COMPARISON OF DATA MINING CLASSIFICATION ALGORITHMS: C5.0 AND CART FOR CAR EVALUATION AND CREDIT CARD INFORMATION DATASETS

EI THINZAR WIN MAUNG

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

COMPARISON OF DATA MINING CLASSIFICATION ALGORITHMS: C5.0 AND CART FOR CAR EVALUATION AND CREDIT CARD INFORMATION DATASETS

By

EI THINZAR WIN MAUNG

B.C.Sc. (Hons:)

A Dissertation Submitted in Partial Fulfilment of the Requirements for the Degree of

Master of Computer Science

(M.C.Sc.)

UNIVERSITY OF COMPUTER STUDIES, YANGON

JANUARY, 2020

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and appreciation to all persons who contributed directly or indirectly towards the success of this thesis.

First, I would like to show my respect to **Dr. Mie Mie Thet Thwin**, Rector of the University of Computer Studies, Yangon, for her kind permission to prepare this thesis and for valuable suggestion and general guidance during the period of study.

I continue to be indebted to course coordinator, **Dr. Thi Thi Soe Nyunt**, Professor and Head of Faculty of Computer Science, University of Computer Studies, Yangon, for providing administrative support and sympathetic suggestion during the period of the development of this thesis.

Especially, I would like to express my special appreciation and sincere thanks to my supervisor, **Dr. Zin May Aye**, Professor of the Cisco Network Lab, University of Computer Studies, Yangon, who gave me encouragement, kind supervision on the accomplishment of my research work. I am highly thankful to her for helpful guidance, patient supports, remarkably advices and comments throughout this process.

I would like to give my respect to **Daw Aye Aye Khine**, Associate Professor and Head of the Department of Language, University of Computer Studies, Yangon, for editing my thesis from language point of view.

I am also grateful to all teachers from the University of Computer Studies, Yangon, who taught and guided me during the period of academic years.

Finally, I am greatly thankful to my parents for their continuously and practically encouragement, support and kindness through my life. I also thank my friends who willingly helped and cooperated with me throughout the preparation of this thesis.

STATEMENT OF ORIGINALITY

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

---------- ----------

Date Ei Thinzar Win Maung

ABSTRACT

Big data and its analysis have become a widespread practice in recent times, applicable to multiple industries. Data mining is a technique that is based on statistical applications. It is the process of discovering hidden or unknown patterns in huge datasets that are potentially useful and ultimately understandable. The goal of data mining is to extract useful information from huge data sets and to store it as an understandable and structured model for future use, using combined technique of statistics, machine learning and database systems. Classification is a supervised method, which is used to predict categorical class label of a given data instance so as to classify it into a predetermined class. Decision tree is the simple and most commonly used algorithm among the classification algorithms. This system analyses the performance of CART and C5.0 algorithms based on training and testing phases for two UCI datasets: car evaluation and credit card datasets using evaluation metrics such as accuracy, processing time and decision rules. It is a two-step process, in the first step, algorithm uses training data to build a classifier, and then in second step it uses this classifier to estimate the class label of data instance. The classifier is like a function that maps a data instance to a label. The system aims to compare the results of both algorithms and discovers in which either one of them is significantly outperforming the other.

This system implemented using C# programing language with Microsoft Visual Studio 2013 and Microsoft SQL Server Management Studio 2012 platform to build the database.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

In recent years, there has been an enormous amount of data being produced and stored in different places around the world. Users are provided with many tools to find the repositories with small size of data in organizations and research fields. Topic and subject browsing, keyword searching and other techniques can help users to mine important pieces of information quickly. Index search mechanisms allow the user to retrieve a set of relevant documents. However, these search mechanisms are sometimes not sufficient. The amount of available data is increasing rapidly. Without automatic extraction methods, it is very difficult for humans to extract the necessary information. Gaining new knowledge, retrieving the meaning of text documents and associates it to other knowledge become a major challenge.

Data mining refers to the process of extracting or mining knowledge from large amounts of data. It is the process of searching available patterns by scanning the huge amount of data. Storing enormous quantity of data is utile to extract precious knowledge. To seek out constructive patterns within the data, there are different kinds of algorithms which can categorize the data either automatically or semi-automatically. These patterns are used to obtain the sets of rules. The patterns discovered must be meaningful such that they may lead to many advantages like decisions making, market analysis, financial growth, business intelligence etc. To get such meaningful patterns, significantly large amount of data is required. To cope up with this huge data, data mining takes the benefit of derived concept from machine learning and statistics. Data mining gain insights, understanding of data and provides knowledge. It is also provided capability to predict the future observations. Besides predicting future observation, data mining is also useful for summarizing the underlying relationship in data. Data mining can mine data from different data storage like text data, databases, data warehouse, transactional data, multimedia data, sequence, web, stream, time-series, multi-media, spatiotemporal, graphs and social and information networks [5].

Nowadays, data mining has grown up so huge that it is producing fruitful results in many fields like insurance, risk management, health aids, customer management, financial analysis, operation activity in manufacturing and anticipates reimbursement of corporate expense claims etc. The focus of this paper is on how data mining is relevant

in knowledge discovery at multiple levels of abstraction. Data mining examines data from various angles and sum up the outcome into precious information. It also explores data from different dimensions, after that it categorizes and summarizes the associations among them. To be precise, the process of searching the patterns and interrelation among data is known as data mining. Ongoing development in data mining contributed in several types of algorithms, drawn from the areas of database and statistics machine learning and pattern recognition, which is utile for technology utilization and adaptation. Data mining is mainly used today by companies to acquire information about their products, customers, marketing strategies and other affecting aspects. The companies can find out associations among the "external" element like customer demography and economic indicators etc. and "internal" elements such as product positioning, staff skills and price etc. by using data mining [7].

Classification is one of the techniques of data mining in which instances are gathered into identified classes. Classification is a well-liked task in data mining mainly in knowledge discovery and future plan, it provides the intelligent decision making, and classification is not only analyses the existing sample data but also estimates the future behavior to that sample data. The classification includes two phases: first is learning phase in which analysis training data, the rule and pattern generated. The second phase tests the data and evaluates the accuracy of classification patterns. Classification technique has different algorithms such as decision tree, nearest neighbor, genetic algorithm support vector machine (SVM), etc. [14]. Decision tree algorithm is widely used and one of the most effective methods of classification to approach large amounts of data in comparison to other available methods. In this paper, it is intended to survey two classification algorithms of decision tree, See 5.0 (C5.0) and Classification and Regression Tree (CART) on two different University of California Irvine datasets and compares these algorithms based on their performance and results.

1.1 Related Works

Prof. Nilima Patil, Prof. Rekha Lathi and Prof. Vidya Chitre [11] proposed “Comparison of C5.0 and CART Classification Algorithms using Pruning Technique”. In this paper, it is presented the recommendation of the membership card service in business field using classification mining techniques. The advantage of this paper is when the system constructs the decision tree; it can reduce the size of decision tree by

using pruning technique and get the better predictive accuracy. CART algorithm used pre-pruning method using Cost complexity model and C5.0 used the post pruning method by Binomial Confidence Limit. The data source as a training for this classification process had 5000 records that were calculated from membership card. As a result, the output was categorized in four classes such as normal, bronze, gold, and silver card. They performed some test cases and make a conclusion that the performance accuracy was 99.6% for C5.0 and 94.8% for CART.

Alvin Nguyen [10] analyzed "Comparative Study of C5.0 and CART algorithms". This paper is intended to compare the most two widely-used classification algorithms in data mining: C5.0 and CART for three different datasets: Iris flower, Titanic and Pima Indians Diabetes datasets. The Iris flower dataset or Anderson's Iris dataset is a multivariate dataset consisting of 50 samples from each of three species (Setosa, Virginica, and Versicolor). And each sample is explained by 4 numerical attributes: Sepal Length, Sepal Width, Petal Length and Petal Width. The system uses 120 instances for classification. Based on the implementation, the both decision trees have yielded the same percentage of accuracy 93.33%. The Titanic dataset described the survival status of individual passengers on the Titanic. It includes 1046 instances described by 6 nominal attributes. With a respect of classification capacity, it seems like C5.0 has more misclassifications than its counterpart (19.6% error rates in C5.0 compared to 18.8% error rates in CART). A total of 768 instances in Prima Indians Diabetes Database described by 9 attributes. According to the comparison test of accuracy, C5.0 (79.49%) outperforms the CART (76.92%) with respect to generalization capacity.

Su Myat Thu [17] presented "Comparative Study of Decision Tree Algorithms: ID3 and CART". In this thesis, decision tree algorithms: ID3 and CART are implemented and compared experimental results of two algorithms Stalog (Credit), Mushroom and Stalog (Heart) datasets. In training phase, ID3 produces more rules than CART for all datasets. The time complexity of ID3 is inconsiderable in the dataset contained only categorical attributes. For Mushroom dataset, CART builds the model slower than ID3 because it calculates impurity of a partition for binary tree for each attributes. In testing phase, CART outperforms ID3 in terms of classification accuracy for all datasets.

Rathinasamy Revathy and Raj Lawrance [15] proposed "Comparative Analysis of C4.5 and C5.0 Algorithms on Crop Pest Data". This paper focuses on the comparison

of C4.5 and C5.0 decision tree algorithms for pest data analysis with an experimental approach. C5.0 proved its efficiency by giving more accurate result rapidly and holding less memory while comparing c4.5 algorithm. The accuracy rate of C4.5 is predicted by a test dataset which is up to 98.48%. It obtains the error rate of 1.52%. 99.49% of data are correctly classified in C5.0 model. The error rate in C5.0 is measured as 0.51%. This research proved the efficiency of the C5.0 algorithm since it predicted more accuracy and less error rate as compared to the C4.5 algorithm.

1.2 Motivation

The proposed system is intended for classification of large volume of data from the big dataset. In this system, the two decision tree algorithms: C5.0 (See 5) and Classification and Regression Tree (CART) algorithms have been applied step by step in order to understand the rule extraction process of the algorithm and to compare the experimental results obtained from both training and testing phases. This thesis aims to compare the processing time to build the decision tree, number of rules count and classification accuracy for these two algorithms by using two UCI datasets: car quality dataset and German credit card dataset.

1.3 Objectives of the System

The system intends to study the characteristics of decision tree algorithms an under data mining system. By implementing of C5.0 and CART algorithms, the system can know how these algorithms are applied to the mining of the real-world database. Based on practical implementation, the system can examine the performance of two classification algorithms: C5.0 and CART and decide which one is better in terms of classification accuracy.

1.4 Organization of the Thesis

This thesis describes the comparison of data classification system for the two UCI datasets of car evaluation and German credit card information datasets by using C5.0 and CART algorithm. There are various data mining techniques for classification. Among them, this system uses decision tree method to classify the data. It is organized into five chapters.

Chapter 1 introduces the system and then describes the related work and objectives of the system.

Chapter 2 presents the data mining process, classification and decision tree learning technique.

Chapter 3 discusses the flow of the system, the theory and example calculation of C5.0 and CART algorithm, performance measurement of the system.

Chapter 4 describes the system architecture and detailed implementation of the system.

Chapter 5 concludes the system and discusses the limitations and further extension of the system.

CHAPTER 2
THEORETICAL BACKGROUND OF THE PROPOSED SYSTEM

This chapter involves process of data mining**,** concept of classification in data mining system, decision tree learning techniques.

2.1 Process of Data Mining

Data mining is the process of discovery through big datasets of patterns, relationships and insights that guide enterprises measuring and managing where they are and predicting where they will be in the future. Huge amount of data and databases can get from various data sources and may be stored in different data warehouses. Methods of data mining such as artificial intelligence (AI), machine learning and predictive modeling can be included. The data mining process requires commitment. But experts agree, across all industries, the data mining process is the same [8]. There are the six essential phases of the data mining process in the Figure 2.1.

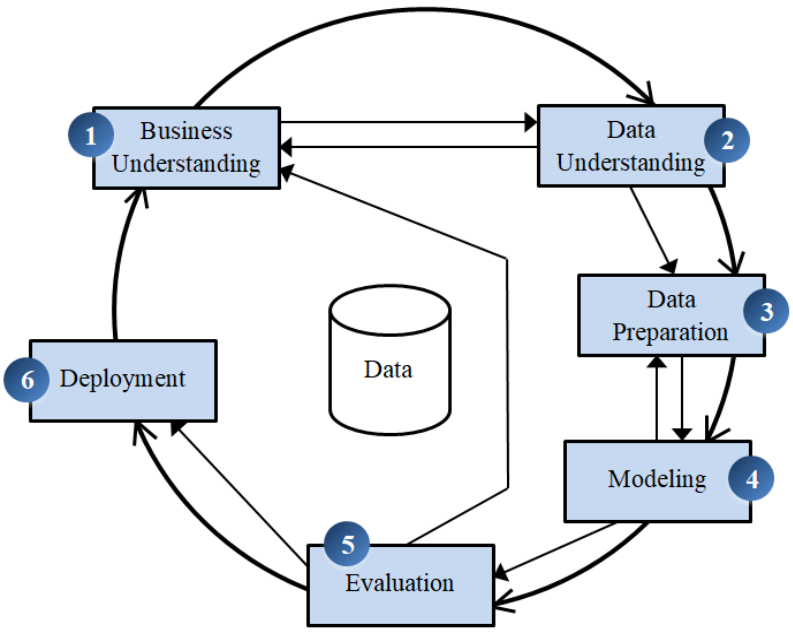


Figure 2.1 Steps of Data Mining Life Cycle

2.2 Concept of Classification in Data Mining System

Classification is the data structuring in particular classes. It uses the class labels to order the items in the collection of data. Classification techniques generally use a training dataset where all items are already connected with class labels. The classification algorithm learns from the training dataset and builds a tree structure model. The model is used to classify new objects. The classification analysis would generate a model that could be used to either accept or reject credit requests in the future.

It is a data analysis mission, i.e. the process of building a model that designates and differentiates data classes and concepts. Classification is the problem of specifying to which of a set of groups, a new study belongs to on the basis of a training data contain observations and whose types of membership is known.

There is a two-step process to predict the class labels:

1. **Learning Step:** Construct the classification model which is used to build a classifier by forming the model using the training dataset. The model has to be trained for the prediction of accurate results.
2. **Classification Step**: Testing the derived model on test data and predict the class label for the accuracy estimation of the classification rules.

In machine learning analysis, classification is a supervised learning technique in which the computer program learns from the data input given to it and then uses this method to classify new survey. This data may clearly be bi-class (like identifying whether the exam result is pass or fail or that the attendant of the student is present or absent) or it may be multi-class too [9]. Some examples of classification problems are: handwriting recognition, speech recognition, document classification, bio metric identification, etc.

Common classification algorithms in machine learning includes

1. Naive Bayes Classifier
2. Nearest Neighbor
3. Boosted Trees
4. Neural Networks
5. Decision Trees
6. Random Forest
7. Support Vector Machines

There are effective data types associated with data mining that actually relates the format of the file (whether it is in numerical formatter text format). Attributes represent different features of an object. There are two main types of attribute. These are

1. Qualitative attribute (Nominal, Ordinal, Binary)
2. Quantitative attribute (Numeric, Discrete, Continuous)

Some types of qualitative attribute and their description are

1. **Binary**: It has only two values. (For example, pass or fail, yes or no, true or false).
2. **Symmetric**: Both values are equally important in all aspects. (Gender-Male, Female).
3. **Asymmetric**: Both values are not equally important. (Result-Pass, Fail).
4. **Nominal-related to names:** The values are possible more than two outcomes, name of things, and some kind of symbols. It is in alphabet form rather than being in integer form.
5. **Categorical attributes** which is in alphabet form and there is no order among values of nominal attribute. (Color-Red, Green, Black, Yellow)
6. **Ordinal**: It has a meaningful sequence order between them, but the magnitude between values is not actually known, the order of values that expresses what is important but never point out how important it is. (Grades-A, B, C, D)

Some types of quantitative attribute and their description are

1. **Numeric:** it is a measurable quantity, defined in integer or real values. There are two types of numerical attributes, **interval** and **ratio**.
2. **Continuous**: It has infinite number of values and is float type. There may be many values between 2 and 3.(Weight-50, 51, 52, 53)
3. **Discrete**: Finite states that may be numerical and sometimes may also be in categorical form. These attributes has finite or countable infinite set of values. (Zip Code-098765, 123456)

2.3 Decision Tree Learning

Decision tree is one of the simplest and most useful Machine Learning structures. Decision trees, as the name implies, comes from the fact that the algorithm keeps dividing the dataset down into smaller and smaller portions until the data has been divided into single instances, which are then classified. It is a decision support tool that uses a tree-like model of decisions. It is one way to display an algorithm that only contains conditional control statements [12]. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning. A decision tree structure looks like a flowchart. Decision trees operate in essentially the same manner; with every internal node in the tree represent a test on an attribute. The nodes on the outside, the endpoints of the tree, are the class label after computing all attributes and they are represented leaf nodes. The branches that lead from the internal nodes to the next node represent the outcome of the test. The rules that run from the root to the leaves used to classify the data points are classification rules [6]. A decision tree consists of

Nodes: test for the value of a certain attribute

Edges: correspond to the outcome of a test connect to the next node or leaf

Leaves: terminal nodes that predict the outcome

Each node tests some attributes of dataset and each branch going out from the node corresponds to a value of that attribute. Given a tree, the process of deciding will be:

1. Start at the root (main decision)
2. Observe value of the attribute at the root
3. Follow the path (edge) that corresponds to the observed outcome
4. Repeat to expand until every line reach an end point, which give the final decision
5. predict that outcome associated with the leaf

A decision tree is a useful machine learning algorithm used for both regression and classification tasks. Decision trees operate on an algorithmic approach which splits the dataset up into individual data points based on different criteria. These splits are done with different variables, or the different features of the dataset [9]. For example, if the goal is to determine whether or not a cat or dog is being described by the input features, variables are split on might be things like “claws” and “barks”.

