		3/4	1.593	1.344	8.48%

From table-4.7, we can make the following comments on the results:

1. For the first version of the experiment, there was 5.05% payoff even though the sample size was very small.
2. For the second version of the experiment, the best payoff was 4.56% when we used the half of data sample.
3. For the third version of the experiment, the best payoff was 4.1% when we used the entire and 1/3 of data sample.
4. For the fourth version of the experiment, the best payoff was 7.31% when we used 1/4 of data sample.
5. For the fifth version of the experiment, there was 9.1% payoff even though the sample size was entire of data sample.

4.7 Seventh experiment

In the seventh experiment, we use the shape data set [13], which consists of 14 objects and 4 attributes which are (ID1, Color, Outline and Dot) and the class attribute (triangle/ square) and the size of the original data set is 156KB. For the sake to examine our system, we increased ,randomly, this data set to contain 560 instances. The final results of this experiment are depicted table-4.8.

Table-4.8: final results of the seventh experiment.

Experiment version	Number of objects	Sample size	Time before reduction(*t1*)	Time after reduction(*t2*)	Reduction payoff
1	14	all	94	109	-7.39%
2	56	all	109	110	-0.46%
		1/2	109	110	-0.46%
3	112	all	125	125	0%
		1/3	140	179	-12.23%
		2/3	141	141	0%
4	140	all	140	125	**5.66%**
		1/4	141	141	0%
		1/2	141	140	0.36%
		3/4	141	140	0.36%
5	560	all	281	281	0%
		1/4	297	250	**8.59%**
		1/2	297	282	2.59%
		3/4	296	281	2.6%

From table-4.8, we can make the following comments on the results:

1. The first version of the experiment did not result in a payoff; in fact, there was no payoff there was a drawback of 7.39% in time.
2. The second version of the experiment, there was no payoff, there was a drawback of 0.46% in time.
3. The third version of the experiment, did not result in a payoff, there was no payoff in fact there was a drawback of 12.23% in time.
4. The fourth version of the experiment, the best payoff was 5.66% when entire of the data sample was used.
5. The fifth version of the experiment, there was 8.59% payoff despite the fact that a 1/4 of the data sample.

4.8 Eighth experiment

In the eighth experiment, we use the university students data set [18], which consists of 26 objects and 5 attributes which are (SoftEng, ARIN, HCI, CSA and Project) and the class attribute (UPPER/ FIRST) and the size of the original data set is 140KB. For the sake to examine our system, we increased ,randomly, this data set to contain 780 instances. The final results of this experiment are depicted table-4.9.

Table-4.9: final results of the eighth experiment.

Experiment version	Number of objects	Sample size	Time before reduction(*t1*)	Time after reduction(*t2*)	Reduction payoff
1	26	all	172	172	0%
2	52	all	187	172	**4.18%**
		1/2	187	187	0%
3	104	all	218	219	-0.23%
		1/3	218	219	-0.23%
		2/3	235	203	**7.31%**
4	130	all	328	297	4.96%
		1/4	328	218	**20.15%**
		1/2	328	351	-3.39%
		3/4	312	282	5.05%
5	780	all	797	719	5.15%
		1/4	797	593	**14.68%**
		1/2	781	687	6.4%
		3/4	828	734	6.02%

From table-4.9, we can make the following comments on the results:

1. For the first version of the experiment, there was no payoff.
2. For the second d version of the experiment, the best payoff was 4.18% when we used the entire of data sample.
3. For the third version of the experiment, the best payoff was 7.31% when we used 2/3 of data sample.
4. For the fourth version of the experiment, the best payoff was 20.15% when we used 1/4 of data sample.
5. For the fifth version of the experiment, there was 14.68% payoff even though 1/4 of data sample.

4.9 Ninth experiment

In the eighth experiment, we use the play tennis data set [27], which consists of 14 objects and 4 attributes which are (outlook, temperature, humidity and windy) and the class attribute (yes/no) and the size of the original data set is 119KB. For the sake to examine our system, we increased ,randomly, this data set to contain 1120 instances. The final results of this experiment are depicted table-4.10.

Table-4.10: final results of the ninth experiment.

