`

A
Dissertation
On

"Taxonomy of Tree Based Classification Algorithm in Data Mining and their Applications in predicting students Behavior in Education"

Submitted in Partial fulfillment of the requirement
For the award of Degree of

MASTER OF ENGINEERING
Computer Technology and Application
Delhi University, Delhi

SUBMITTED BY

DILPREET SINGH KOHLI
ENROLLMENT NO.:07/CTA/09

Under the Guidance of:
Dr. Daya Gupta
Head of Department
Department Of Computer Engineering
Delhi College of Engineering, Delhi

DEPARTMENT OF COMPUTER ENGINEERING
DELHI COLLEGE OF ENGINEERING
DELHI UNIVERSITY
2009-2011

`

CERTIFICATE

This is to certify that the work contained in this dissertation entitled “**Taxonomy of Tree Based Classification Algorithm in Data Mining and their Applications in predicting students Behavior in Education**” submitted in the partial fulfillment, for the award for the degree of M.E in Computer Technology and Applications at **DELHI COLLEGE OF ENGINEERING** by **DILPREET SINGH KOHLI Enrolment no. 07/CTA/09** is carried out by her under my supervision. This matter embodied in this project work has not been submitted earlier for the award of any degree or diploma in any university/institution to the best of our knowledge and belief.

(Dr. Daya Gupta)

Project Guide

Head of the Department (COE)

DCE, DELHI

Date: __/__/____

`

ACKNOWLEDGEMENT

First of all, let me thank the almighty god and our parents who are the most graceful and merciful for their blessing that contributed to the successful completion of this project.

I would like to take this opportunity to express the profound sense of gratitude and respect to all those who helped us throughout the duration of this project. **DELHI COLLEGE OF ENGINEERING**, in particular has been the source of inspiration, I acknowledge the effort of those who have contributed significantly to this project.

I feel privileged to offer sincere thanks and deep sense of gratitude to **Dr. DAYA GUPTA**, project guide for expressing her confidence in me by letting me work on a project of this magnitude and using the latest technologies and providing their support, help & encouragement in implementing this project.

`

ABSTRACT

C4.5 is a very renowned tree based classification algorithm, developed by Ross Quinlan. It is an extension of ID3 Algorithm and is used to generate a decision tree which is used for classification (a pre-processing step of data mining).It is a statistical classifier based on the concept of information entropy. There are two critical factors to this algorithm i.e. prediction accuracy and time complexity. These are directly associated to heuristic function to measure the importance of attributes which in turn is used to generate the decision tree.

In this research I propose improvements over an existing C4.5 Algorithm by introducing two new heuristic functions which are better than the one used by C4.5 Algorithm by some way or the other. The main focus is on 2 performance measures 1) Time to build the tree, 2) Prediction accuracy. To prove the existence of these improvements I apply these algorithms on some case studies (examples), two of which are proposed by me in my minor project as part of my research.

One of the biggest challenges that higher education faces today is predicting the paths of students. Colleges would like to know, for example, which students will take admission in particular course, and which students will need assistance in order to graduate. So based on the research I developed two case studies.

- A scheme of student evaluation that can help the universities (at the time of counseling) to judge whether the student matches the offered program. We take the student attributes and combine them with branch attributes and based on the historical data, satisfaction level of student for that branch is calculated.
- Here we are computing the grade of a student in a class for a particular subject. The system actually combines student attributes and subject attributes and based on the historical data, grade of the student for a particular subject is calculated.

In another case study (example) I am testing the proposed algorithms on a real AIEEE data which is in the range of thousands. The idea behind taking more than one case study (example) is to prove that the algorithms not only work well for one type of data sets but also for varied data sets. The improvements proposed over C4.5 can have a significant impact on the practical applications. We want the practical applications to be solved more efficiently and effectively. In future, a generic tool for tree based classification algorithms can be developed where user can select an appropriate algorithm for its application, depending upon its need in terms of prediction accuracy or time complexity. For example, if user is working on an application where results need to be generated faster, then it can select Algorithm1 or if the user is working on some critical application where results of classification need to be more accurate, then it can select algorithm 2.