Decision tree methods have strengths for making decision. They are easy to understand for non-experts. It displayed graphically in a set of rules. It follows more closely approach as human decision making generally than others while modeling behavior. It can handle both categorical and numerical data while other techniques are specialized in analyzing data that have only one type of variable. It can analyze well with large amounts of data and can be robust. There are the weakness points for all methods. Decision tree methods also have weaknesses. There is a high probability of over fitting in decision tree. Generally, it often relatively inaccurate for a dataset as compared to other machine learning algorithms. Information gain in a decision tree with categorical variables gives a biased response for those attributes with more levels. Calculations are a little bit complex when there are many class labels and uncertain values.

CHAPTER 3
COMPARATIVE STUDY OF DECISION TREE ALGORITHMS: C5.0 AND CART

This chapter includes system overview, structure of proposed algorithms, datasets used for the system, calculation of algorithms with sample data and measurement unit for comparison.

3.1 Overview of the System

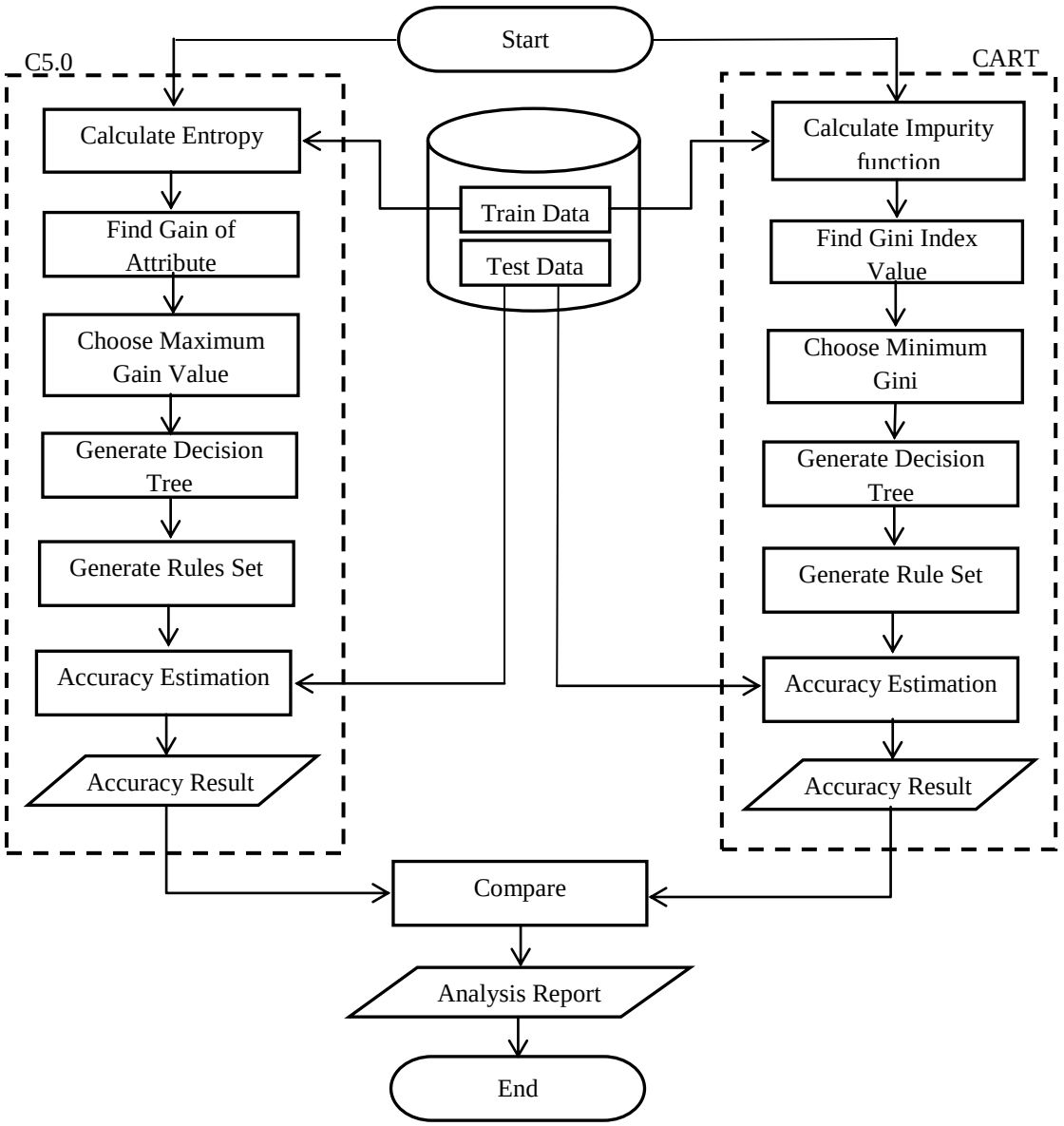


Figure 3.1 Flow Diagram of the System

In this system, there are two decision tree algorithms: C5.0 and CART are implemented for the purpose of classification. For C5.0 algorithm, the system calculates the Entropy for the whole dataset with the imported training data. The next step is the system finds the Gain values for each attributes and selects the splitting sample that has the maximum Gain value. According to that attribute, the decision tree model is constructed to classify the instances. After constructing the decision tree model, decision rules are converted into if-then format. The system accuracy is calculated with the result of decision rules and data from validation set. Finally, accuracy percentage for C5.0 is outperformed as a result. For CART algorithm, the system calculates the Gini value for the whole attributes of the dataset as an impurity function. And the system also finds the Gini index values for each attributes and selects the minimum value for splitting attribute. The decision tree is built the branches according to that splitting attribute. Decision rules are derived from the decision tree. According to that rules and testing data, the system analyzes the accuracy for CART algorithm to compare with C5.0 algorithm. At the end, the accuracies of the two algorithms are compared and displayed the comparative analysis result. Figure 3.1 shows the system flow diagram of proposed system.

3.2 Algorithms for Building Decision Tree

For the purpose of comparison, two data mining classification algorithms C5.0 and CART are implemented.

3.2.1 C5.0/ See 5 Algorithm

C5.0 is widely used as a decision tree method in machine learning, developed by J. Ross. Quinlan in 1994. C5.0 is a successor algorithm of Quinlan's earlier C4.5 algorithm which is extension of ID3. The decision trees derived by C5.0 can be used for classification, and for this reason, it is pointed to as a statistical classifier. Decision trees construction of C5.0 from a set of training dataset is as same as C4.5, using the idea of entropy and information gain.C5.0 model calculates the information gain for each attribute and select the maximum gain value as root node or the best splitting attribute. C5.0 can be easily handled all types of data like categorical, continuous, dates, times and timestamps data. It can also deal with missing values of data from dataset.

C4.5 made a number of improvements to ID3 for these facts. It can handle both discrete and continuous attributes - In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute values are less than or equal to threshold and those which are greater than it. It can also handle training dataset with missing values. For missing attribute values, C4.5 allows to be marked as "question mark (?)". Missing values are simply not taking into account in entropy and information gain calculations. It can handle attributes with differing costs. It can prune trees after creation - C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with terminal nodes.

C5.0 offers a number of improvements on C4.5. It is obviously faster than C4.5 (several orders of magnitude). C5.0 is more memory efficient than C4.5. It gets similar results to C4.5 with considerably smaller decision trees. Boosting improves the trees and gives them more accurate results. C5.0 allows you to weight different cases and misclassification types. C5.0 automatically winnows the attributes to remove those that may be unhelpful.

C5.0 has the several advantages. It can handle all types of attributes. The C5.0 rules set have noticeably lower error rates on unseen cases. It commonly needs less memory space because even it gets similar results to other algorithms with considerably smaller decision trees. It is much faster to complete the rules set construction task. The disadvantages of C5.0 are: C5.0 constructs empty branches in decision tree; it is the most crucial step for rule generation in it. The system has been found many nodes with zero values or close to zero values. These values neither contribute to generate rules nor help to construct any class for classification task. Rather it makes the tree bigger and more complex. Over fitting happens when algorithm model picks up data with uncommon characteristics. Generally C5.0 algorithm constructs trees and grows it branches 'just deep enough to perfectly classify the training examples'. This strategy performs well with noise free data. But most of the time this approach over fits the training examples with noisy data. Currently there are two approaches are widely using to bypass this over-fitting in decision tree learning. It can be susceptible to noise.

Algorithm to generate C5.0 decision tree

Input

a. Partitioning Data, D, a set of training instances and their relevant class labels
b. Attribute List, the set of attributes
c. Attribute Selection Method, a procedure to determine the splitting criterion partitions the data tuples into particular classes. This criterion consists of a splitting Attribute and either a split-pointer splitting subset

Output: C5.0 decision tree

Method:

1. create a node N
2. if tuples in data all of the same class, C, then
3. return N as a leaf node labeled with the class C
4. if attribute List is empty, then
5. return N as a leaf node labeled with the majority class in D
6. apply attribute Selection Method (D, attribute List) to find the best splitting Criterion
7. label node N with splitting Criterion
8. if splitting Attribute is discrete-valued and multi way splits allowed then
9. attribute List←attribute List - splitting Attribute
10. For each outcome j of splitting Criterion Let Dj be the set of data instances in D satisfying outcome j if Di is empty then attach a leaf labeled with
11. majority class in D to node N else, attach the node returned by generate C5.0 decision tree (Dj, attribute List) to node N
12. Return N

The expected information needed to classify a data instance in D is given by in Equation 3.1.

$$\text{Info(D)} = -\sum_{i=1}^{m} p_i \log_2(p_i) \quad (3.1)$$

where,

m = the quantity in class label

p_i = probability that an arbitrary tuple in D belongs to class i

Info(D) = the expected information in data D

Attribute A can be used to split D into v partitions or subsets, where D_j contains those tuples in D that have outcome a_j of A. The amount of information needed in order to arrive at an exact classification is measured by Equation 3.2.

$$\text{Info}_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \quad (3.2)$$

where,

$\text{Info}_A(D)$ = the expected information of each attribute in data D

v = types of the data in that attribute

In Equation 3.3, information gain is specified as the difference between the original information requirement and the new requirement.

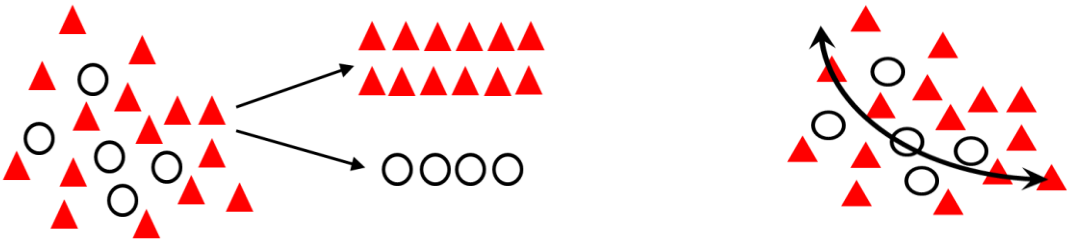
$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D) \quad (3.3)$$

Gain(A) shows that how much would be gained by branching on A. It is the expected reduction in the information requirement caused by knowing the value of A.

3.2.2 Classification and Regression Tree Algorithm (CART)

CART, short for Classification and Regression Tree was introduced by group of statisticians, Leo Breiman, Jerome Friedman, Richard Olshen, Charles Stone in 1984. It can produce either classification or regression trees, depending on the dependent variable are numeric or categorical. If the outcome variables are categorical, CART produces classification trees: if variables are continuous, CART produces regression trees. It can construct the decision tree into two values only (binary tree). It uses diversity index (Gini index) as impurity measure for selecting an attribute. Gini index will be computed by subtracting the computed sum of the squared probabilities of each class, from one. After that, select the split with minimum Gini index (or, equivalently, largest reduction in impurity). This process repeats until a suitable tree is constructed. CART accepts data with numerical or categorical values and also handles missing attribute values [3]. Classification and regression visual structures are shown in Figure 3.2.

- **Classification Trees**: where the target variable is categorical and the tree is used to identify the "class" within which a target variable would likely fall into.
- **Regression Trees**: where the target variable is continuous and tree is used to predict its value [13].



Classification visual Regression visual

Figure 3.2 Structures of Classification and Regression

There are some advantages and disadvantages of CART algorithm. It can handle both categorical and numerical variables. Classification process is done with less calculation. CART algorithm will itself identify the most significant variables and eliminate non-significant ones. It can also easily handle outliers. One of the disadvantages of CART algorithm is it can split only by one variable. It may have unstable decision tree. Insignificant modification of learning sample such as eliminating several observations and cause changes in decision tree: increasers decrease of tree complexity, changes in splitting variables and values. Complex calculation done when the problem space is bigger and chances of classification error rates that occur while training samples with few numbers of classes.

Algorithm to generate CART decision tree

```
Starting point - the tree that has a single root node
repeat
        pick a non-homogeneous tip v such that Q(v)=1
        attach to v two daughter nodes v1 and v2
        for all covariates Xj do
                find the threshold tj in the rule Xj<tj that minimizes
N(v1)Q(v1)+N(v2)Q(v2)
        end for
        find the rule Xj<tj that minimizes N(v1)Q(v1)+N(v2)Q(v2) in j and set this
best rule to node v
until all tips v are homogeneous (Q(v)=0)
set the labels of all tips
```

To measure the impurity of D, a data partition or set of training tuples as in Equation 3.4.

$$\text{Gini(D)} = 1 - \sum_{i=1}^{m} p_i^2 \quad (3.4)$$

where,

p_i = the probability that a tuple in D belongs to class C_i

m = total number of classes

For each attribute, if a binary split on A partitions D into D1 and D2, the Gini index of D given that partition is in Equation 3.5.

$$\text{Gini}_A\text{(D)} = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \quad (3.5)$$

The reduction in impurity that would be incurred by a binary split on an attribute A is in Equation 3.6.

$$\text{Gini(A)} = \text{Gini(D)} - \text{Gini}_A\text{(D)} \quad (3.6)$$

The attribute that has the minimum Gini index is selected as the splitting attribute.

3.3 Differences between C5.0 and CART Algorithm

There are well-known differences between CART and C5.0. C5.0 can have a multi way splitting or binary decision tree, whereas CART only gives a binary tree. C5.0 uses Information Gain or Entropy as an attribute selection measure to build a decision tree while CART use Gini index. Both algorithms can support for boosting and also handle the missing values. For the time complexity, C5.0 is one of highest methods of decision tree while CART algorithm can run normally. For the pruning process, CART uses pre-pruning technique called Cost – Complexity pruning to remove redundant braches from the decision tree to improve the accuracy, whereas C5.0 pruning technique adopts the Binomial Confidence Limit method to reduce the size of the tree without any loss of its predictive accuracy. Finally, in a problem of handling missing values, CART surrogates test to approximate outcomes while C5.0 apportions values probability among outcomes.

The basic characteristic of the above two algorithms are explained in Table 3.1 below. These algorithms are the most powerful data mining algorithms in the research area.

Table 3.1 Comparisons between Two Decision Tree Algorithms: C5.0 and CART

	C5.0 Algorithm	**CART Algorithm**
Type of data	Categorical, continuous, dates, times and timestamps data	Continuous and nominal data
Speed	Highest	Average
Pruning	Pre pruning (Pessimistic pruning)	Post pruning (Cost complexity pruning)
Boosting	Supported	Supported
Missing values	Can handle	Can handle
Splitting criteria	Multi split	Binary split
Formula	Use entropy and information gain	Use Gini diversity index

3.4 Datasets used in the System

Data mining is one of the critical steps in knowledge discovery involving theories, methodologies and tools for revealing patterns in data. It is important to understand the rationale behind the methods so that tools and methods have appropriate fit with the data and the objective of pattern recognition. For comparison, there may be several options for tools available for two different UCI datasets: car evaluation and German credit card information datasets.

3.4.1 Car Evaluation Dataset

Cars are one of the crucial things for our daily life activities. There are different kinds of cars as manufactured by different factory owners; therefore the purchaser makes a choice to buy. The choice of buyers is mostly depends on the price, safety, and how luxurious the car is. These points based on models, types, and manufacturers of the car. However, these points are so important in aspect like lower rate of accidents. Standard tool includes performance enhancers, safety equipment and conveniences which is part of the factors to consider when buying a car. Safety as mentioned in the factors, is really essential, also as much as conveniences which in the case of this study falls under the attributes; maintenance price, luggage boot size and number of doors.

Cost consideration is important to ensure the buying car is worth what it costs, because buying a car is a great step towards independence, but independence comes with responsibilities. To succeed it is important to understand the true financial responsibility that comes with owning a car. In this study, the attribute 'buying price' which means the price of a car to determine its acceptability or not based on its cost in relation the other necessary attributes are; maintenance price, number of doors included, number of persons to carry, space for luggage boots, and safety level of car [1]. Table 3.2 shows the car model evaluation according to the concept structure.