Experiment version	Number of objects	Sample size	Time before reduction(*t1*)	Time after reduction(*t2*)	Reduction payoff
1	28	all	125	125	0%
2	56	all	125	125	0%
		1/2	140	109	**12.45%**
3	112	all	156	156	0%
		1/3	172	141	**9.9%**
		2/3	172	156	4.88%
4	140	all	187	156	9.04%
		1/4	188	141	**14.29%**
		1/2	187	156	9.04%
		3/4	172	156	4.88%
5	1120	all	610	532	6.83%
		1/4	594	422	**16.93%**
		1/2	610	516	8.35%
		3/4	656	562	7.72%

From table-4.10, we can make the following comments on the results:

1. For the first version of the experiment, there was no payoff in fact there was 0% in time.
2. For the second version of the experiment, the best payoff was 12.45% when we used half of data sample.
3. For the third version of the experiment, the best payoff was 9.9% when we used 1/3 of data sample.
4. For the fourth version of the experiment, the best payoff was 14.29% when we used 1/4 of data sample.
5. For the fifth version of the experiment, there was 16.93% payoff even though 1/4 of data sample.

Chapter Five

Conclusion and further works

The objectives of this work was to study the concept of data reduction by the use of decision trees. The decision trees algorithm used for this purpose is the well-known algorithm ID3. A software system is developed to implement the ID3 algorithm and to time the performance of the system before and after the data reduction takes place. Our system is tested by a number of well-known data sets in the form of experiments.

The objectives of this study had been accomplished by implementing our system in VB6 programming language with a user friendly interfaces.

5.1 Conclusion

We have carried out on our system a total of 126 experiments. The data sets sizes use in these experiments ranged form 55.808 bytes to 434.176 bytes. The best results of the 126 experiments are summarized in table5.1.

Table-5.1: Obtained results from experiments

Experiments	Version	Data size	Sample size	Time		Payoff in time
				t1/ms	*t2*/ms	
1	1	6	all	63	47	14.55%
	2	54	all	63	47	14.55%
	4	126	all	78	62	11.43%
			1/4	78	62	11.43%
	5	276	1/4	125	94	14.16%
2	1	14	all	109	94	7.39%
	2	57	all	141	125	6.02%
	3	112	2/3	172	141	9.9%
	4	140	1/4	172	140	10.26%
	5	322	1/4	281	219	12.4%
3	1	30	all	141	125	6.02%
	2	60	all	157	125	11.35%
	3	107	2/3	172	78	37.6%
	4	130	1/4	187	78	41.13%
	5	1590	all	1.032	703	18.96%
4	2	40	all	63	47	14.55%

Experiments	Version	Data size	Sample size	Time		Payoff in time
				t1/ms	*t2*/ms	
			1/2	63	47	14.55%
	3	78	all	78	62	11.43%
			2/3	78	62	11.43%
	4	118	1/4	94	47	33.33%
	5	708	1/4	296	156	30.97%
5	1	12	all	47	31	20.51%
	2	48	all	47	16	49.21%
	3	96	2/3	63	31	34.04%
	4	120	all	78	31	43.12%
			1/2	78	31	43.12%
	5	1920	1/4	641	297	36.67%
			1/2	641	297	36.67%
6	1	16	all	156	141	5.05%
	2	48	1/2	172	157	4.56%
	3	96	all	203	187	4.1%
			1/3	203	187	4.1%
	4	144	1/4	235	203	7.31%
	5	2560	all	1.594	1.328	9.1%
7	4	140	all	140	125	5.66%
	5	560	1/4	297	250	8.59%
8	2	52	all	187	172	4.18%
	3	104	2/3	235	203	7.31%
	4	130	1/4	328	218	20.15%
	5	780	1/4	797	593	14.68%
9	2	56	1/2	140	109	12.45%
	3	112	1/3	172	141	9.90%
	4	140	1/4	188	141	14.29%
	5	1120	1/4	594	422	16.93%

The summary of the results we have obtained from the previously conducted experiments, they are as follows:

The profit percentage at 0.409 when the size of the sample represented 1/4 of the original data, the profit result was 0.318 while the profit's result was 0.114 when the data sample size was 1/2, and 2/3 of the original data. The profit's percentage was 0.045 when the sample size was 1/3 of the original data. The profit's percentage was 0.114 in both 1/2 and 2/3's of the data sample. When the data sample represented 3/4 o the original data there was no profit's percentage at all. Taking what has been stated into consideration, we

notice that the major profit's percentage that was obtained was when data sample representing all of the original data.