`

TABLE OF CONTENTS

CERTIFICATE……………………………………………………………………………...2

ACKNOWLEDGEMENT……………………………………………………………..3

ABSTRACT…………………………………………………………………………4

TABLE OF CONTENTS…………………………………………………………....5

LIST OF FIGURES…………………………………………………………………...10

LIST OF TABLES……………………………………………………………………12

1. INTRODUCTION…………………………………………………………….....13

1.1 Motivation……………………………………………………………………..13

1.2 Related Work…………………………………………………………………..14

1.2.1 Improvements suggested by Quinlan………………………………………14

1.2.2 Mathematical Improvements……………………………………………….14

1.2.3 Miscellaneous Improvements…………………………………………….....15

1.3 Problem Statement……………………………………………………………..15

1.4 Scope of Work……………………………………………………………………16

1.5 Organization of Thesis………………………………………………………...16

2. DECISION TREE ALGORITHMS IN DATA MINING AND RELATED………………...18
RESEARCH WORK

2.1Data Mining Basic Concepts…………………………………………………..18

2.1.1What is Data Mining? 18

2.1.2 Data Mining Background, Research and Evolution…………………………........18

2.1.3 Introduction to Data Mining Techniques………………………………………….19

`

2.1.3.1 Clustering...19

2.1.3.2 Association...19

2.1.3.3 Classification………………………………………………………………...20

2.2 Decision Tree Based Classification..22

2.2.1 What are Decision Trees?..22

2.2.2 Using Decision Trees for Classification…………………………….…………….24

2.2.3 Types of Decision Trees…………………………………………..………………25

2.2.4 Concepts……………………………………………………………………………..25

2.2.4.1 Gini Impurity…………………………………………………………………25

2.2.4.2 Information Entropy………………………………………………………….25

2.2.5 Decision Tree Algorithms …………………………………...…………………26

2.2.5.1 Early approaches for Decision Tree Algorithm.....………..……….………..27

2.2.5.1.1 CART ………………………………………………………………...27

2.2.5.1.2 ID3……………………………………………………………………27

2.2.5.2 C4.5 – A Very Popular Decision Tree Algorithm Used For………........29 Classification

2.2.5.2.1 Exploring C4.5 Algorithm..29

2.2.5.2.1.1 Algorithm Detailed Explanation (With Flowchart) 29

2.2.5.2.1.2 Example-A case study (To explain the Algorithm)...........................31

2.2.5.2.2 Improvements made in C4.5 from ID3…………………………………35

2.2.5.2.2.1 Dealing with continuous variables…………………………………36

2.2.5.2.2.2 Improvement on Information Gain………………………………...37

2.2.5.2.2.3 Decision Tree Pruning…………………………………….…………38

2.2.5.3 Very Recent Approaches……………………….……………...……………...39

`

2.2.5.3.1 Improved Decision Tree Based Weights……………………………..39

2.2.5.3.2 Improvement by introducing a Balanced Coefficient…………………...41

2.2.5.3.3 Weighted and simplified Entropy………………………………………42

2.2.5.3.4 Improvement of Information Gain Formula……………………………43

2.2.6 Decision Tree Advantages…………………………………………………...44

2.2.7 Decision Tree Limitations …………………………………………………..44

2.2.8 Tools and software for Decision Tree Learning...45

2.2.9 Conclusion…………………………………………………………………...45

3. PROPOSED TAXANOMY OF TREE BASED CLASSIFICATION ALGORITHM 46

3.1 Introduction of New Heuristic Functions to form Improved Algorithms.........................46

3.1.1 Simplified Measure of Disorderness to reduce the Time Complexity....................47
(Algorithm 1)

3.1.1.1 Algorithm Explanation (with flow chart)..48

3.1.1.2 Example-A case study (To explain the Algorithm)...49

3.1.1.3 Comparison with C4.5 Algorithm-To prove better Time Complexity.............55

3.1.2 A More Realistic Heuristic Function that leads to better prediction accuracy........55
(Algorithm 2)

3.1.2.1 Algorithm Explanation (with flow chart)..56

3.1.2.2 Example-A case study (To explain the Algorithm)...57

3.1.2.3 Comparison with C4.5 Algorithm-To prove better Prediction accuracy……..62

4. CASE STUDIES AND RESULTS……………………………………………………..63

4.1 A case study on weather..63

4.1.1 Explanation...63

4.1.2 Application of Algorithms on the case study..64

4.1.2.1 Running C4.5 Algorithm on the Training Data..64

`

4.1.2.2 Running Algorithm 1 on the Training Data..64

4.1.2.3 Running Algorithm 2 on the Training Data..66

4.1.3 Analysis and Comparison of Results (With Snap Shots).......................................67

4.2 Predicting and Analyzing Student's behavior in Education ……………….................70
using Decision Tree Algorithm (Counseling Help)

4.2.1 Explanation...70

4.2.2 Application of Algorithms on the case study..71

4.2.2.1 Running C4.5 Algorithm on the Training Data..71

4.2.2.2 Running Algorithm 1 on the Training Data..72

4.2.2.3 Running Algorithm 2 on the Training Data..72

4.2.3 Analysis and Comparison of Results (With Snap Shots).....................................73

4.3 Predicting the level of students and monitoring their performance……………………76
using Decision Tree Algorithm (Grading System)