Table 3.2 Evaluation of Car Model according to the Concept Structure

No.	Features of Cars	Description of Cars' Features
1.	**PRICE** buying maintenance	overall price cost to buy the car fees to maintain the car
2.	**TECH**	Technical characteristics
3.	**COMFORT** doors persons luggage boot safety	comfort number of doors involved capacity in terms of persons to carry the size of luggage boot estimated safety level of car

The dataset which is obtained from the UCI dataset was donated by Marco Bohanec in 1997. The car evaluation dataset was derived from simple hierarchical decision, and is categorized descriptively in Table 3.3.

Table 3.3 Description of Car Evaluation Dataset

Dataset Characteristics:	Multivariate	**Number of Instances:**	1728
Attribute Characteristics:	Categorical	**Number of Attributes:**	6
Associated Tasks:	Classification	**Missing Values?**	No
Area:	Industry	**Date Donated**	1997-06-01

There are 6 attributes and 4 classes in car dataset. It is a collection of the records on specific attributes on cars. All attributes and their values of this dataset are shown in Table 3.4.

Table 3.4 Attributes and Values of Car Evaluation Dataset

No.	Attributes	Values
1	Buying price	very high high medium low
2	Maintenance price	very high high medium low
3	Number of doors	2 3 4 5more
4	Capacity in terms of persons to carry	2 4 more
5	Size of luggage boot	small medium big
6	Estimated safety of car	low medium high

A standard data analysis was done on the dataset to identify some design patterns in the data and the data was also presented in tables based on attribute range and their frequencies. The distribution of the four class attributes and frequencies of output classes from the analysis of data shown in Table 3.5.

Table 3.5 Frequency of Class Output from the Dataset

No.	Accessibility level of the car	Class Frequency	Relative Frequency in %
1	un-accessed	1210	70.02%
2	accessed	384	22.22%
3	good	69	3.99%
4	very good	65	3.76%

The total 1728 car data are showed in the dataset, 1210 (70.02 %) were unacceptable, 384 (22.22 %) were acceptable, 69 (3.99 %) were good, and 65 cars (3.76%) were very good. From the data mentioned above, it can be concluded that more than half of the cars were not acceptable.

3.4.2 German Credit Card Information Dataset

The bank has to make a decision regarding whether to go ahead with the bank loan approval or not, when a bank receives a loan application based on the customer's profile. There are two types of risks that are associated with the decision of bank. It is likely to repay the loan, then not approving the loan to the person results in a loss of business to the bank if the applicant is a good credit risk. It is not likely to repay the loan, then approving the loan to the person results in a financial loss to the bank if the applicant is a bad credit risk.

The goal of this analysis is to minimize the risk and maximize the profit on behalf of the bank. To reduce loss from the bank's perspective, the bank needs a decision rule regarding who to give loan approval and who not to. An applicant's demographic and socio-economic profiles are considered by loan administrators before a decision is taken by regarding his/her loan application.

The German credit dataset contains 13 variables and for 953 loan applicants the classification whether an applicant is considered a bad or a good credit risk. A predictive model developed on this data is expected to provide bank supervisor guidance for making a decision whether to approve a loan to a customer based on his/her corresponding profiles. Table 3.6 represents the description of German credit card dataset from UCI dataset.

Table 3.6 Description of German Credit Card Dataset

Dataset Characteristics:	Multivariate	**Number of Instances:**	953
Attribute Characteristics:	Categorical, Integer	**Number of Attributes:**	13
Associated Tasks:	Classification	**Missing Values?**	No
Area:	Financial	**Date Donated**	1994-11-17

In credit card dataset, there are 13 categorical attributes and 2 classes for the types of credit risks for loan applicants. All attributes and their values are shown in Table 3.7.

Table 3.7 Attributes and Values of German Credit Card Information Dataset

No.	Attributes	Values
1	Checking Status	less than 0 greater than or equal 0 and less than 200 greater than or equal 200 no checking
2	Credit History	all credits at this bank paid back duly critical account/ other credits existing (not at this bank) delay in paying off in the past existing credits paid back duly till now no credits taken/ all credits paid back duly
3	Saving Status	less than 100 greater than or equal 100 and less than 500 greater than or equal 500 and less than 1000 no known savings
4	Employment	unemployed less than 1 greater than or equal 1 and less than 4 greater than or equal 4 and less than 7 greater than or equal 7

5	Personal Status	male female
6	Other Parties	none co-applicant guarantor
7	Residence Since (years)	1 2 3 4
8	Property Magnitude	car or other life insurance/ building society savings agreement real estate unknown/ no property
9	Other Payment Plan	bank none stores
10	Housing	own rent for free
11	Job	unemployed/ unskilled and non-resident unskilled and resident skilled employee/ official management/ self-employed/ highly qualified employee/ officer
12	Own Telephone	none yes
13	Foreign Worker	yes no

In credit card dataset, 294 (30.79%) were bad credit risk, 661 (69.21 %) were good credit risk as shown in Table 3.8.

Table 3.8 Class Label of German Credit Card Information Dataset

No.	Types of risks	N# samples	N[%]
1	bad	294	30.79%
2	good	661	69.21%

3.5 Calculation of Algorithms with Sample Data

UCI Car evaluation dataset is used to show as a sample for case study. There are four types of class that are indicated the acceptability of the car. As an example, 6 categorical attributes in the small samples of 17 data records are used to train the sample decision tree model. Table 3.9 shows the sample instances for training data.

Table 3.9 Sample Training Data Records

Buying Price	Maintenance Price	Number of Doors	Capacity of Person to Carry	Size of Luggage Boot	Estimated Safety of Car	Acceptability of the Car
medium	high	3	2	small	medium	un-accessed
medium	medium	3	2	big	medium	un-accessed
low	very high	4	2	big	medium	un-accessed
low	medium	2	2	small	low	un-accessed
very high	medium	5more	2	small	high	un-accessed
very high	low	2	more	big	high	accessed
high	medium	5more	4	small	high	accessed
low	low	3	more	big	low	un-accessed
very high	very high	4	more	big	low	un-accessed
high	high	3	more	big	medium	accessed
very high	very high	5more	4	medium	medium	un-accessed
medium	very high	5more	more	small	high	accessed
very high	medium	5more	more	medium	low	un-accessed
low	medium	5more	4	medium	high	very good
medium	medium	4	4	big	high	very good
low	medium	2	4	medium	high	good
medium	low	5more	more	medium	medium	good

In car dataset for sample implementation, there are 14 tuples for “un-accessed”, 6 tuples for “accessed”, 3 tuples for “good” and 2 tuples for “very good”. By using these training sample data records, both C5.0 and CART algorithms are implemented to construct the decision tree model.

3.5.1 Implementation of C5.0 Algorithm

To find the information gain in C5.0 algorithm, the expected information needed to classify a tuple in training set is computed by using the Equation 3.1.

$$\text{Info (D)} = -\left[\left(\frac{9}{17}\right)\log_2\left(\frac{9}{17}\right) + \left(\frac{4}{17}\right)\log_2\left(\frac{4}{17}\right) + \left(\frac{2}{17}\right)\log_2\left(\frac{2}{17}\right) + \left(\frac{2}{17}\right)\log_2\left(\frac{2}{17}\right)\right]$$

$$= 1.7$$

Next, to find the splitting criterion for the partition, the expected information gain requirement for each attribute must be computed by using Equation 3.2.

$$\text{Info}_{\text{BuyingPrice}}\text{ (D)} = \left(\frac{5}{17}\right)\left[-\left(\frac{3}{5}\right)\log_2\left(\frac{3}{5}\right) - \left(\frac{0}{5}\right)\log_2\left(\frac{0}{5}\right) - \left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right) - \left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right)\right]$$
$$+ \left(\frac{2}{17}\right)\left[-\left(\frac{0}{2}\right)\log_2\left(\frac{0}{2}\right) - \left(\frac{2}{2}\right)\log_2\left(\frac{2}{2}\right) - \left(\frac{0}{2}\right)\log_2\left(\frac{0}{2}\right) - \left(\frac{0}{2}\right)\log_2\left(\frac{0}{2}\right)\right]$$
$$+ \left(\frac{5}{17}\right)\left[-\left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right) - \left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right) - \left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right) - \left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right)\right]$$
$$+ \left(\frac{5}{17}\right)\left[-\left(\frac{4}{5}\right)\log_2\left(\frac{4}{5}\right) - \left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right) - \left(\frac{0}{5}\right)\log_2\left(\frac{0}{5}\right) - \left(\frac{0}{5}\right)\log_2\left(\frac{0}{5}\right)\right]$$
$$= 1.18$$

$$\text{Info (Buying Price)} = \text{Info (D)} - \text{Info}_{\text{BuyingPrice}}\text{ (D)}$$
$$= 1.7 - 1.18$$
$$= 0.52$$

Info (Maintenance Price) = 1.7 - 1.41 = 0.29

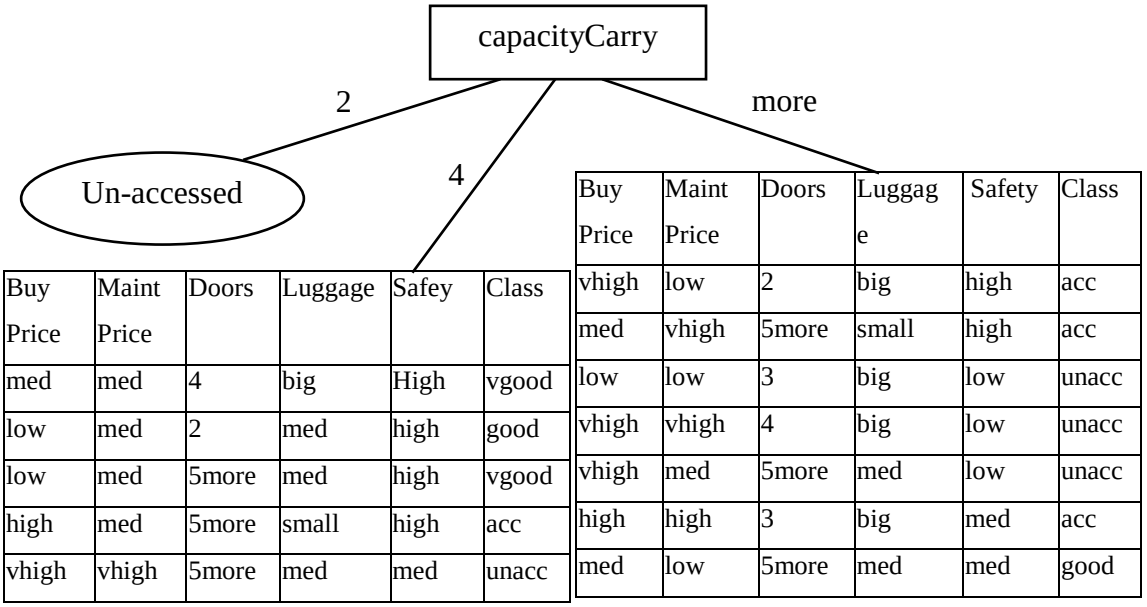
Info (Doors) = 1.7 –1.39 = 0.31

Info (Person) = 1.7 –1.16 = 0.54

Info (Luggage) = 1.7 –1.3 = 0.4

Info (Safety) = 1.7 –1.2 = 0.5

As a result, the attribute “Person” has the maximum information gain among the attributes. So, the system selects that attribute as the splitting criterion for root node. A node is created and labeled with “Person” and branches are grown for each of the attribute’s values (2, 4, more) and split the database into two dataset based on the records of the split attribute. It is shown in Figure 3.3.



Buy Price	Maint Price	Doors	Luggage	Safey	Class
med	med	4	big	High	vgood
low	med	2	med	high	good
low	med	5more	med	high	vgood
high	med	5more	small	high	acc
vhigh	vhigh	5more	med	med	unacc

Buy Price	Maint Price	Doors	Luggage	Safety	Class
vhigh	low	2	big	high	acc
med	vhigh	5more	small	high	acc
low	low	3	big	low	unacc
vhigh	vhigh	4	big	low	unacc
vhigh	med	5more	med	low	unacc
high	high	3	big	med	acc
med	low	5more	med	med	good

Figure 3.3 The First Classification Step according to the Highest Gain Attribute "Capacity Carry"

For the next partition, the attribute "Person" is deleted from the remaining data partition and the Entropy and Information Gain is computed again to select the splitting criterion for the resulting partition. This process will be repeated if all tuples in the resulting partition do not belong to one class. There is no left of attribute or tuple in the resulting partition, the process will be terminated.

3.5.2 Implementation of CART Algorithm

CART algorithm use Equation 3.4 for Gini index to find the impurity of the training set:

$$\text{Gini (D)} = 1 - \left(\frac{9}{17}\right)^2 - \left(\frac{4}{17}\right)^2 - \left(\frac{2}{17}\right)^2 - \left(\frac{2}{17}\right)^2$$

$$= 0.6367$$

To discover which attribute will be the start point node of decision tree for splitting criterion, the Gini index for each attribute is needed to compute. Since CART can split only the binary value, the system may search all possible binary split to each attribute according to its value. To determine possible binary splits on the attribute, 2^{k-1} - 1 possible subsets combination are needed to consider for an attribute with n values. For example, in attribute "Buying Price", there are four attribute values, namely {very high, high, medium, low}. Therefore, 2^{4-1} -1 possible ways to form two partitions of the data,

based on binary split on that attribute, {very high, high, medium / low}, {very high, high, low / medium }.

Apply Equation 3.5 to find the Gini index value of each attribute and search the best splitting attribute or minimum Gini index value attribute as follows:

$$Gini_{BuyingPrice \in \{very\ high,\ high,\ medium\ /\ low\}}(D) = \frac{12}{17}\left(1 - \left(\frac{6}{12}\right)^2 - \left(\frac{4}{12}\right)^2 - \left(\frac{1}{12}\right)^2 - \left(\frac{1}{12}\right)^2\right) + \frac{5}{17}\left(1 - \left(\frac{3}{5}\right)^2 - \left(\frac{0}{5}\right)^2 - \left(\frac{1}{5}\right)^2 - \left(\frac{1}{5}\right)^2\right)$$