5.2 Further works

- Use another algorithms for example C4.5,C5 eal.
- Use another method for choose sample.
- Improvement the program to accept another values for example the numeric values.

References

1. Barbar D. and eal. “The New Jersey Data Reduction Report”. IEEE Data Engineering Bulletin: Special Issue on Data Reduction Techniques, Joseph M. Hellerstein. Vol(20). Number 4 (1997).pages 3-45.
2. Berry M., and Linoff G., ”Data Mining Techniques for marketing, sales, and customer relationship management”. Second Edition. Wiley, Inc., Indianapolis, Indiana. (2004). Pages 643.
3. Borenstein E., Sharon E., and Ullman S., “Combining Top-Down and Bottom-Up Segmentation”. Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 4, , June 27-July 02, (2004). Pages 46.
4. Bronnimann H. and eal. “Efficient Data-Reduction Methods for On-Line Association Rule Discovery”. Chapter 7 of Data Mining: Next Generation Challenges and Future Directions, Hillol Kargupta, Anupam Joshi, Krishnamoorthy Sivakumar, and Yelena Yesha (editors), AAAI/MIT Press,. (2004). pages 125–146.
5. Bukhman L., (M.Sc., Polytechnic University, June 2005). “Approximation of iceberg cubes using data reduction techniques”. M.Sc. Thesis, Department of Computer and Information Science, Polytechnic University, May (2005). Belrus.
6. Chen M., Han J., and Yu P., “Data Mining: An Overview from a Database Perspective”. IEEE Transactions on Knowledge and Data Engineering, Vol.8, No.6, (1996). Pages 866-883.
7. Cochran, W., “Sampling Techniques”, third edition. John Wiley & Sons, Inc., New York, (1977). Pages 428.
8. Corston-Oliver S. and Gamon M., “Combining decision trees and transformation-based learning to correct transferred linguistic representations”. Association for Machine Translation in the Americas, September (2003).
9. Craig, J. and Webb, J., “Microsoft Visual Basic 6.0 Developer’s Workshop. Microsoft Press”, Washington, USA. (1998). Pages 804.

10. Fayyad U., Piatetsky-Shapiro G., and Smyth P. “From Data Mining to Knowledge Discovery in Databases”. In Fayyad U., Piatetsky-Shapiro G., Smyth P. and Uthurusamy R. (eds.), Advances in Knowledge Discovery and Data Mining, AAAI Press, (1995). Pages 37–54.
11. Frawley W., Piatetsky-Shapiro, G., and Matheus, C., “Knowledge Discovery in Databases: An Overview”. AI Magazine, 13(3), (1992). Pages 57–70.
12. Han J., and Kamber M., “Data Mining: Concepts and Techniques”. Morgan Kaufmann publishers, San Francisco, CA. USA Data Mining. 2001. Pages 1-30.
13. http://www.ccs.neu.edu/home/futrelle/teaching/csu520csg120sp2007/Zupan-Bratko%20decision%20trees.pdf, Accessed by 12/5/2010.
14. http://www.cis.temple.edu/%7Eingargio/cis587/readings/id3-c45.html, Accessed by 1/2/2010.
15. http://www.cs.nyu.edu/faculty/davise/ai/id3.pdf, Accessed by 11/2/2010.
16. http://www.ebookee.com/Data-Warehousing-OLAP-and-Data-Mining 340515.html, Accessed by 28/4/2010.
17. http://www.math-usk.org/tfa/dm/dm-classification.pdf, Accessed by 1/8/2010.
18. http://www.maxbramer.org.uk/papers/mining.doc, Accessed by 13/4/2010.
19. http://www.profesores.elo.utfsm.cl/~tarredondo/info/soft-comp/codigo/dt/id-3%20algorithm.pdf, Accessed by 27/4/2010.
20. http://www.soc.napier.ac.uk/~peter/vldb/dm/node11.htm, Accessed by 15/9/2010.
21. http://www.usyd.edu.au/su/agric/acpa/fkme, Accessed by 21/9/2010.
22. Larose, D., “Discovering Knowledge in Data: an Introduction to data Mining”. JOHN WILEY & SONS, Hoboken, New Jersey. December (2005). Pages 1-25.
23. Mitra S. and Acharya T., “Data Mining Multimidia, Soft Computing and Bioinformatics”. Wiley-Interscience, Hoboken, New Jersey, Adaptive computation and machine learning. Published simultaneously in Canada. (2003). Pages 1-26.
24. Murthy S., “Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey”. Data Mining and Knowledge Discovery, v.2 n.4. December 1998. Pages 345-389.