4.3.1 Explanation...76

4.3.2 Application of Algorithms on the case study..77

4.3.2.1 Running C4.5 Algorithm on the Training Data...77

4.3.2.2 Running Algorithm 1 on the Training Data...78

4.3.2.3 Running Algorithm 2 on the Training Data...78

4.3.3 Analysis and Comparison of Results (With Snap Shots).....................................79

4.4 Predicting the eligibility of students regarding AIEEE Counseling.……………………82
(Based on Real AIEEE data)

4.4.1 Explanation 82

4.4.2 Application of Algorithms on the case study..82

4.4.2.1 Running C4.5 Algorithm on the Training Data...82

4.4.2.2 Running Algorithm 1 on the Training Data...83

4.4.2.3 Running Algorithm 2 on the Training Data...84

`

4.4.3 Analysis and Comparison of Results (With Snap Shots)....................................84

5. IMPLEMENTATION..88

5.1 Tools and software used..88

5.2 Relations used..88

5.3 Data source creation...90

5.4 Program Architecture..91

5.4.1 Program Modules...91

5.4.2 Top-down design...92

5.4.3 Step by Step Execution of Modules (With Snapshots)..92

6. CONCLUSION AND FUTURE WORK..96

7. REFRENCES...98

APPENDIX A..100

APPENDIX B..136

APPENDIX C..169

`

LIST OF FIGURES

Figure-1. Model Construction (step 1 of classification)...21

Figure-2. Model Evaluation (step 2 of classification)..21

Figure-3. Model Use (step 3 of classification)...22

Figure-4. A Sample Decision Tree...23

Figure-5. A Sample Training Data...24

Figure-6. Example-Using Decision Tree for classification..24

Figure-7. Flow Chart for a Generic Tree Based Classification Algorithm...........................26

Figure-8. Flow Chart for a Heuristic Function Used in C4.5 Algorithm.............................31

Figure-9. Decision Tree Generated on applying C4.5 Algorithm on Weather Training......35 data

Figure-10. Flow Chart for a Generic Tree Based Classification Algorithm..........................46

Figure-11. Flow Chart for a Heuristic Function Used in Algorithm 1..................................49

Figure-12. Decision Tree Generated on applying Algorithm 1 on Weather Training data....54

Figure-13. Flow Chart for a Heuristic Function Used in Algorithm 2..................................57

Figure-14. Decision Tree Generated on applying Algorithm 2 on Weather Training data....61

Figure-15. Decision Tree Generated on applying C4.5 Algorithm on weather training data.64

Figure-16. Decision Tree Generated on applying Algorithm 1 on weather training data.......65

Figure-17. Decision Tree Generated on applying Algorithm 2 on weather training data.......67

Figure-18. Snapshot- comparing the execution time of algorithms-with an example/case.....68 study of weather

Figure-19. Snapshot-Comparing the prediction accuracies of algorithms-with an................69 example/case study of weather

`

Figure-20. Snapshot-Comparing the execution time of algorithms-with an example/case study (counselling system)....74

Figure-21. Snapshot-Comparing the prediction accuracies of algorithms-with an example/case study (counselling system)................75

Figure-22. Snapshot-Comparing the execution time of algorithms-with an example/case study (grading system)....80

Figure-23. Snapshot-Comparing the prediction accuracies of algorithms-with an example/case study (grading system)................81

Figure-24. Snapshot-Comparing the execution time of algorithms-with an example/case study (AIEEE Counselling).....85

Figure-25. Snapshot-Comparing the prediction accuracies of algorithms-with an example/case study (AIEEE Counselling)................86

Figure-26. Top-down design of the application (AIEEE Counselling)...................................92

Figure-27. Snapshot 1 (AIEEE Counselling)..92

Figure-28. Snapshot 2 (AIEEE Counselling)..93

Figure-29. Snapshot 3 (AIEEE Counselling)..94

Figure-30. Snapshot 4 (AIEEE Counselling)..95

`

LIST OF TABLES

Table-1. UCI weather Data set- Example to explain C4.5 Algorithm…………………….31

Table-2. Subset outlook=”sunny” (Produced from Table 1)……………………………...34

Table-3. Subset outlook=”overcast” (Produced from Table 1)…………………………...34

Table-4. Subset outlook=”rain” (Produced from Table 1)………………………………..34

Table-5. Subset outlook=”sunny” (Produced from Table 1)……………………………...37

Table-6. Subset outlook=”overcast” (Produced from Table 1)…………………………...37

Table-7. Subset outlook=”rain” (Produced from Table 1)………………………………..38

Table-8. UCI weather Data set- Example to explain Algorithm 1………………………..49

Table-9. Subset outlook=”sunny” (Produced from Table 8)……………………………...53

Table-10. Subset outlook=”overcast” (Produced from Table 8)…………………………...54

Table-11. Subset outlook=”rain” (Produced from Table 8)………………………………..54

Table-12. Comparison of no. of calculation/operations performed in C4.5 algorithm……..55 and algorithm 1

Table-13. UCI weather Data set- Example to explain Algorithm 2………………………..58

Table-14. Subset humidity<=76 (Produced from Table 13)………………………………60

Table-15. Subset humidity>76 (Produced from Table 13)………………………………...61

Table-16. UCI Weather data set (A case study-To compare performance of algorithms)....63

Table-17. Comparing build time and prediction accuracy (Weather Case study)………...67

Table-18. Comparing build time and prediction accuracy (Counseling system)………….74