$$= 0.6059$$

$Gini_{BuyingPrice \in \{very\ high,\ high,\ low\ /\ medium\}}(D) = 0.6235$

$Gini_{BuyingPrice \in \{very\ high,\ medium,\ low\ /\ high\}}(D) = 0.5176$

$Gini_{BuyingPrice \in \{high,\ medium,\ low\ /\ very\ high\}}(D) = 0.5941$

$Gini_{BuyingPrice \in \{very\ high,\ high\ /\ medium,\ low\}}(D) = 0.5899$

$Gini_{BuyingPrice \in \{very\ high,\ medium\ /\ high,\ low\}}(D) = 0.6269$

$Gini_{BuyingPrice \in \{very\ high,\ low\ /\ high,\ medium\}}(D) = 0.5681$

$Gini_{BuyingPrice}(D) = 0.6367 - 0.5176 = 0.1191$

$Gini_{MaintenancePrice \in \{very\ high,\ high,\ medium\ /\ low\}}(D) = 0.6134$

$Gini_{MaintenancePrice \in \{very\ high,\ high,\ low\ /\ medium\}}(D) = 0.6095$

$Gini_{MaintenancePrice \in \{very\ high,\ medium,\ low\ /\ high\}}(D) = 0.6235$

$Gini_{MaintenancePrice \in \{high,\ medium,\ low\ /\ very\ high\}}(D) = 0.6131$

$Gini_{MaintenancePrice \in \{very\ high,\ high\ /\ medium,\ low\}}(D) = 0.6061$

$Gini_{MaintenancePrice \in \{very\ high,\ medium\ /\ high,\ low\}}(D) = 0.6098$

$Gini_{MaintenancePrice \in \{very\ high,\ low\ /\ high,\ medium\}}(D) = 0.6235$

$Gini_{MaintenancePrice}(D) = 0.6367 - 0.6061 = 0.0306$

$Gini_{Doors \in \{2,\ 3,\ 4\ /\ 5more\}}(D) = 0.6269$

$Gini_{Doors \in \{2,\ 3,\ 5more\ /\ 4\}}(D) = 0.6078$

$Gini_{Doors \in \{2,\ 4,\ 5more\ /\ 3\}}(D) = 0.6131$

$Gini_{Doors \in \{3,\ 4,\ 5more\ /\ 2\}}(D) = 0.6134$

$Gini_{Doors \in \{2,\ 3\ /\ 4,\ 5more\}}(D) = 0.6235$

$Gini_{Doors \in \{2,\ 4\ /\ 3,\ 5more\}}(D) = 0.631$

$Gini_{Doors \in \{2,\ 5more\ /\ 3,\ 4\}}(D) = 0.5966$

$Gini_{Doors}(D) = 0.6367 - 0.5966 = 0.0401$

$Gini_{Person \in \{2,\ 4\ /\ more\}}(D) = 0.5933$

$Gini_{Person \in \{2,\ more\ /\ 4\}}(D) = 0.5549$

$Gini_{Person \in \{4,\ more\ /\ 2\}}(D) = 0.5098$

$Gini_{Person}(D) = 0.6367 - 0.5098 = 0.1269$

$Gini_{Luggage \in \{small, medium / big\}}(D) = 0.6235$

$Gini_{Luggage \in \{small, big / medium\}}(D) = 0.5706$

$Gini_{Luggage \in \{medium, big / small\}}(D) = 0.6118$

$Gini_{Luggage}(D) = 0.6367 - 0.5706 = 0.0661$

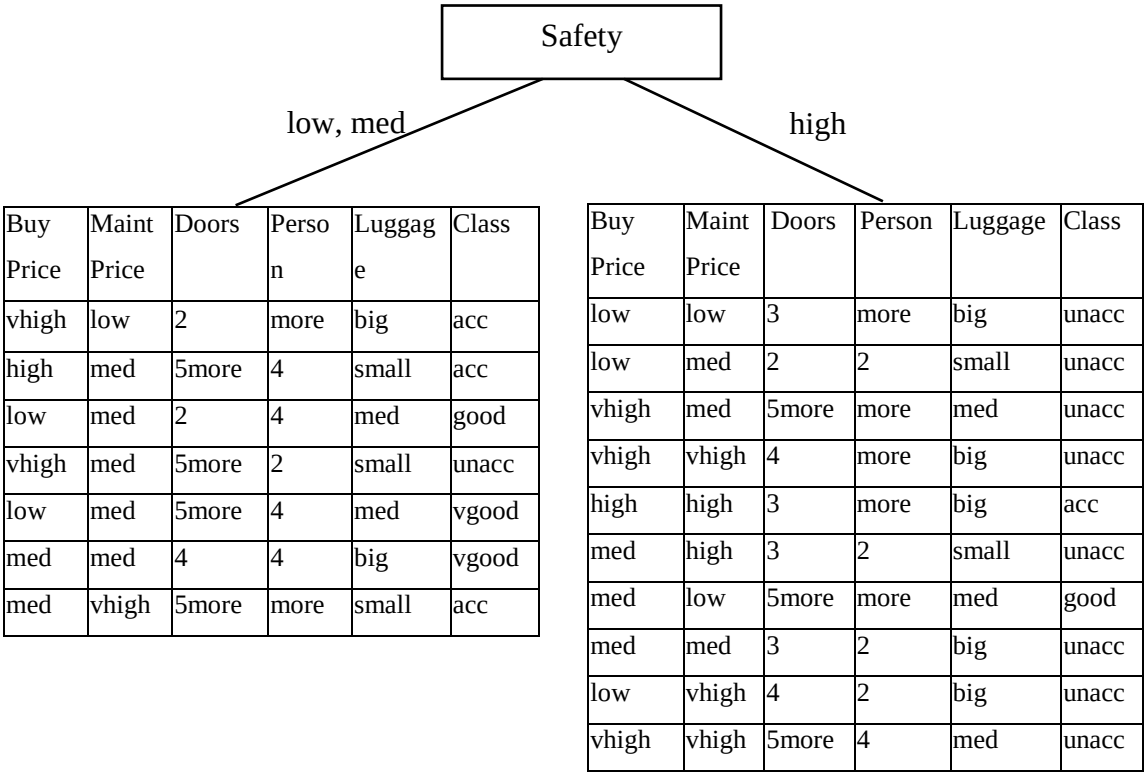
$Gini_{safety \in \{low, medium / high\}}(D) = 0.4857$

$Gini_{safety \in \{low, high / medium\}}(D) = 0.615$

$Gini_{safety \in \{medium, high / low\}}(D) = 0.543$

$Gini_{safety}(D) = 0.6367 - 0.4857 = 0.151$

After computing the above steps, the attribute "Safety{low, medium}{high}" has the smallest value of Gini index with a reduction in impurity of $\Delta Gini(Safety) = Gini(D) - Gini_{safety}(D) = 0.6367 - 0.151 = 0.4857$. The attribute "Safety{low, medium}{high}" is selected as the root node and split the data into two partition on the values of that attribute and "safety" attribute is eliminated for next the next step. The first classification according to the minimum Gini index value is shown in Figure 3.4.



Buy Price	Maint Price	Doors	Person	Luggage	Class
vhigh	low	2	more	big	acc
high	med	5more	4	small	acc
low	med	2	4	med	good
vhigh	med	5more	2	small	unacc
low	med	5more	4	med	vgood
med	med	4	4	big	vgood
med	vhigh	5more	more	small	acc

Buy Price	Maint Price	Doors	Person	Luggage	Class
low	low	3	more	big	unacc
low	med	2	2	small	unacc
vhigh	med	5more	more	med	unacc
vhigh	vhigh	4	more	big	unacc
high	high	3	more	big	acc
med	high	3	2	small	unacc
med	low	5more	more	med	good
med	med	3	2	big	unacc
low	vhigh	4	2	big	unacc
vhigh	vhigh	5more	4	med	unacc

Figure 3.4 The First Classification Step according to the Best Splitting Gini Attribute "Safety"

3.6 Stopping Criteria

The tree growing process can be continued until a stopping criterion is reached. The conditions that are common rules to stop splitting are:

1. If a node becomes pure; that is all cases in a node have identical values of the dependent variable, the node cannot be branch.
2. The node cannot be split, if all cases in a node have identical values for each predictor.
3. If the user-specified maximum tree depth limit value is equal to the current tree depth, the tree growing process will terminate.
4. If the size of user-specified minimum node is greater than the size of a node, it cannot be split [16].

3.7 Decision Rules

Once a decision tree has been constructed, it is a simple case to convert it into a decision rule sets. Some forms of predictive data mining generate rules that are conditions that imply a given outcome. Rules are if-then expressions; they explain the decisions that lead to the prediction. They are produced from a decision tree or association (such as association rule).

Converting a decision tree to rules has three main advantages:

1. Converting to rules allows distinguishing among the different contexts in which a decision node is used.
2. Unlike the decision tree, the rules do not maintain a distinction between attribute tests that occur near the root nodes of the tree and those that occur near the leaves.
3. Decision rules are simpler for people to read and easier to understand.

To generate rules, trace each path in the decision tree from root node to leaf node. Record the test outcomes as antecedents and the leaf-node classification as the consequent.

A decision rule is a simple IF-THEN statement consisting of a condition and a prediction [4]. For example: IF it snows today AND if it is December (condition), THEN it will snow tomorrow or the day after tomorrow (prediction). A single decision rule or a combination of several rules can be used to make estimations.

Decision rules follow a general structure: IF the conditions are met THEN make a certain prediction. Decision rules are probably the most interpretable prediction models. Their IF-THEN format semantically resembles natural language and provides that the condition is built from intelligible features, the length of the condition is short and there are not too many rules. In programming, it is very natural to write IF-THEN rules. New in machine learning is that the decision rules are learned through an algorithm.

3.8 Performance Measurement of the System

The system is carried out an experiment to compare the two decision tree algorithms based on their performance accuracy and precision, execution time and number of generated rules set. This experiment has carried out on two datasets taken from the University of California, Irvine Machine Learning Repository and chosen holdout method to carry out the experiment. Based on the observations on the experimental results the comparison is as follows:

3.8.1 Decision Rule Counts of the Tree

Both C5.0 and CART can produce classifiers expressed either as decision trees or rule sets. In many applications, rule sets are more favorable because they are uncomplicated and easy to understand compared with decision trees, but decision rule set methods are a little bit slow and memory-hungry. The total numbers of rules represent the total number of leaves generated by the decision tree model. This is clear that the more the number of the decision rules grows, the more the number of functioning, which has to be done for correct classification increases. If the number of decision rules decreases, the probability of correct classification will be decrease. Therefore, the clearness of this approach leads to the number of decision rules generated by the decision tree is directly proportional to the more accurate process of classification.

3.8.2 Execution Time of the Classifiers

Selecting a right classification algorithm is an important step for the success of any data mining process. Run time can be used to assess efficiency of a classification algorithm of interest. Experimenting with several algorithms can increase the cost of a

data mining project. Using the idea of meta-learning, the system has presented an approach to estimate the run time of a particular classification algorithm on an arbitrary dataset. This parameter is the time in seconds which is taken for learning and constructing decision trees. Different approaches try to shorten the time. According to the observation, the number of instances increases the time taken to construct the decision tree increases. It showed that there is a directly relationship between implementation time in building the decision tree model and the magnitude of data records and also there is an indirectly relationship between processing time in building the model and attribute size of the datasets.

3.8.3 Accuracy Analysis using Holdout Method

There are a variety of techniques to evaluate the classification accuracy and their applicability depends mainly on the dataset. The holdout method is the simplest type of cross validation approach. The dataset is splitted into mutually independent two sets, called the training set (two-third of data) and the testing set (the rest one-third) [2]. The function approximation fits a function using the training set only. Then the function approximation is asked to predict the outcome for the testing data. The errors it makes are accumulated as before to give the mean absolute test set error, which is used to evaluate the model. The advantage of this method is that it is usually preferable to the residual method and takes short time to complete the task. However, its execution can have a high variance. The evaluation may depend on which data points terminate in the training set and which terminate in the test set, and thus the analysis may be clearly different depending on how the partitioning is made. Figure 3.5 displays the holdout method of cross validation.

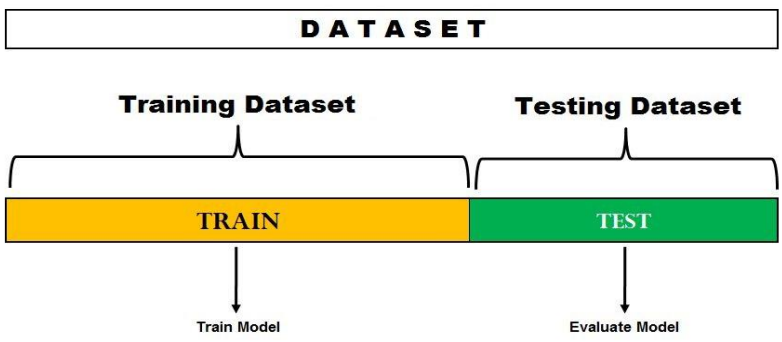


Figure 3.5 Holdout Cross Validation Method

This is the reliability of the decision tree and one of the main parameters which is used to compare the different methods. This parameter is appropriate to the percentage of test samples that are correctly classified.

In Equation 3.7, system accuracy is the ratio of the number of correctly classified instances from the test set and all instances in the test set.

$$\text{Accuracy} = \frac{\text{Number of correct classification}}{\text{Total number of instances in the dataset}} * 100 \qquad (3.7)$$

CHAPTER 4
SYSTEM ARCHITECTURE AND IMPLEMENTATION

This system is implemented to classify the corresponding class label for the two UCI datasets. In this system, there are two parts: training part to derive the decision tree and rules, and testing part for accuracy performance with rules. Figure 4.1 shows flow diagram for classification algorithms: C5.0 and CART.

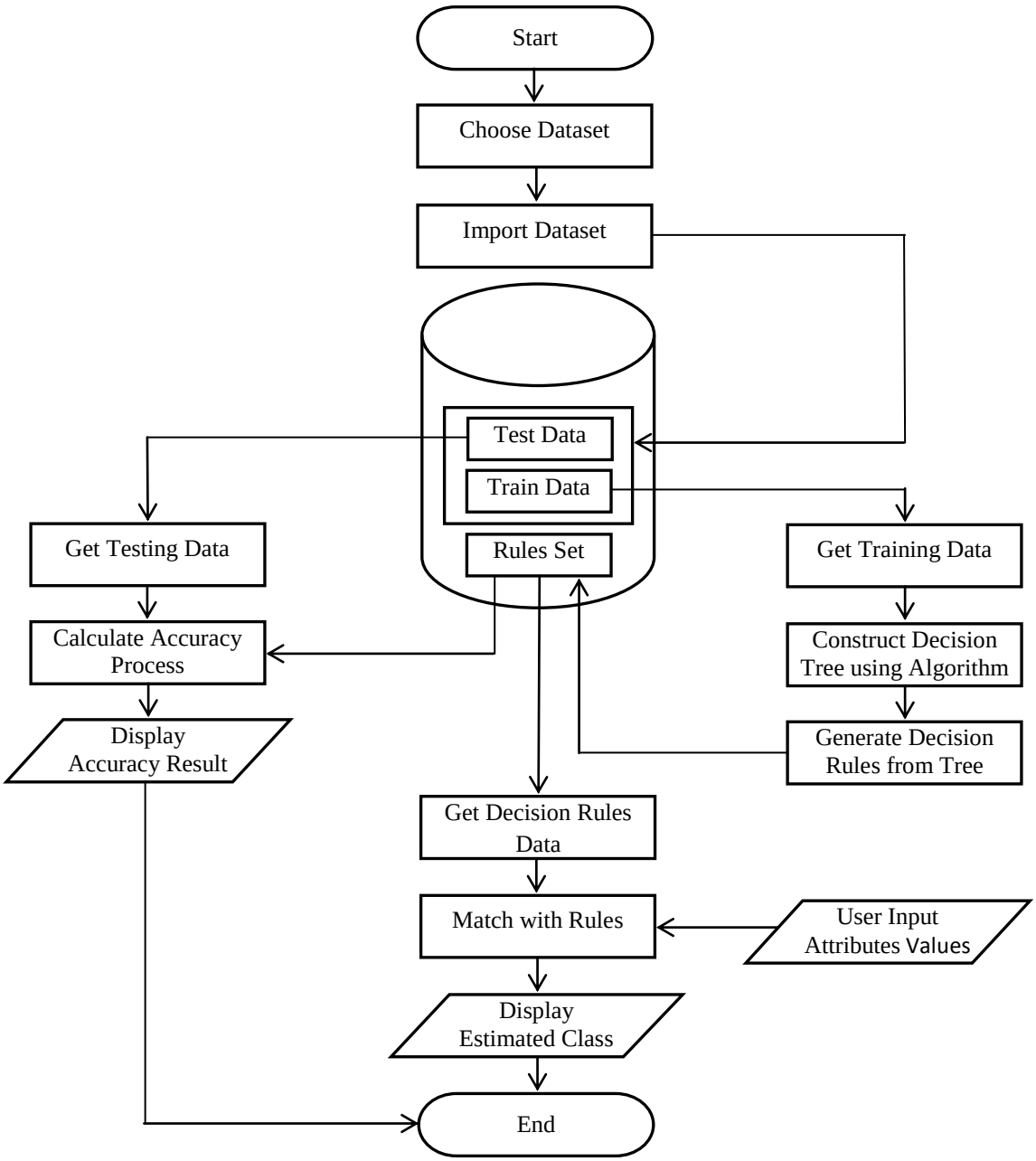


Figure 4.1 Flow Diagram for Classification Algorithms: C5.0 and CART

Firstly, the user can choose the dataset (credit card information or car evaluation) and the system can be imported the data to the corresponding database. After importing the data to the database, the data is randomly partitioned into two sets by the technique of holdout method. Two-third of the whole data is training data and the remaining (one-third) is testing data. At the training section, the proposed algorithm uses the training data and then constructs the decision tree. After construction the tree, the system produces the appropriate rules according to the decision tree. These producing rules are stored in rule dataset. At the testing section, the user can verify the system accuracy according to the test data and rules from dataset. The user can also input the appropriate information and compute the final result for the unseen cases.

4.1 Implementation of the System

This system is set up as the window based system using C# programming language on the platform of Microsoft Visual Studio 2013, and Microsoft SQL Server Management Studio 2012 for the database platform. It is implemented by using two UCI datasets, car evaluation dataset with 1728 instances and German credit card information dataset with 953 records.

4.1.1 Start Form of the System

The main form of the system is shown in Figure 4.2. The system may start by clicking submenu “Start” of the “File” menu.

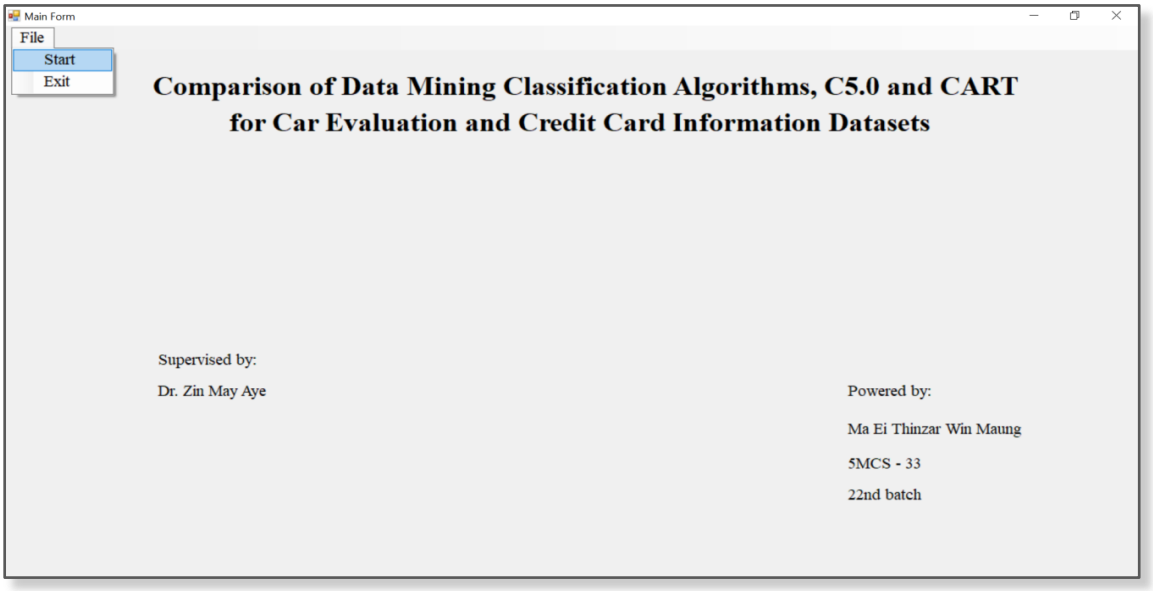


Figure 4.2 Start Point of the System

4.1.2 Import Data into the Database

Figure 4.3 shows the data importing form for the execution. Firstly, the system may choose a dataset, car evaluation dataset or German credit card dataset from the dropdown list. Then, the system may also upload the corresponding excel file to database and dataset importing process may successfully completed.