25. Park, J., Chen, M., and Yu, P., “An effective hash-based algorithm for mining association rules”. In Proc. ACM-SIGMOD Int. Conf. Management of Data (SIGMOD’95), San Jose, CA. (1995). Pages 175–186.
26. Pjesivac-Grbovic J., and eal. “Decision Trees and MPI Collective Algorithm Selection Problem”. Euro-Par (2007). pages 107-117.
27. Quinlan, J.R., “Induction on decision trees”. Machine Learning, vol. 1, (1986). Pages 81-106.
28. Smith E., Whisler V., and Marquis H., “Visual Basic 6 Bible”. IDC Books Worldwide, Inc. An International Data Group Computer. (1998). Pages 1021.
29. Tylee, L., “Learn Visual Basic 6.0”. (1998). Pages 471.
30. Ye N., “The Handbook of Data Mining. Lawrence Erlbaum Associates”. Inc., Mahwah, New Jersey. (2003). Pages 689.

الملخص

نظرا لوجود كميات هائلة من البيانات المخزنة في قواعد البيانات (Databases) ومستودعات البيانات (Data warehouses) الضخمة فقد تزايدت الحاجة إلى تطوير و استحداث ادوات جديدة تمتاز بالقوة لتحليل البيانات واستخراج المعلومات والمعرفة منها. ومن هدا الاحتياج ظهر مجال جديد يسمى بالتنقيب في البيانات (Data Mining (DM)) كتقنية تهدف إلى استخراج المعرفة من كميات هائلة من البيانات.

يعتبر تخفيض او تقليص (Reduction) حجم البيانات من اهم الخطوات العملية في المعالجة الاولية للبيانات في نظم التنقيب في البيانات. حيث إن عملية تقليص البيانات هي اختيار مجموعة من البيانات تكون اصغر في الحجم بشرط الإبقاء على خصائص وسلامة و شمولية البيانات الاصلية.

هذا العمل يختص بدراسة طريقة لتقليص البيانات باستخدام شجرات القرار (Decision Trees) وإنتاج منظومة حاسوب لبرمجة إحدى خوارزميات شجرات القرار ID3 في تقليص البيانات. الهدف من إنتاج المنظومة لإثبات ان هناك فائدة كبيرة لتقليص البيانات مع الحصول على نفس النتائج باستعمال جميع البيانات.

إنّ هدف هده الدراسةِ هو تطبيق نظام حاسوب يستعمل اشجار القرارِ في عملية تقليص حجم بياناتِ. ودراسة تاثير استعمال مفهومِ اشجارِ القرارِ في مجال تقليص حجم البياناتِ. هدا العملِ سيقارنَ وقتا المعالجة في بناء نفس شجرة القرار مرة عند استخدام البياناتِ الاصليةِ و مرة اخرى عند استخدام البياناتِ المقلصة. حيث ان إجمالي الوقت في المرة الاولى يتضمّنَ الوقت المستغرق في تطبيق خوارزمية ID3 بالإضافة إلى الوقت المستغرق في رسم شجرةِ القرار, اما إجمالي الوقت في المرة الثانية يتضمن الوقت المستغرق في عمليةِ التخفيضِ والوقت المستغرق في تطبيق خوارزمية ID3 بالإضافة إلى الوقت المستغرق في رسم شجرة القرار.

طبق هذا العمل من خلال 9 تجارب رئيسية لبيانات مختلفة وباحجام مختلة, وتمت برمجة المنظومة بلغة الفيجوال بيسك 6.0. واخيرا سيتم عرض النتائج التي تم الحصول عليها من اختبار هدا النظام مع تحليل مفصل للنتائج.

جامعه قاريونس

كليه تقنيه المعلومات

قسم علوم الحاسوب

دراسه مقدمه لغرض استكمال متطلبات الحصول درجه الإجازة العاليه

الماجستير في علوم الحاسوب بعنوان:

تقليص البيانات باستخدام شجرات القرار

إعداد الطالب:

سالمه إبراهيم المبروك شليتيت

إشراف الدكتور

فرج عبدا لقادر المؤدب

2010