Table-19. Comparing build time and prediction accuracy (Grading system)……………..80

Table-20. Comparing build time and prediction accuracy (AIEEE Counseling)…………..85

`

1. INTRODUCTION

In this research I am making a proposal for improvement of C4.5 Algorithm by designing two heuristic functions which are better than the one used by C4.5 by one way or the other. First heuristic function is a simplified form of the one used by C4.5. It's Time Complexity is less than C4.5 and the output decision tree generated is more or less similar to the one generated by C4.5 classifier in terms of either structure or prediction accuracy or both. Second heuristic function gives more importance to realistic attributes and thus is more real and gives more accurate and reasonable results. Although it takes more time to build but prediction accuracy comes out to be more than C4.5 algorithm. I establish my claims by implementing these algorithms on four case studies. First case study is a very simple example. It's from a weather domain which predicts whether to go for play or not. There are total four attributes outlook, temperature, humidity and windy. These attributes decide the value of the 5th attribute i.e. decision. Decision attribute can take either of the two values "yes" which means go for play and "No" which means don't go for play. Second example is a scheme of student evaluation that can help the universities (at the time of counseling) to judge whether the student matches the offered program. There are total 21 attributes (student + branch attributes) that decide the value of 22^{nd} attribute "satisfaction level".We take the student attributes and combine them with branch attributes and based on the historical data, satisfaction level of student for that branch is calculated. Third example is another case study. This system predicts the grade of a student for a particular subject by constructing a decision tree on the training set. There are around 18 parameters (student + subject attributes) that decide the grade of a student for a particular subject. Last two examples are much more complex examples than the first one. The last case study is based on the real AIEEE data containing instances in the range of thousands. Here we are predicting student's eligibility to attend AIEEE counselling i.e. whether student is eligible to attend AIEEE Counselling or not. The Decision regarding student's eligibility is estimated based on the attributes like marks in test, All India rank, state rank, category ,date of birth etc. The idea behind taking more than one case study (example) is to prove that the algorithms not only work well for simpler data sets but also for complex data sets.

1.1 Motivation

One of the biggest challenges that education sector/ society faces today is predicting the paths of students. Colleges would like to know, for example, which students will take admission in particular course and which students will need assistance in order to graduate. Students for example would like to know which subject/ course is best for them. Data mining techniques can

`

be used for such applications to predict the behavior of students. ID3 and C4.5 are very renowned classification algorithms. There are two critical factors associated with them 1) Time complexity and 2) prediction. These algorithms can be customized based on application domain. We want practical applications to be solved more effectively and efficiently. Two types of applications always exist in the real word. First type of applications is the one where results need to be generated faster e.g. real estate market prediction or share market prediction. Second ones are the critical applications where results of classification need to be more accurate e.g.: predicting student's behavior in education, medical applications where doctor needs to predict a disease based on the symptoms of the patient, business applications where a marketing professional needs complete description of customer segments to successfully launch a marketing campaign etc. That is why I propose two improvements over an existing C4.5 Algorithm by introducing two new heuristic functions. First heuristic function is better in terms of execution time. Second heuristic function is more realistic, gives importance to realistic attributes and thus gives more accurate and reasonable results. These Improvements can have a significant impact on practical applications.

1.2 Related Work

Many decision-tree algorithms have been developed. One of the most famous is ID3 (Quinlan 1986), [21] whose attribute selection criterion is based on information entropy. C4.5 is a very renowned tree based classification algorithm. It is an extension of ID3 algorithm (Quinlan 1997) [22]. It improves prediction accuracy, deals with continuous attributes, handles missing attribute values, avoids over fitting and performs other functions [3, 22]. Several attempts have been made by researchers to improve the existing ID3 and C4.5 algorithm. These improvements can be categorized in to three types.

1.2.1 Improvements suggested by Quinlan

Quinlan himself proposed some improvements over an ID3 algorithm [3, 22]. Firstly he proposed an improved method to handle continuous attributes [3, 22]. He offered to select a threshold on the values of attributes and then divide the given set of training examples in to two subsets [3]. One set contains attribute values greater than the threshold and other set contains attributes values less than or equal to the threshold [3, 22]. Secondly he suggested to not to use missing attribute values in the calculation of information gain ratio [22]. Thirdly he proposed the method of pruning [22]. It is the process of removing unwanted sections of the tree which are generated due to noise or too small set of training data or large no. of parameters/attributes [22]. This is done to improve prediction accuracy [22]. They are explained in detail in section 2.2.5.

1.2.2 Mathematical Improvements

`

Several attempts have been made to mathematically improve the performance of C4.5 algorithm. Researchers have tried to simplify the formula of information gain to reduce the time complexity. In one case they removed the log operations [1,15,16,17] and in other case they separated a big constant and then removed it from the formula of information gain [2,18,19,20] to reduce the no. of operations. They are explained in detail in section 2.2.5.