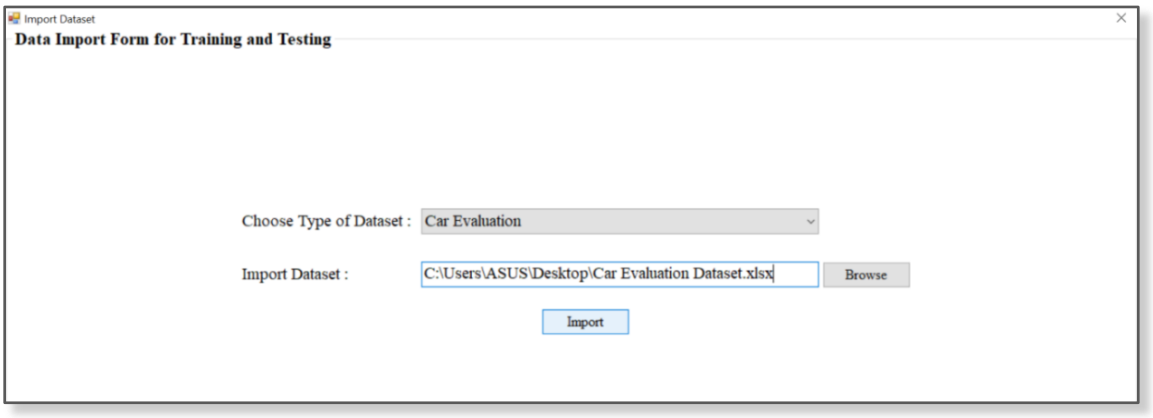


Figure 4.3 Data Import Form for the Training and Testing Data

4.1.3 System Implementation using Car Evaluation Dataset

After importing car data that may need to train and test, it may change to the two independent sets. One-third is the testing set for validation and the rest is training set for calculation. To classify the instances with the algorithms, there are four menus in Figure 4.3, namely “Training Data List”, “C5.0”, “CART”, and “Comparison Chart”.

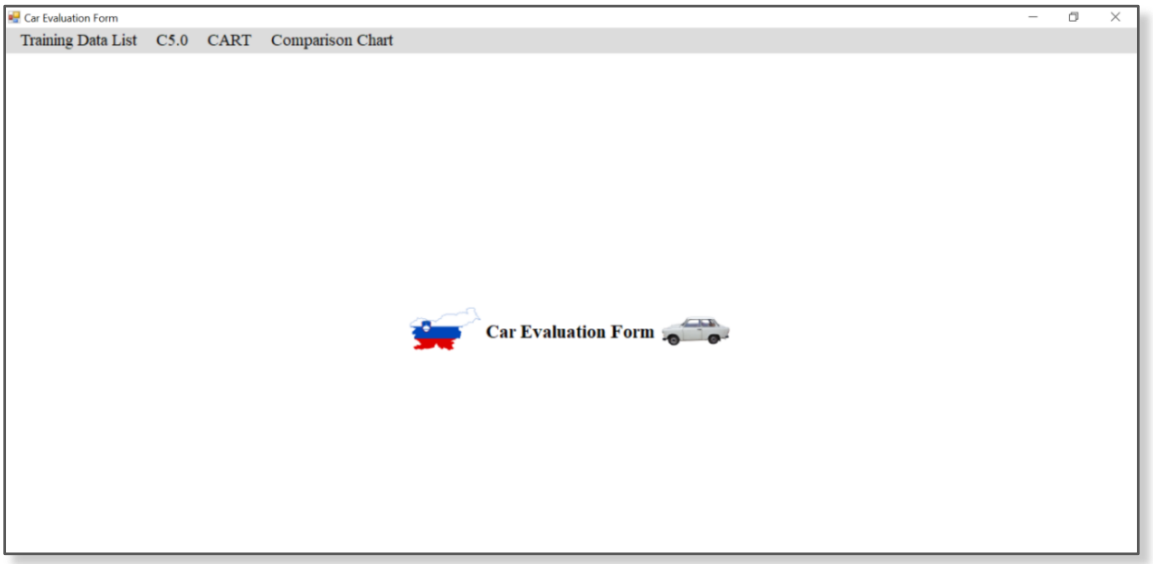
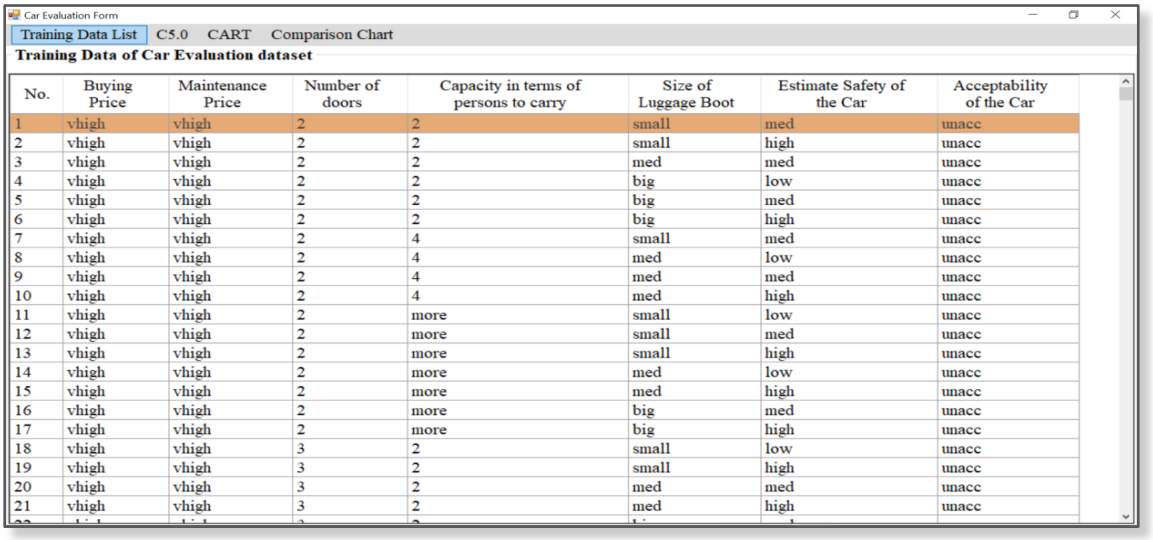


Figure 4.4 Home Form of the Car Evaluation Dataset

Figure 4.4 represents the lists of training data of the system. The dataset from processing hold-out method with the prepare dataset is the training dataset. The system may use the training dataset to generate the decision tree and rules. The list of training data is shown in Figure 4.5.



No.	Buying Price	Maintenance Price	Number of doors	Capacity in terms of persons to carry	Size of Luggage Boot	Estimate Safety of the Car	Acceptability of the Car
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	med	unacc
4	vhigh	vhigh	2	2	big	low	unacc
5	vhigh	vhigh	2	2	big	med	unacc
6	vhigh	vhigh	2	2	big	high	unacc
7	vhigh	vhigh	2	4	small	med	unacc
8	vhigh	vhigh	2	4	med	low	unacc
9	vhigh	vhigh	2	4	med	med	unacc
10	vhigh	vhigh	2	4	med	high	unacc
11	vhigh	vhigh	2	more	small	low	unacc
12	vhigh	vhigh	2	more	small	med	unacc
13	vhigh	vhigh	2	more	small	high	unacc
14	vhigh	vhigh	2	more	med	low	unacc
15	vhigh	vhigh	2	more	med	high	unacc
16	vhigh	vhigh	2	more	big	med	unacc
17	vhigh	vhigh	2	more	big	high	unacc
18	vhigh	vhigh	3	2	small	low	unacc
19	vhigh	vhigh	3	2	small	high	unacc
20	vhigh	vhigh	3	2	med	med	unacc
21	vhigh	vhigh	3	2	med	high	unacc

Figure 4.5 Lists of Training Data of the System

4.1.3.1 Implementation of C5.0 Algorithm for Car Data

When clicking “Decision Tree” of the “C5.0” tab in Figure 4.5, the decision tree generation form may appear. As shown in Figure 4.6, the decision tree model is totally generated by C5.0 based on the chosen training data records after clicking “Show Decision Tree” button on the right side panel.

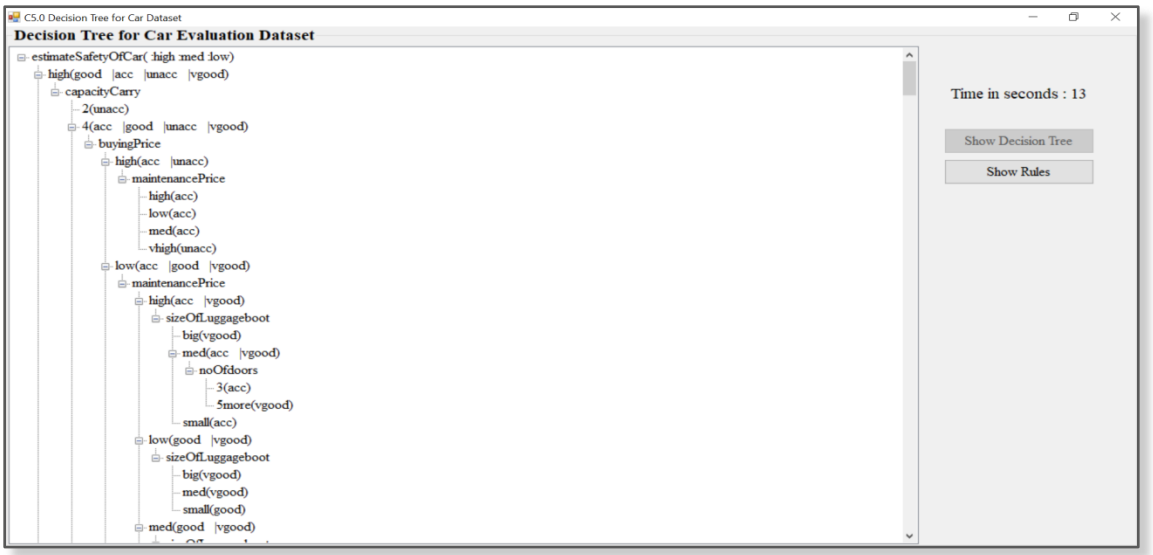


Figure 4.6 Decision Tree Constructed by the C5.0 Algorithm (Car)

The rules derived by C5.0 algorithm on the training data will display when pressing the “Show Rules” button. The IF-THEN format decision rules are shown in Figure 4.7. These rules can be easily extracted from the classification model based on the training data tuples.

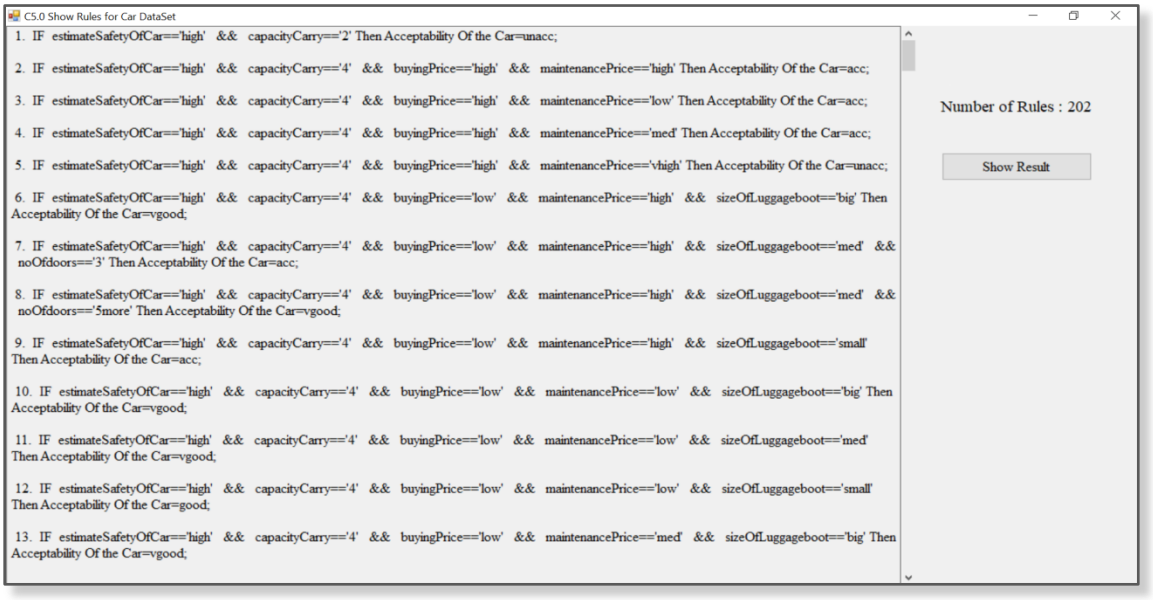


Figure 4.7 Decision Rules Generated by the C5.0 Algorithm (Car)

Figure 4.8 serves as displaying the result of execution with C5.0 algorithm for car records by pressing “Show Result” button. According to the selected dataset, number of training instances, processing time, number of rules and accuracy percentage are shown as an analysis result.

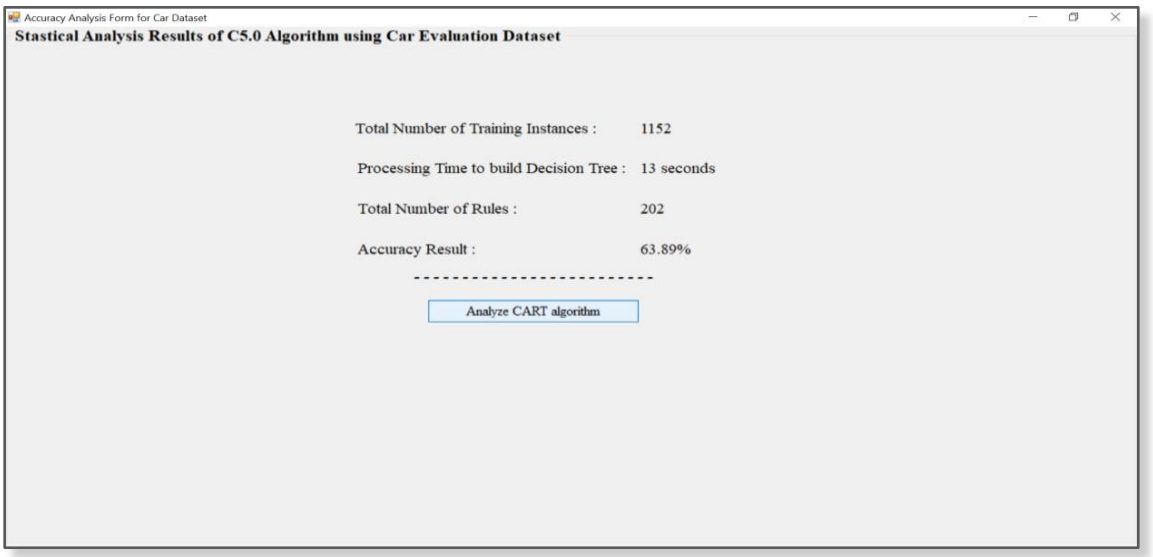


Figure 4.8 Analysis Report Form Tested by C5.0 Algorithm (Car)

By clicking on the “Testing Data” in “CART” menu, the user can find out the class level concerned with the acceptability of the car by selecting cars’ information. Figure 4.9 illustrates the classification form for the new data tuples. When the user chooses the relevant information of car from the dropdown boxes, the system classifies and estimates the class label by using the derived rules. If the classification is matched with the rules, the system generates the class label for cars’ acceptability level (un-accessed, accessed, good or very good). If not, the system produces “Not match” for unknown classification that means the error rate of the classification and can effect for the accuracy estimation.

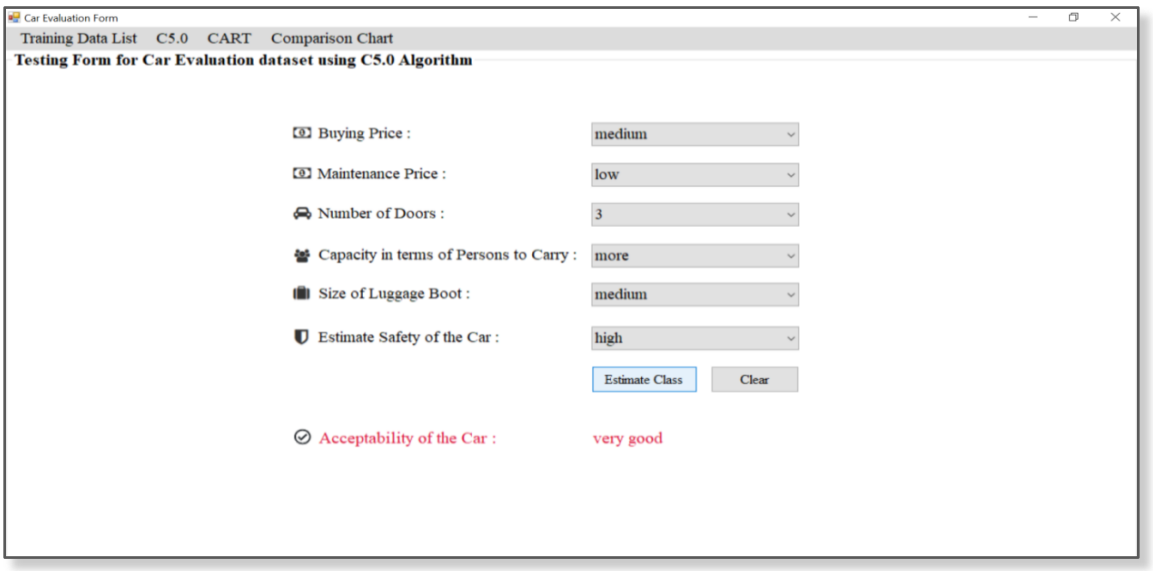


Figure 4.9 Testing the Decision Tree Model Trained by C5.0 Algorithm (Car)

4.1.3.2 Implementation of CART Algorithm for Car Data

The flow of process is the same as C5.0. Firstly, the system builds the decision tree with the training data to classify the model and evaluates the decision rules from that generated tree. According to the derived rules, the system calculates the performance accuracy and can estimate the class label for the new tuples. When the user clicks the button of “Analyze CART algorithm” from C5.0 analysis form or clicks “Decision Tree” from the menu of “CART” in main form, the decision tree form for CART algorithm is showed. In this form, the tree building process is started when the user presses the “Show Decision Tree” button. After finishing the task of building the tree, tree structure classification model is generated with the running time to build the tree as in Figure 4.10

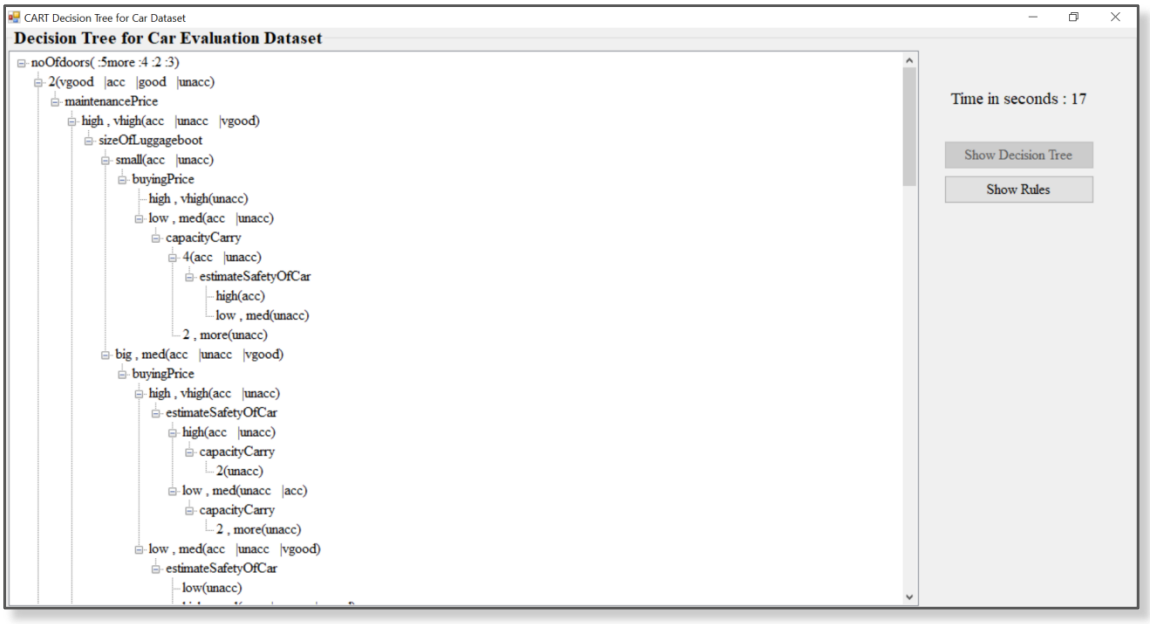


Figure 4.10 Decision Tree Constructed by the CART Algorithm (Car)

After constructing the decision tree, the decision rules are derived according to the tree model when "Show Rule" button is pressed. CART decision rules for car data are shown in Figure 4.11.

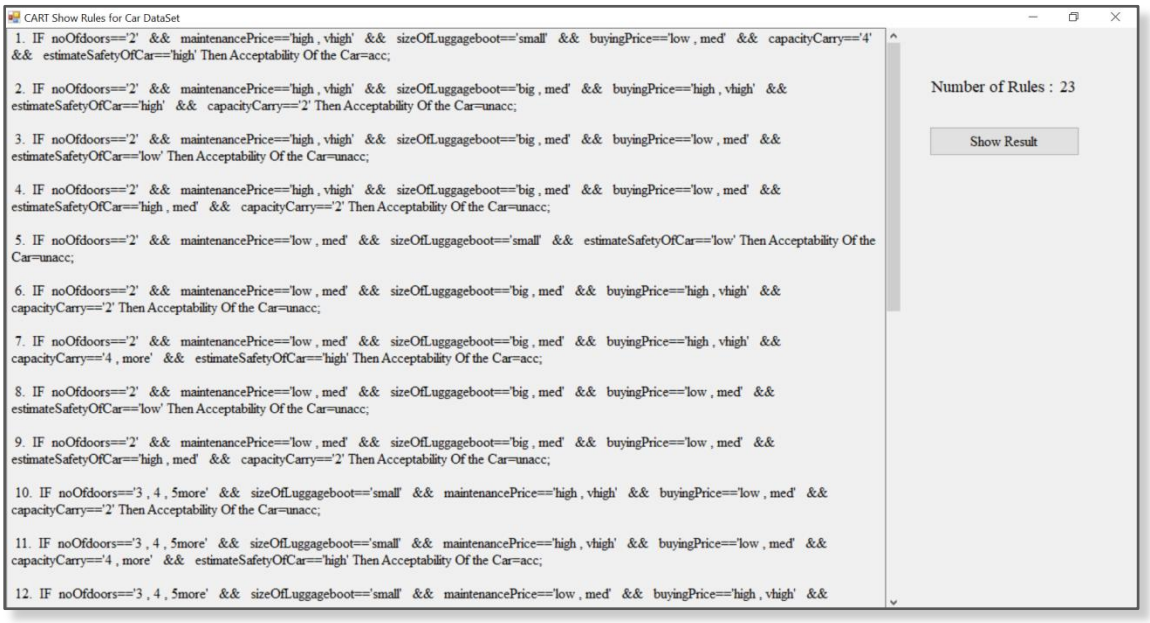


Figure 4.11 Decision Rules Generated by the CART Algorithm (Car)

Based on the implementation of decision rules, analysis report for CART algorithm that includes processing time to build the tree in seconds, number of generated rules and accuracy percentage is generated as shown in Figure 4.12.

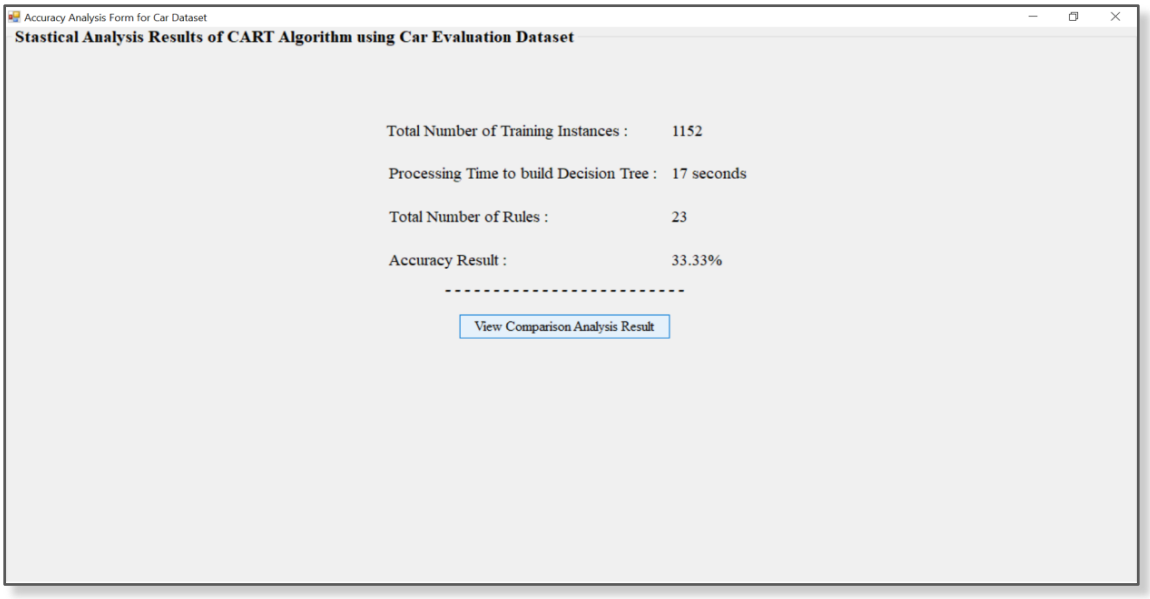


Figure 4.12 Analysis Report Form Tested by CART Algorithm (Car)

When the user clicks the "Testing Data" sub-menu in "CART" menu, testing data form is appeared as shown in Figure 4.13. If the user selects the appropriate values of each attributes for car data, the system estimates and shows the corresponding class label according to the derived decision rules.

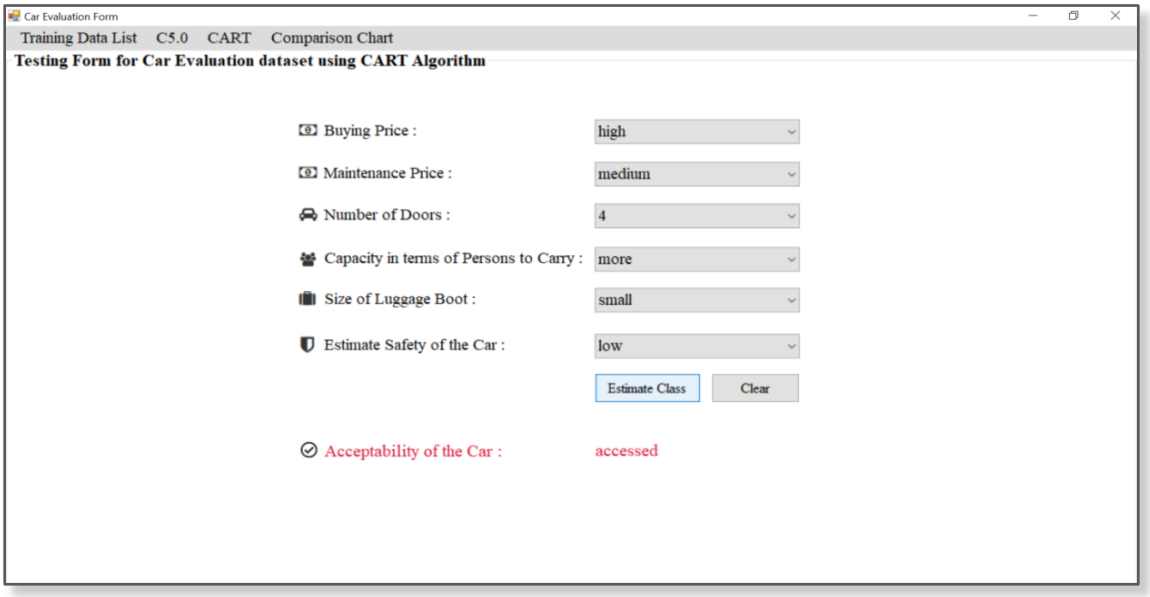


Figure 4.13 Testing the Decision Tree Model Trained by CART Algorithm (Car)

4.1.3.3 Analysis Report Chart for the Compare Process

According to the classification, the final compared outputs of the system to the selected dataset (Car data) are showed by chart by clicking “Comparison Chart” menu. Figure 4.14 shows three comparative points, execution time, rule counts, and accuracy report. For the result of processing time to build decision tree, C5.0 algorithm is slightly faster than CART because CART algorithm calculates the purity of a partition by testing ($2^{\text{number of values of an attribute-1}}$ -1) possible ways for each of categorical attributes which have more than two values. The total numbers of decision rules generated by the tree model using C5.0 is more than that derived by CART algorithm. This means that C5.0 can produce more rules number than CART to classify the test condition for the whole validation set. According to the result of testing phase, C5.0 outperforms CART in terms of predictive accuracy for this dataset.

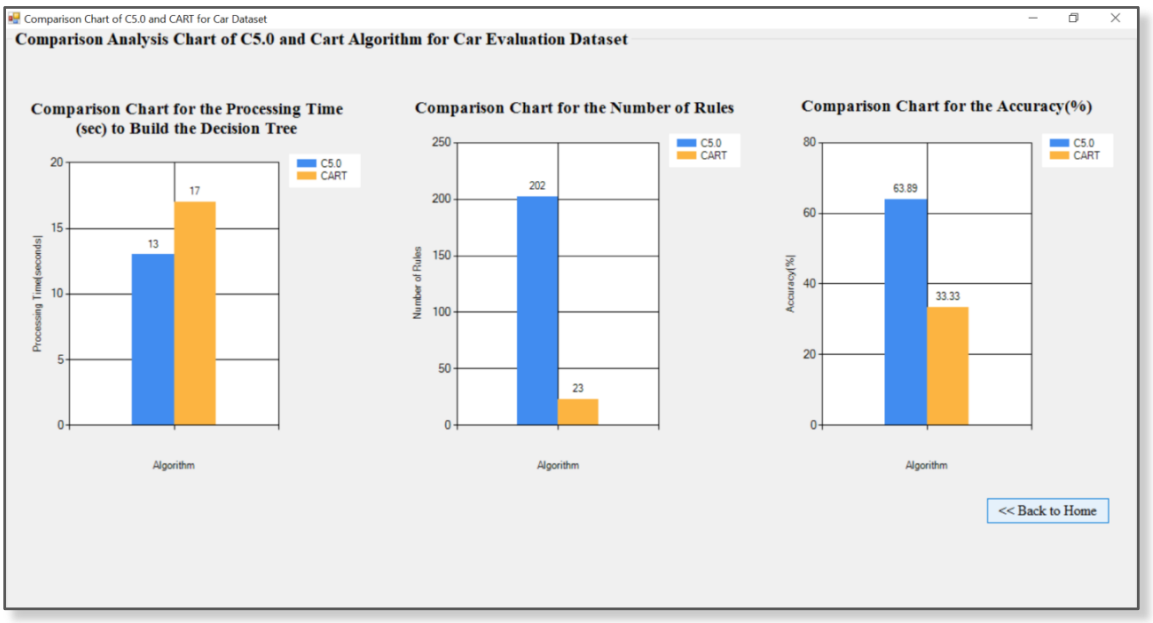


Figure 4.14 Comparison Charts that Show the Analysis Report of the Algorithms for Car Dataset

4.1.4 System Implementation using German Credit Card Dataset

When the user chooses the German credit card information data from the data source and imports to the database successfully, the home page for the credit card data is shown as in Figure 4.15.