1.2.3 Miscellaneous Improvements

Another improvement was given by LI Rui, WEI Xianmei and YU Xue-wei [4]. Since C4.5 uses divide and conquer strategy and searches the best node locally, they suggested a method to improve optimality of C4.5 algorithm by proposing a balanced coefficient [4]. Users can decide the value of this balanced coefficient according to the situation and by using their intellectual and domain knowledge [4]. It's an unclear concept and only artificially improves the efficiency of the algorithm. According to Weizhao Guo and Jian Yin [5] , since many real world data sets are imbalanced in nature so they introduced a new improved decision tree based weights, which considers imbalanced weights between different instances, to address the class imbalanced problems. The proposed decision tree algorithm is simple and more effective in implementation than previous decision trees and the experiment results testify that the proposed algorithm outperforms C4.5 significantly, in terms of the improvement of the classification accuracy in UCI data sets [5]. They are explained in detail in section 2.2.5.

Drawback* Most of the improvements done on C4.5 Algorithm till now are mathematical improvements where the researchers have tried to simplify the formula of information gain by using some approximations and laws of mathematics. Such improvements do not have any impact on prediction accuracy and also they marginally improve the time complexity of the algorithm. The Miscellaneous improvements that are discussed above also have very less impact on time complexity and prediction accuracy. So in this research I propose two conceptual improvements which have a huge impact on time complexity and prediction accuracy.

1.3 Problem Statement

In this research I develop taxonomy of tree based classification algorithms which are improvement of C4.5 Algorithm.

"To propose improvements of C4.5 Algorithm by developing two heuristic functions which are better than the one used by C4.5 Algorithm. First one is better in terms of time taken to build the tree and second one is better in terms of prediction accuracy."

First heuristic function is a simplified form of the one used by C4.5. It's Time Complexity is less than C4.5 and the output decision tree generated is more or less similar to the one generated by C4.5 classifier in terms of either structure or prediction accuracy or both. Second heuristic function gives more importance to realistic attributes and thus is more real and gives more accurate and reasonable results. Although it takes more time to build but prediction accuracy

`

comes out to be more than C4.5 algorithm. I establish my claims by implementing these algorithms on four case studies. These algorithms have been implemented using JAVA and have been tested on wide range of data. First case study is a very popular example. It's from a weather domain which predicts whether to go for play or not. There are total four attributes outlook, temperature, humidity and windy. These attributes decide the value of the 5th attribute i.e. decision. Decision attribute can take either of the two values "yes" which means go for play and "No" which means don't go for play. Second example is a scheme of student evaluation that can help the universities (at the time of counseling) to judge whether the student matches the offered program. There are total 21 attributes (student + branch attributes) that decide the value of 22nd attribute "satisfaction level".We take the student attributes and combine them with branch attributes and based on the historical data, satisfaction level of student for that branch is calculated. Third example is another case study. This system predicts the grade of a student for a particular subject by constructing a decision tree on the training set. There are around 18 parameters (student + subject attributes) that decide the grade of a student for a particular subject. Last two examples are much more complex examples than the first one. The last case study is based on the real AIEEE data containing instances in the range of thousands. Here we are predicting student's eligibility to attend AIEEE counselling i.e. whether student is eligible to attend AIEEE Counselling or not. The Decision regarding student's eligibility is estimated based on the attributes like marks in test, All India rank, state rank, category ,date of birth etc.

1.4 Scope of Work

We have mainly focused on 2 performance measures 1) Time to build the tree, 2) Prediction accuracy. We have tried to improve these two things. These improvements can have a significant impact on the practical applications. We want the practical applications to be solved more efficiently and effectively. In future we shall be developing a generic tool for tree based classification algorithms where user can select an appropriate algorithm for its application, depending upon its need in terms of prediction accuracy or time complexity. For example, if user is working on an application where results need to be generated faster, then it can select Algorithm1 or if the user is working on some critical application where results of classification need to be more accurate, then it can select algorithm 2.

1.5 Organization of Thesis

Remainder of the thesis is organized as follows:

Section 2 explains the basic concepts of data mining which includes the definition of data mining and brief explanations of data mining techniques. Then there is an introduction to a specific type of classification called Decision Tree learning. The subsection corresponding to it incorporates the definition of Decision Tree, how decision trees can be used for classification, different types of decision tree, decision tree concepts, detailed explanations of various decision tree algorithms like CART, ID3, C4.5 and improvements proposed by researchers which include

`

dealing with continuous attributes, improvement on information gain, decision tree pruning, improved Decision Tree Based Weights, Improvement by introducing a Balanced Coefficient, Weighted and simplified Entropy and Improvement of Information Gain Formula. Then comes the advantages and disadvantages of decision trees and software and tools used for decision tree learning.

Section 3 contains the proposed taxonomy of tree based classification algorithms i.e. introduction two new heuristic functions to form improved algorithms. First heuristic function is the simplified Measure of Disorderrness to reduce the Time Complexity and second is more realistic heuristic Function that leads to better prediction accuracy. These are analysed in greater depth and there performance is compared with C4.5 algorithm.