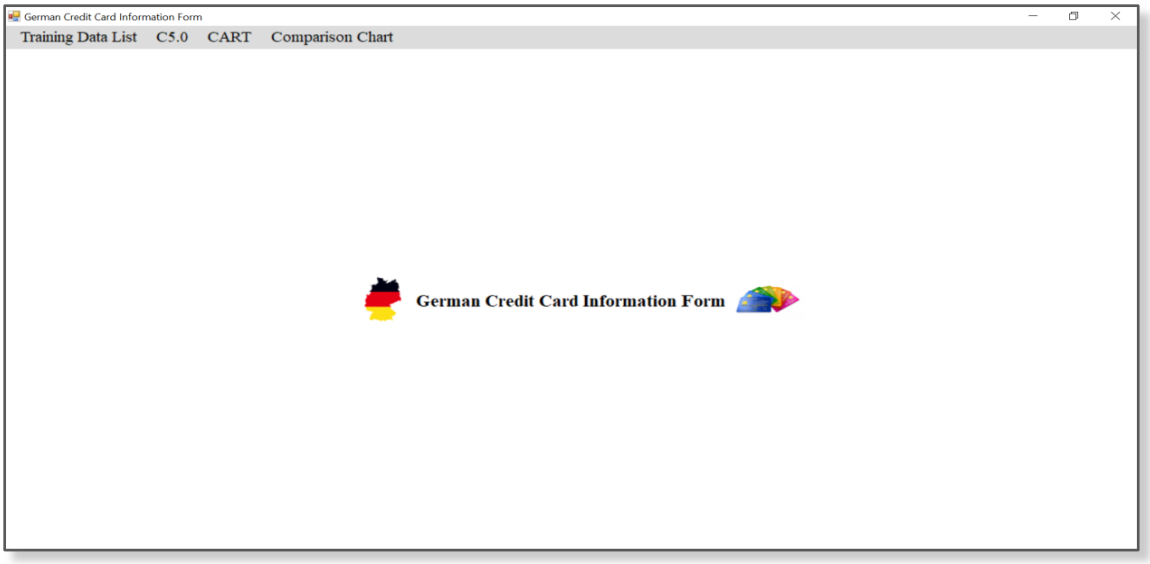
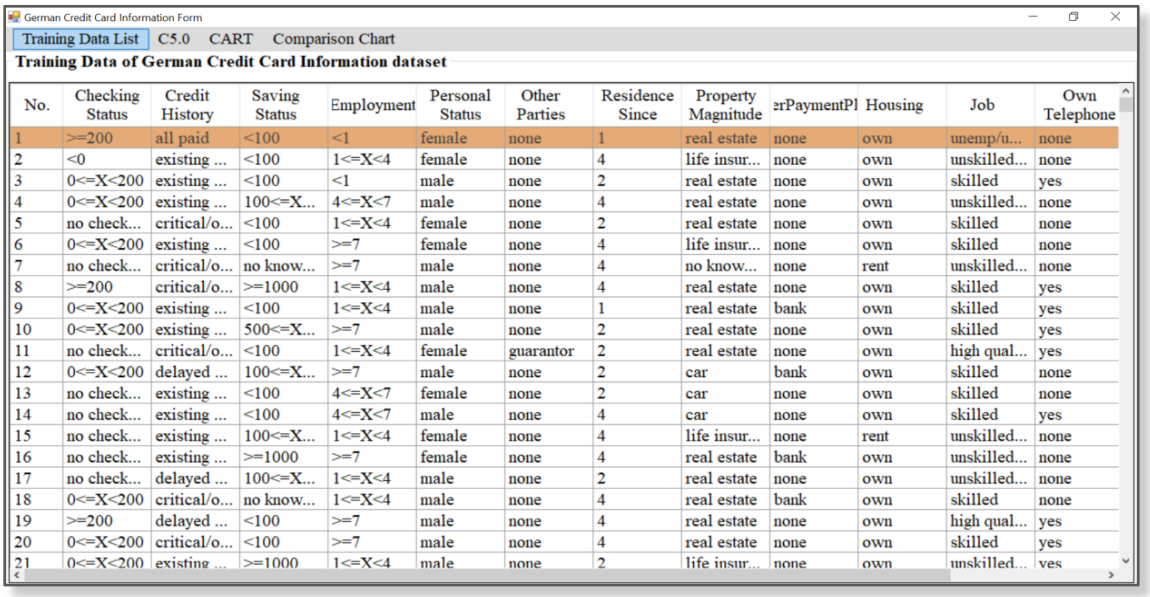


Figure 4.15 Home Form of the German Credit Card Dataset

After importing the data to the database, two-third of the whole data is randomly set as the training data and the remaining one-third is testing data. Figure 4.16 described the training data of credit card information data (636 instances of 953).



No.	Checking Status	Credit History	Saving Status	Employment	Personal Status	Other Parties	Residence Since	Property Magnitude	erPaymentPl	Housing	Job	Own Telephone
1	>=200	all paid	<100	<1	female	none	1	real estate	none	own	unemp/u...	none
2	<0	existing ...	<100	1<=X<4	female	none	4	life insur...	none	own	unskilled...	none
3	0<=X<200	existing ...	<100	<1	male	none	2	real estate	none	own	skilled	yes
4	0<=X<200	existing ...	100<=X...	4<=X<7	male	none	4	real estate	none	own	unskilled...	none
5	no check...	critical/o...	<100	1<=X<4	female	none	2	real estate	none	own	skilled	none
6	0<=X<200	existing ...	<100	>=7	female	none	4	life insur...	none	own	skilled	none
7	no check...	critical/o...	no know...	>=7	male	none	4	no know...	none	rent	unskilled...	none
8	>=200	critical/o...	>=1000	1<=X<4	male	none	4	real estate	none	own	skilled	yes
9	0<=X<200	existing ...	<100	1<=X<4	male	none	1	real estate	bank	own	skilled	yes
10	0<=X<200	existing ...	500<=X...	>=7	male	none	2	real estate	none	own	skilled	yes
11	no check...	critical/o...	<100	1<=X<4	female	guarantor	2	real estate	none	own	high qual...	yes
12	0<=X<200	delayed ...	100<=X...	>=7	male	none	2	car	bank	own	skilled	none
13	no check...	existing ...	<100	4<=X<7	female	none	2	car	none	own	skilled	none
14	no check...	existing ...	<100	4<=X<7	male	none	4	car	none	own	skilled	yes
15	no check...	existing ...	100<=X...	1<=X<4	female	none	4	life insur...	none	rent	unskilled...	none
16	no check...	existing ...	>=1000	>=7	female	none	4	real estate	bank	own	unskilled...	none
17	no check...	delayed ...	100<=X...	1<=X<4	male	none	2	real estate	none	own	unskilled...	none
18	0<=X<200	critical/o...	no know...	1<=X<4	male	none	4	real estate	bank	own	skilled	none
19	>=200	delayed ...	<100	>=7	male	none	4	real estate	none	own	high qual...	yes
20	0<=X<200	critical/o...	<100	>=7	male	none	4	real estate	none	own	skilled	yes
21	0<=X<200	existing ...	>=1000	1<=X<4	male	none	2	life insur...	none	own	unskilled...	yes

Figure 4.16 Lists of Training Data of the German Credit Card Dataset

4.1.4.1 Implementation of the C5.0 Algorithm for German Data

After clicking “Decision Tree” under “C5.0”menu, decision tree is built with the training data. Figure 4.17 shows the decision tree and processing time to construct the tree for the credit card data using CART algorithm.

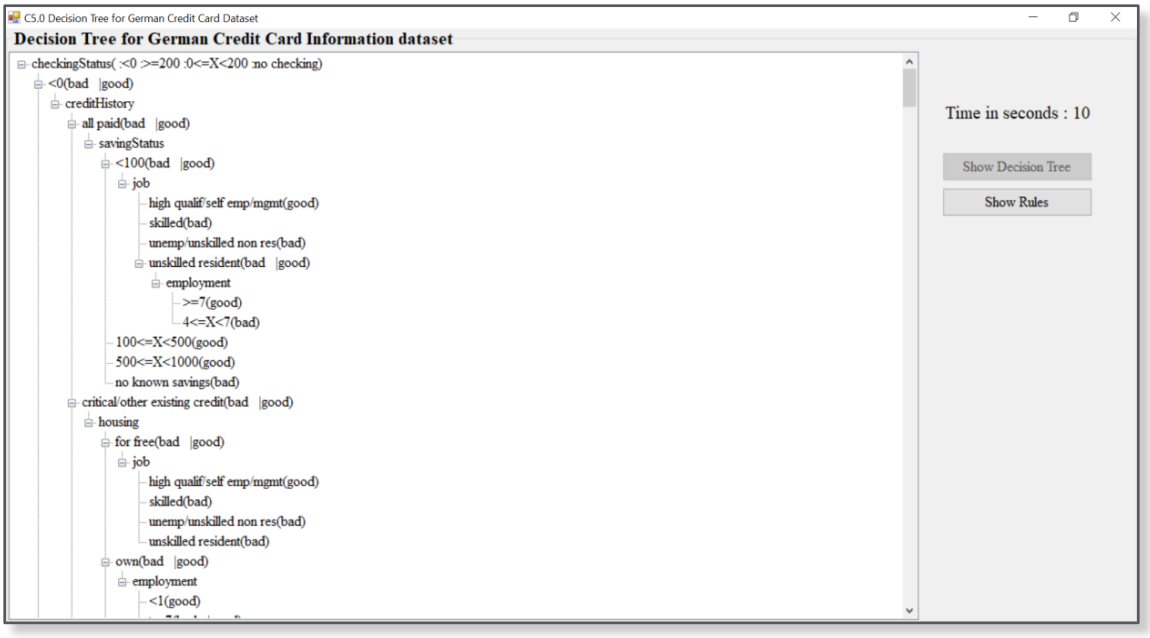


Figure 4.17 Decision Tree Constructed by the C5.0 Algorithm (Credit Card)

In Figure 4.18, decision rules are derived with if-then format when "Show Rules" button is clicked.

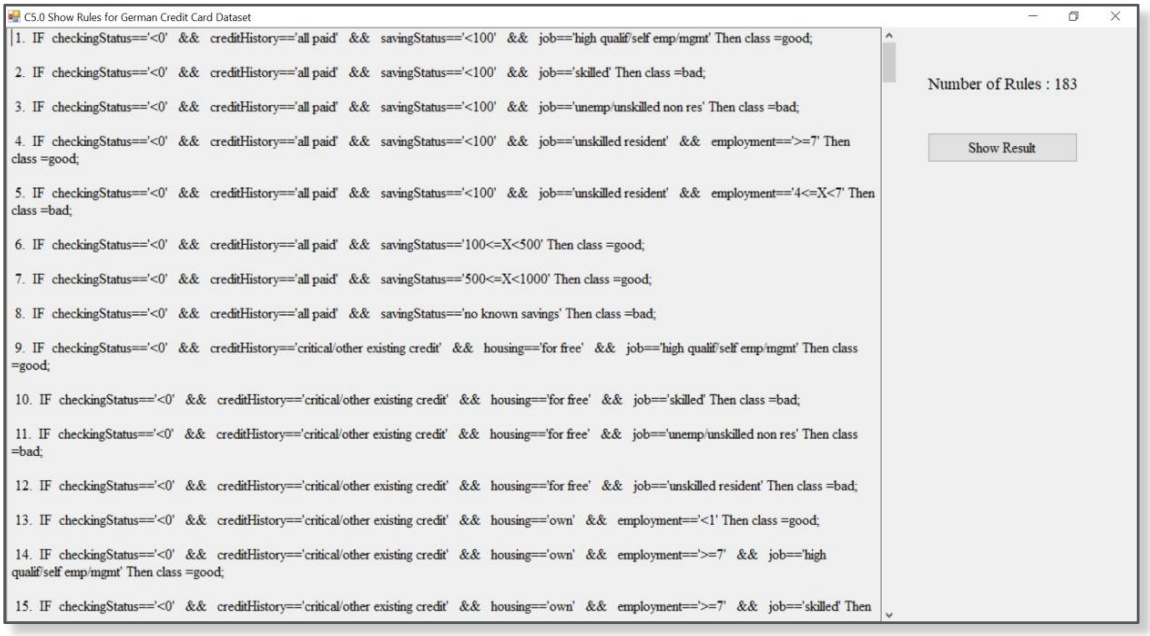


Figure 4.18 Decision Rules Generated by the C5.0 Algorithm (Credit Card)

After constructing the decision tree and rules, processing time in seconds to build the decision tree, number of rules generated and accuracy percentage are displayed as the analysis results of C5.0 algorithm for credit card data in Figure 4.19.

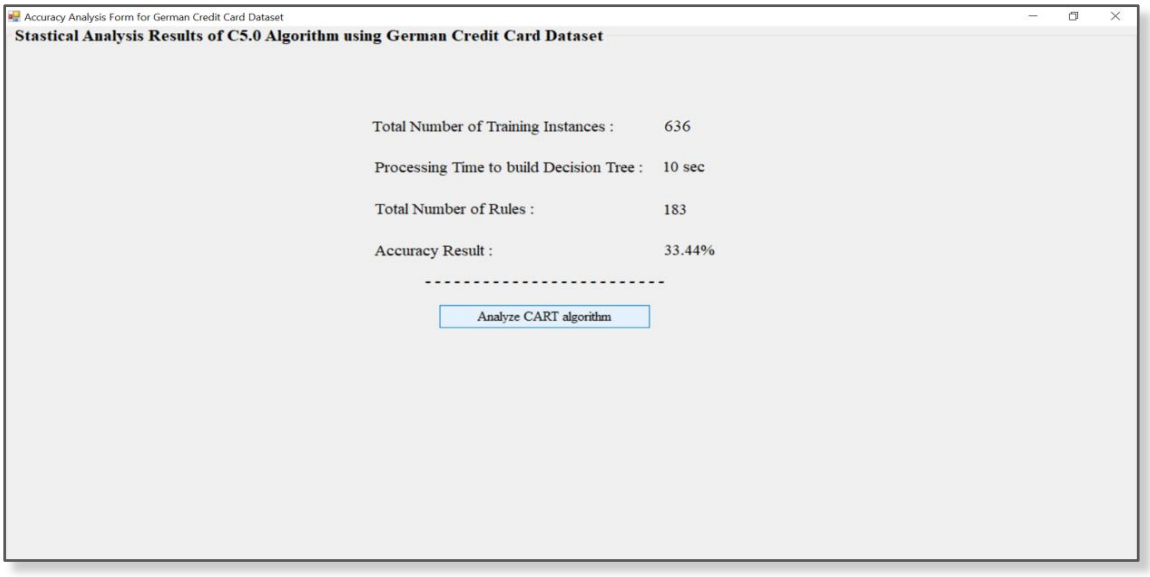


Figure 4.19 Analysis Report Form Tested by C5.0 Algorithm (Credit Card)

"Testing Data" from the C5.0 menu leads to the testing form for the credit card information data. Figure 4.20 represents the testing form of the system. When the user selects the credit card information from dropdown boxes and clicks "Estimate Class" button, the system generates the good or bad risk to pay the loan for the applicants according to the decision rules of the training data.

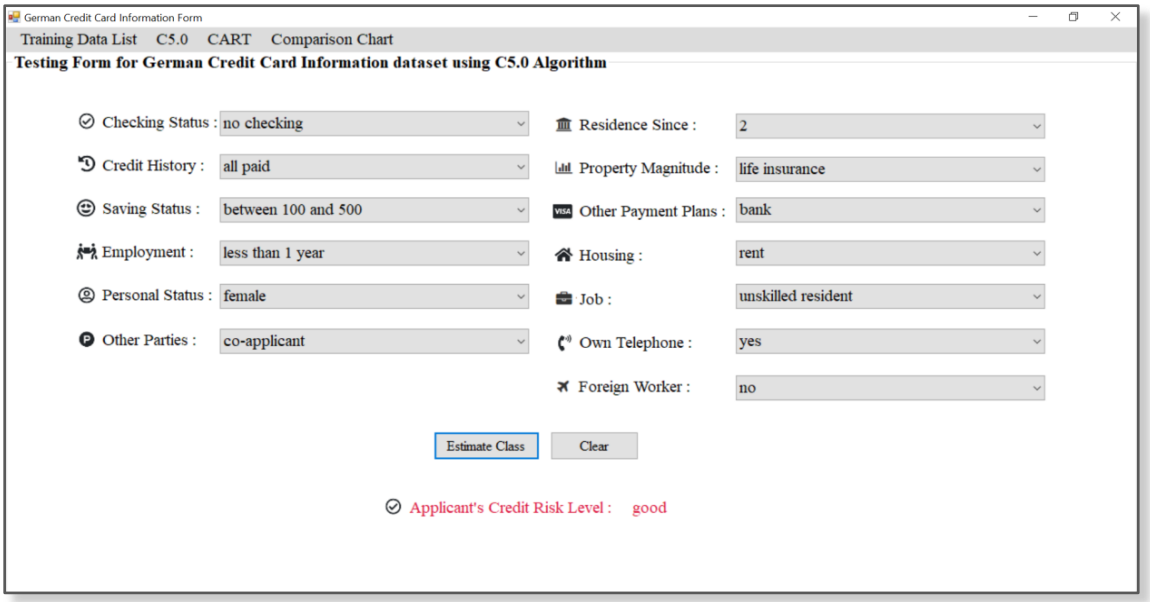


Figure 4.20 Evaluating the Model Trained by C5.0 Algorithm (Credit Card)

4.1.4.2 Implementation of CART Algorithm for German Data

In Figure 4.21, the decision tree for the CART algorithm is generated when "Analyze CART algorithm" button from C5.0 analysis result form.

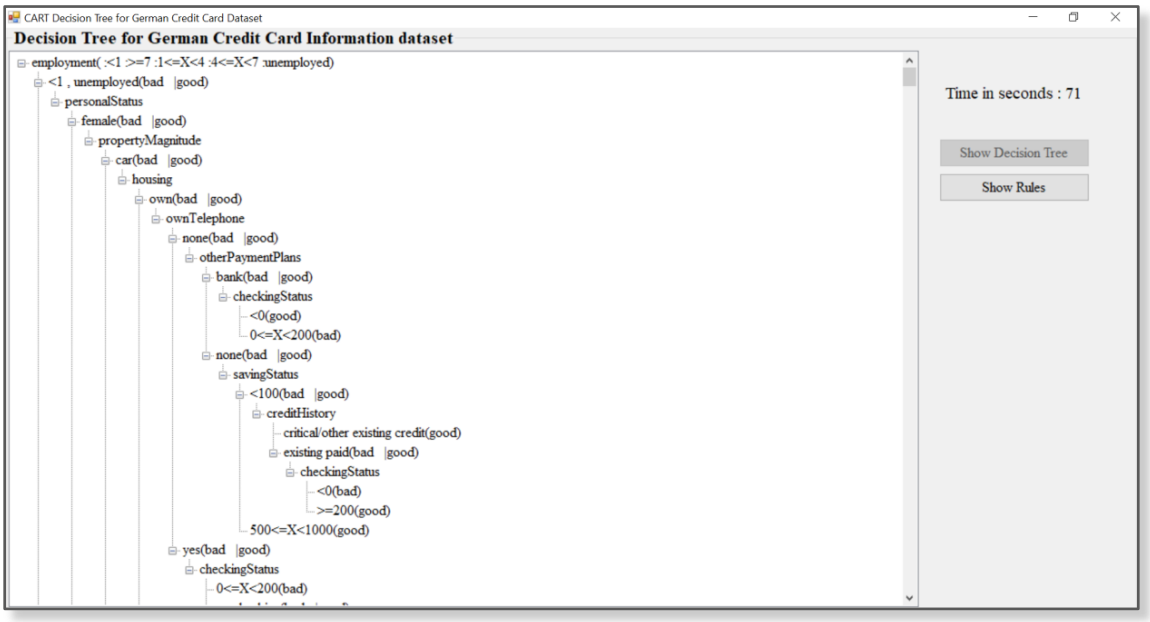


Figure 4.21 Decision Tree Constructed by the CART Algorithm (Credit Card)

When the system finishes the process of constructing the decision tree, decision rules are generated by clicking the button of "Show Rules". Figure 4.22 describes the rules with the generated total number of rules count.

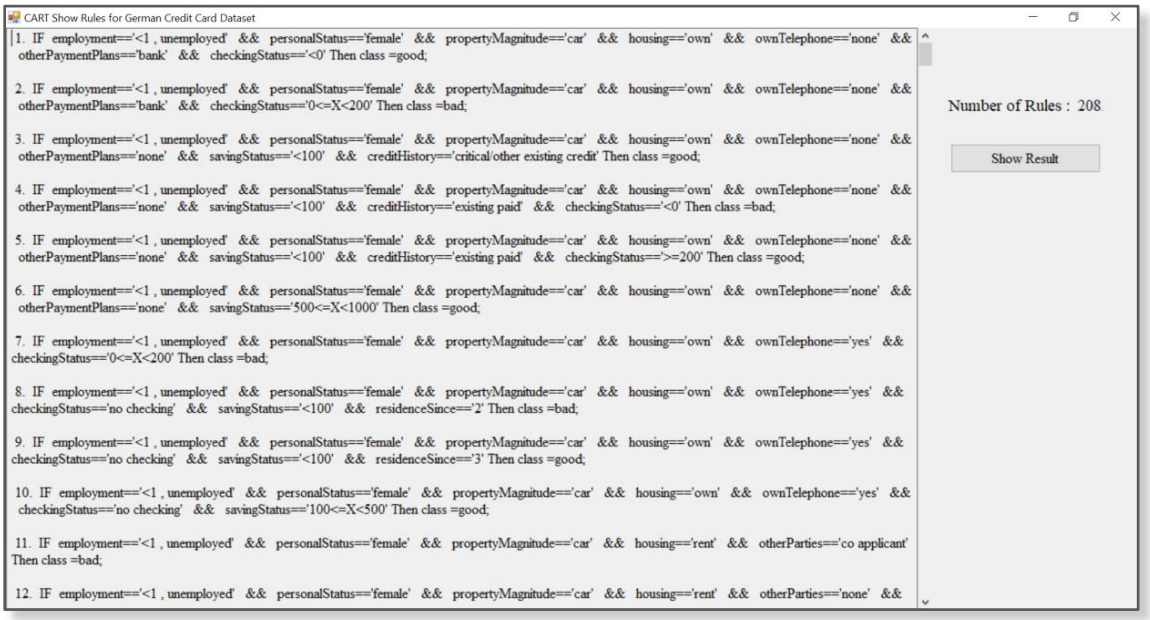


Figure 4.22 Decision Rules Generated by the CART Algorithm (Credit Card)

The analysis result of CART algorithm for credit card data are displayed in Figure 4.23 by clicking “Show Result” button.

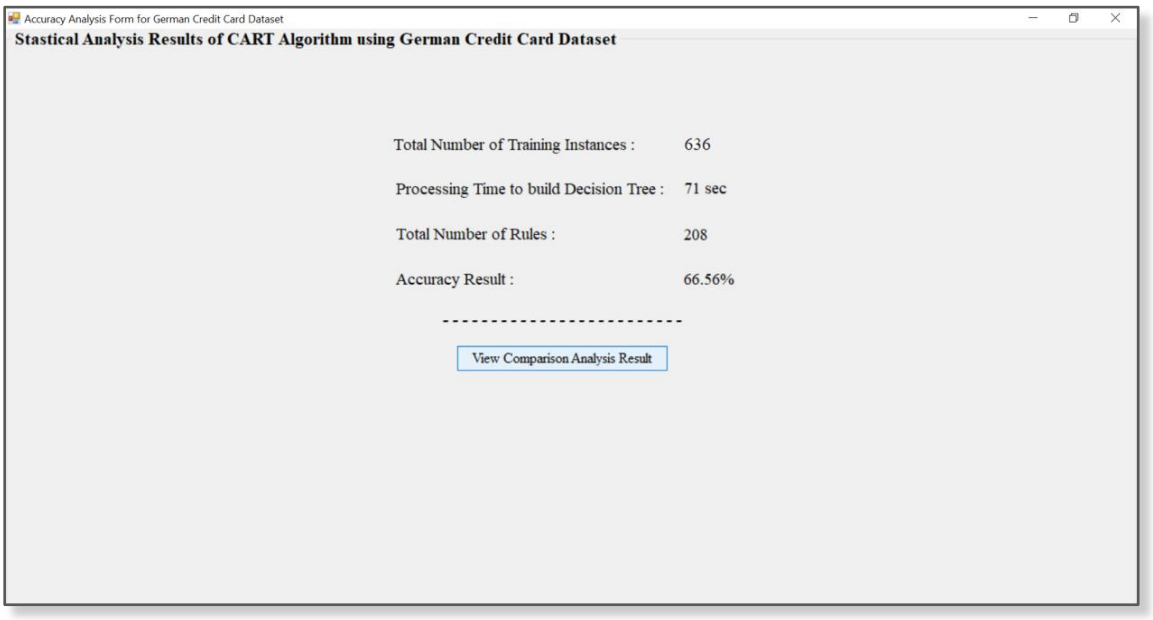


Figure 4.23 Analysis Report Form Tested by CART Algorithm (Credit Card)

Figure 4.24 shows the testing form of CART algorithm. The system can be tested for the unseen cases by selecting the appropriate values of credit card data by clicking “Estimate Class” button.

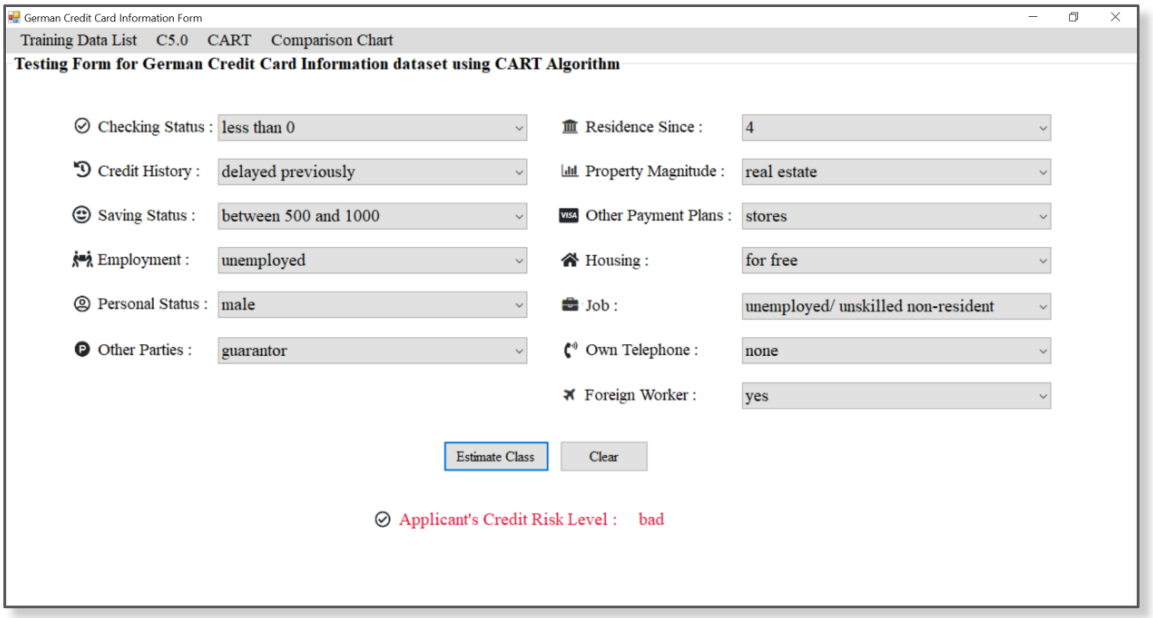


Figure 4.24 Testing the Decision Tree Model Trained by CART Algorithm (Credit Card)

4.1.4.3 Analysis Report Chart for the Compare Process

The Figure 4.25 presents the comparison output of two algorithms on the tree construction time in seconds, total number of rules produced and performance accuracy for the test cases. CART needs more than one minute to execute the tree for credit card data, but C5.0 completed the task in less than ten seconds. So, C5.0 is much faster than CART algorithm. C5.0 commonly requires less memory than CART during rule set construction because CART produces more decision rules than C5.0.Since CART generates more rules to classify the test cases, it has obviously lower error rates on unseen cases than another algorithm for the credit card dataset.

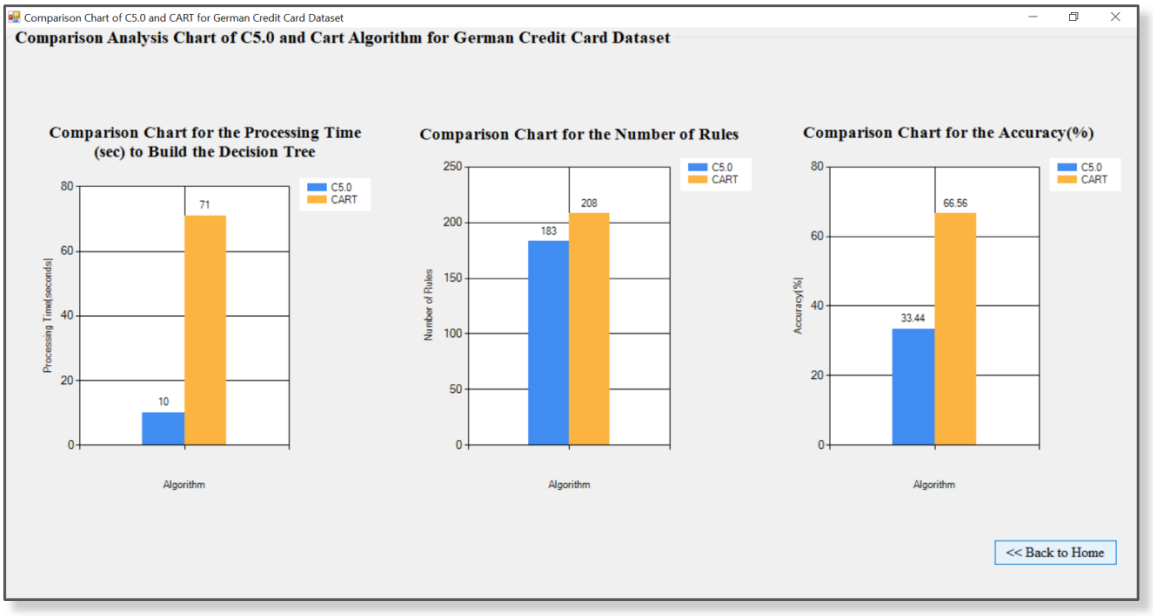


Figure 4.25 Comparison Charts that Show the Analysis Report of the Algorithms for German Credit Card Dataset

CHAPTER 5
CONCLUSION

Decision tree induction is one of the classification techniques used in decision support systems and machine learning process. With decision tree technique the training dataset is recursively partitioned using greedy technique until each partition is pure or belongs to the same class or leaf node. Decision tree model is preferred among other classification algorithms because it is a simple learning algorithm and easy to understand and implement. The system is focused on the comparison of most two widely-used classification algorithms in data mining: C5.0 and CART by using two different UCI datasets. The tree building time of C5.0 is faster than CART for two datasets because CART calculates the binary splitting values for all attributes. In credit card dataset, CART builds the tree significantly slower than C5.0 since it calculates the splitting branches for even 13 categorical attributes. In C5.0 algorithm of car dataset, the attribute of "Estimated Safety of the Car" is first split of decision tree model whereas "Number of Doors" attribute is selected for its counterpart. In the credit card dataset, "Checking Status" attribute is denoted the very first splitting attribute by C5.0 algorithm and "Employment" is first branch for CART algorithm. The main logic behind this difference is the test selection criterion of C5.0 is an information based criterion (Information Gain), whereas CART's is based on a diversity index (Gini index). Based on the model and size of the applied datasets, the decision rules generated by C5.0 algorithm is greater than that generated by CART in car dataset while CART produces more rules than C5.0 in credit card dataset. In car dataset, the system shows that the C5.0 tree model has a stronger predictive power over its counterpart. In German credit card information dataset, it is found that C5.0 usually has more misclassifications than CART in terms of accuracy.

5.1 Limitations of the System

This system focuses on the implementation of two decision tree algorithms to compare their performance. For the usage of data, only two UCI datasets are applied. At the step of data importing, firstly, the user has to prepare the excel file with the correct and exact attribute columns (especially columns' name).

5.2 Further Extension

In future, other decision tree algorithms can be implemented and compared for many classifications to increase the capabilities and efficiency of Data Mining System. In this system, C5.0 and CART classification algorithms are compared only for decision tree model growing phase. Tree pruning technique is set aside for future study. Tree pruning technique is a crucial phase of decision tree construction and is used to get better classification accuracy by ensuring that the generated tree model does not over fit the dataset. In future, the system can be performed the experimental analysis of commonly used parallel implementation tree algorithms and then compare it that implementation of decision tree algorithms and determine which one is better, based on practical implementation.

AUTHOR'S PUBLICATION

[1] Ei Thinzar Win Maung, Zin May Aye, "Comparison of Data Mining Classification Algorithms: C5.0 and CART for Car Evaluation and Credit Card Information Datasets", the Proceedings of the Conference on Parallel and Soft Computing (PSC 2020), Yangon, Myanmar, 2020.

REFERENCES

[1] J. Awwalu, A.A. Bakar, A. Ghazvini, “Performance Comparison of Data Mining Algorithms: A Case Study on Car Evaluation Dataset”, International Journal of Computer Trends and Technology (IJCTT), volume 13 number 2, 2014 July

[2] J. Awwalu, O.F.Nonyelum,"On Holdout and Cross Validation A Comparison between Neural Network and Support Vector Machine", International Journal of Trend in Research and Development, Volume 6(2), 2019 April

[3] J. Brownlee, “Classification And Regression Trees for Machine Learning”, 2016 April 8

[4] B. Hssina, H. Ezzikouri, A. Merbouha, M. Erritali, “A comparative study of decision tree ID3 and C4.5”, International Journal of Advanced Computer Science and Applications (IJACSA), 2014 July

[5] P. Jain, S.K. Vishwakarma, “A Case Study on Car Evaluation and Prediction: Comparative Analysis using Data Mining Models”, 2017 August

[6] S.V.K Kumar, P.Kiruthika, “An Overview of Classification Algorithm in Data Mining”, International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 2015 December 12

[7] B.N Lakshmi, T.S Indumathi, N. Ravi, “An Empherical Study on Decision Tree Classification Algorithms”, International Journal of Science, Engineering and Technology Research (IJSETR), 2015 November

[8] G. Mariscal, O. Marban, C. Fernandez, "A survey of data mining and knowledge discovery process models and methodologies", The Knowledge Engineering Review, Vol. 25:2, 2010

[9] S. Neelamegam, E. Ramaraj, “Classification algorithm in Data mining: An Overview”, International Journal of P2P Network Trends and Technology (IJPTT), 2013 September

[10] Alvin Nguyen, “Comparative Study of C5.0 and CART algorithms”

[11] N. Patil, R. Lathi, V. Chitre, “Comparison of C5.0 & CART Classification algorithms using pruning technique”, International Journal of Engineering Research & Technology (IJERT), 2012 June

[12] A. Priyam, Abhijeet, R. Guptaa, A. Ratheeb, S. Srivastavab, “Comparative Analysis of Decision Tree Classification Algorithms”, International Journal of Current Engineering and Technology, Vol 3, No 2, 2013 June

[13] V. Rao, “Introduction to Classification & Regression Trees (CART)”, 2013 January 13

[14] L.C. Reddy, “Comparative Study on Decision Tree Classification Algorithms in Data Mining”, International Journal of Computer Applications in Engineering, Technology and Sciences (IJCAETS)

[15] D.R. Revathy, R. Lawrance, “Comparative Analysis of C4.5 and C5.0 Algorithms on Crop Pest Data”, International Journal of Innovative Research in Computer and Communication Engineering, 2017 March

[16] S. Singh, P. Gupta, “Comparative Study ID3, CART and C4.5 Decision Tree Algorithm: A Survey”, International Journal of Advanced Information Science and Technology (IJAIST), 2014 July

[17] S.M. Thu, “Comparative Study of Decision Tree Algorithms: ID3 and CART”, M.C.Sc thesis, University of Computer Studies, Yangon, 2012 March