Section 4 explains four case studies(examples) which are taken to prove the effectiveness of algorithms. First case study is a very simple example. It's from a weather domain which predicts whether to go for play or not. There are total 4 attributes outlook, temperature, humidity and windy. These attributes decide the value of the 5th attribute i.e. decision. Decision attribute can take either of the two values "yes" which means go for play and "No" which means don't go for play. Second example is a scheme of student evaluation that can help the universities (at the time of counseling) to judge whether the student matches the offered program. There are total 21 attributes (student + branch attributes) that decide the value of 22nd attribute "satisfaction level".We take the student attributes and combine them with branch attributes and based on the historical data, satisfaction level of student for that branch is calculated. Third example is another case study. This system predicts the grade of a student for a particular subject by constructing a decision tree on the training set. There are around 18 parameters (student + subject attributes) that decide the grade of a student for a particular subject. The last case study is based on the real AIEEE data containing instances in the range of thousands. Here we are predicting student's eligibility to attend AIEEE counselling i.e. whether student is eligible to attend AIEEE Counselling or not. The Decision regarding student's eligibility is estimated based on the attributes like marks in test, All India rank, state rank, category ,date of birth etc. Algorithms are applied on each case study/example and results are compared.

Section 5 consists of implementation details which include tools and software used to develop the applications, relations used, steps to create the data source and the program architecture. The program architecture includes the explanation of modules, top-down design and snapshots.

Section 6 consists of conclusion and future scope of the research this is done on C4.5algorithm

Section 7 consists of references which include research papers and web links.

Section 8 consists of appendixes .It contains the large sized decision trees which are generated by the algorithms on their application on case studies (examples).

`

2. DECISION TREE ALGORITHMS IN DATA MINING AND RELATED RESEARCH WORK

2.1 Data Mining Basic Concepts

2.1.1 What is Data Mining?

Data Mining (DM), [6] can simply be explained as an automated process of discovering unanticipated knowledge from massive amount of data. Data Mining involves complex Data Structures, Algorithms, Statistics and Artificial Intelligence.[6] It also includes learning from previous knowledge and recognizing hidden data pattern and providing the realistic results along with rationalization. Knowledge Discovery in Database also known as KDD a synonym of Data Mining, which comprises of three stages [6]:

- The understanding of business and data.
- Performing the pre-processes tasks.
- Data Mining and Reporting

.2.1.2 Data Mining Background, Research and Evolution

In today's world, the increasing processing power and sophisticated technologies has increased the business need, and now people expect more from systems [6]. These days, the computer systems are not only used for storing data but also for providing information and forecasting. Data Mining is part of a word, which has been recently introduced known as BI or Business Intelligence [6]. The need is to derive knowledge out of the abstract data. With recent technical advances in processing power, memory, interconnectivity data mining is seen as an increasingly important tool by modern business to transform abstract data into business intelligence form giving an additional advantage [6].

The rise of data mining originated from the emergence of data warehouse[7]. As early as 1990s, in "Building the Data Warehouse", William H. Inmon--the U.S.information engineering [7] professional introduced the concept and implementation steps of data warehouse. The concept is that. Data Warehouse is a subject oriented, integrated, nonvolatile, and time variant collection of data, in support of management's decision-making[7]. In his monograph, he divided the implementation steps into three major parts: data preprocessing, data mining and KDD.

`

After 20 years development, while confirming the accuracy of the definition, a number of applications have further confirmed the essence of “Building the Data Warehouse” Data processing is the core to establish data warehouse; Data mining is regarded as the important technology and method [7]; KDD is regarded as the important process and method of data warehouse and data mining is one important step of KDD. From the above, we can see that the major technology and core to establish a complete data warehouse is whether the data mining model system is in common use[7].

Data mining application are characterized by the ability to deal with the explosion of business data and accelerated market changes, these characteristics help providing powerful tools for decision makers, such tools can be used by business users (not only statisticians) for analyzing huge amount of data for patterns and trends [8]. Consequently, data mining has become a research area with increasing importance and it involved in determining useful patterns from collected data or determining a model that fits best on the collected data [8]. Different classification schemes can be used to categorize data mining methods and systems based on the kinds of databases to be studied, the kinds of knowledge to be discovered, and the kinds of techniques to be utilized [8].

Data mining techniques used in business-oriented applications are known as Business intelligence (BI) [8]. BI is a general term to mean all processes, techniques, and tools that gather and analyze data for the purpose of supporting enterprise users to make better decisions [8]. The difficulty of discovering and deploying new knowledge in the BI context is due to the lack of intelligent and complete data mining system [8]. The measure of any business intelligence solution is its ability to derive knowledge from data. The challenge is met with the ability to identify patterns, trends, rules, and relationships from volumes of information which is too large to be processed by human analysis alone [8].

2.1.3 Introduction to Data Mining Techniques

The most important data mining techniques are as follows

2.1.3.1 Clustering

Clustering is a process of partitioning a set of data (or objects) in a set of meaningful sub-classes, called cluster. Cluster is a collection of data objects that are similar to one another and thus can be treated collectively as one group but as a collection, they are sufficiently different from other groups. Clustering is a unsupervised classification which means we do not know the class labels and may not know the number of classes. Clustering has wide applications in Pattern Recognition, Spatial Data Analysis, Image Processing, Market Research, Information Retrieval, Web Mining etc.

2.1.3.2 Association

`

Association in data mining is a method for discovering interesting relations between variables in large databases. In other words it is analyzing and presenting strong rules discovered in databases using different measures. For example, the rule {milk, biscuits}---->{bread} found in the sales data of a supermarket would indicate that if some body buys milk and biscuits together, then he or she is more likely to also buy bread too. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example association rule mining also finds its use in Web usage mining, intrusion detection and bioinformatics.

2.1.3.3 Classification

- **It involves categorizing data in different classes**
    - A model is first created based on the set of data.
    - The model is then used to classify new data
    - Given the model, a class can be predicted for new data

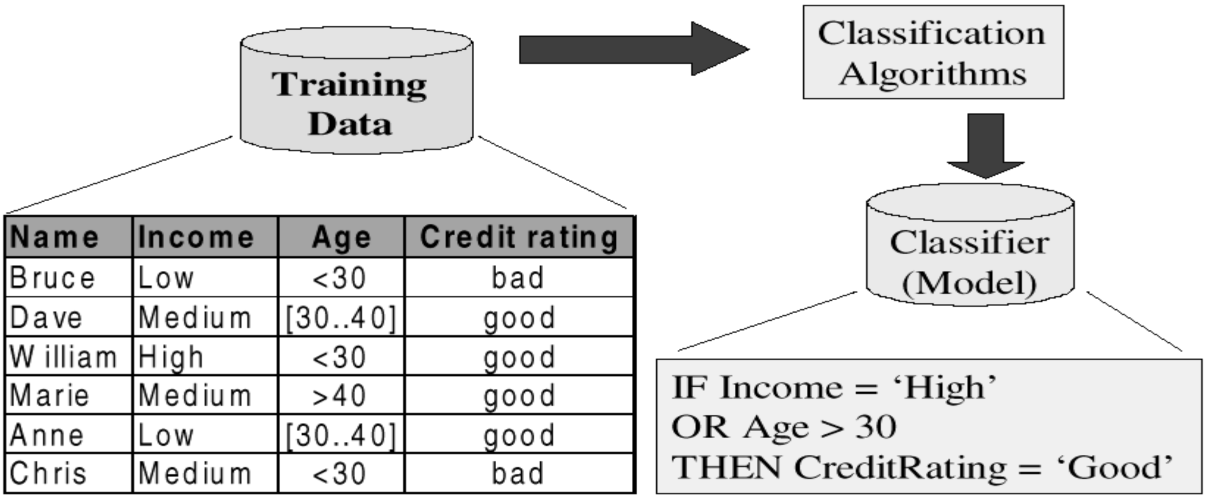
Classification: 3 Step Process [9]

1. **Model construction (Learning):**
    - Each record or instances is associated with a class which determined by one of the attributes, called the classifying attribute.
    - The model is constructed by applying a classification algorithm on the set of data.
    - The set of all records used to construct the model is called training set
    - The model is in the form of either if-then rules or decision trees
2. **Model Evaluation (Accuracy):**
    - The prediction accuracy of the model is measured by applying test data on it.
    - The classifying attribute of test sample is compared with the classified result from the model
    - Prediction accuracy is defined as the percentage of test set samples correctly classified by the model
    - Test set can be independent of training set or it can be dependent on training set if cross validation testing is used.

`

3. **Model Use (Classification):**
    - The model is used to classify new instances which are unclassified i.e. their classifying attribute value is missing.
    - The model is used to predict the value of an attribute.

Model Construction

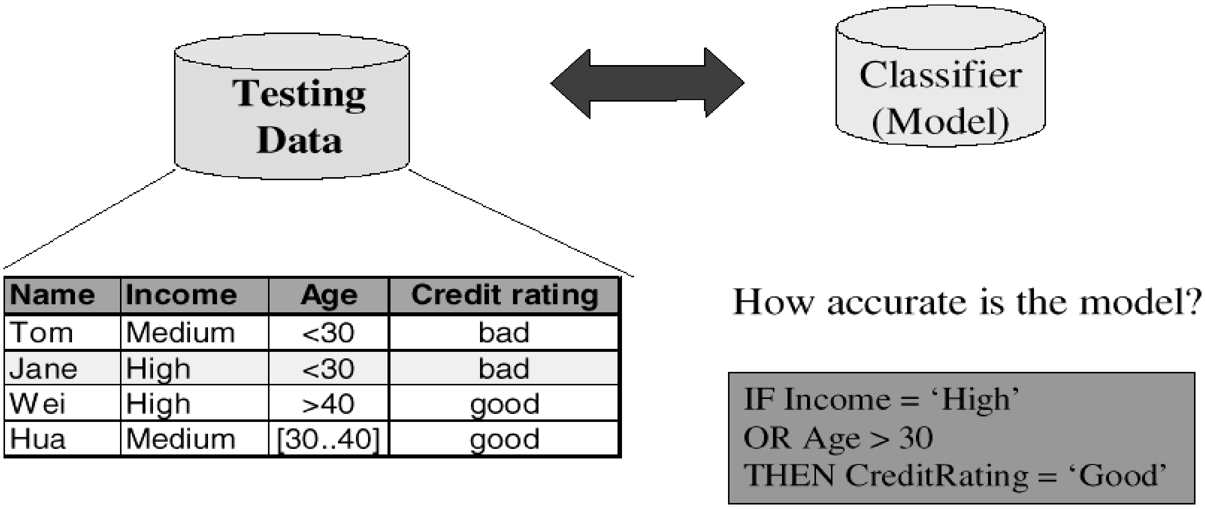


Name	Income	Age	Credit rating
Bruce	Low	<30	bad
Dave	Medium	[30..40]	good
William	High	<30	good
Marie	Medium	>40	good
Anne	Low	[30..40]	good
Chris	Medium	<30	bad

IF Income = 'High'
OR Age > 30
THEN CreditRating = 'Good'

Figure 1-Model Construction (step 1 of classification)

Model Evaluation



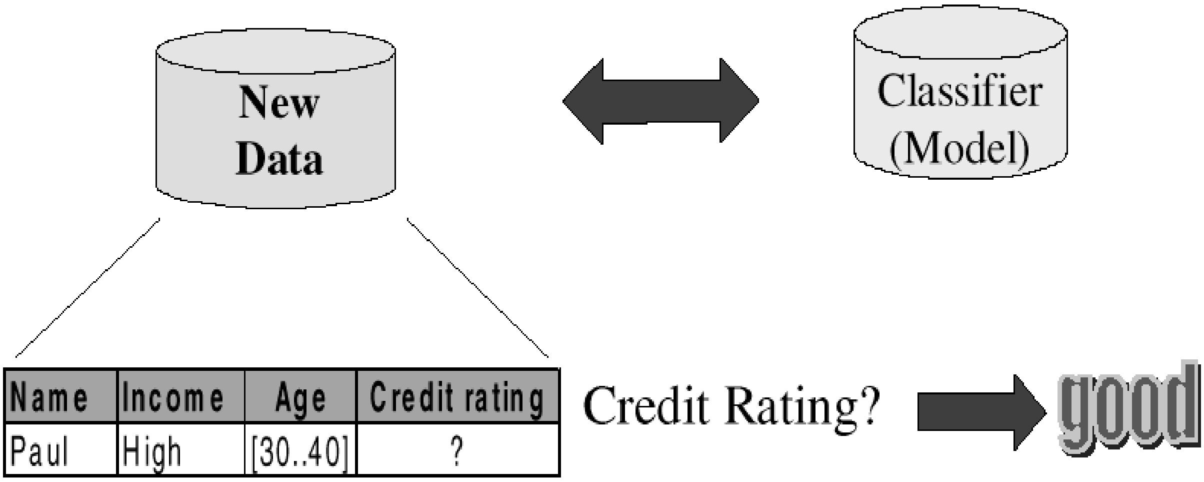
Name	Income	Age	Credit rating
Tom	Medium	<30	bad
Jane	High	<30	bad
Wei	High	>40	good
Hua	Medium	[30..40]	good

IF Income = 'High'
OR Age > 30
THEN CreditRating = 'Good'

Figure 2-Model Evaluation (step 2 of classification)

Model Use: Classification

`



Name	Income	Age	Credit rating
Paul	High	[30..40]	?

Figure 3-Model Use (step 3 of classification)

There are many classification algorithms out of which the most popular ones are decision tree based classification algorithms. This type of classification is easy to understand and most commonly used in real world applications.

2.2 Decision Tree Based Classification

This section contains definition of decision trees, how decision trees can be used for classification, different types of decision tree, decision tree concepts, detailed explanations of various decision tree algorithms like CART, ID3, C4.5 and improvements proposed by researchers which include dealing with continuous attributes, improvement on information gain, decision tree pruning, improved Decision Tree Based Weights, Improvement by introducing a Balanced Coefficient, Weighted and simplified Entropy and Improvement of Information Gain Formula. Then we discuss the advantages and disadvantages of decision trees and software and tools used for decision tree learning.

2.2.1 What are Decision Trees?

- **Decision trees can be described by following characteristics**
    - Internal node represents an attribute.
    - Branch represents a value of an attribute.
    - Leaf node represents a class of a classifying attribute.
    - Decision tree is traversed from top to bottom by performing test on each internal node that comes in the way until the leaf node is encountered.

`

- If attribute is continuous rather than discrete then a threshold is formed and tests are performed on that threshold value.
- Leaf node contains records belonging to the same class.
- A sample decision looks like:

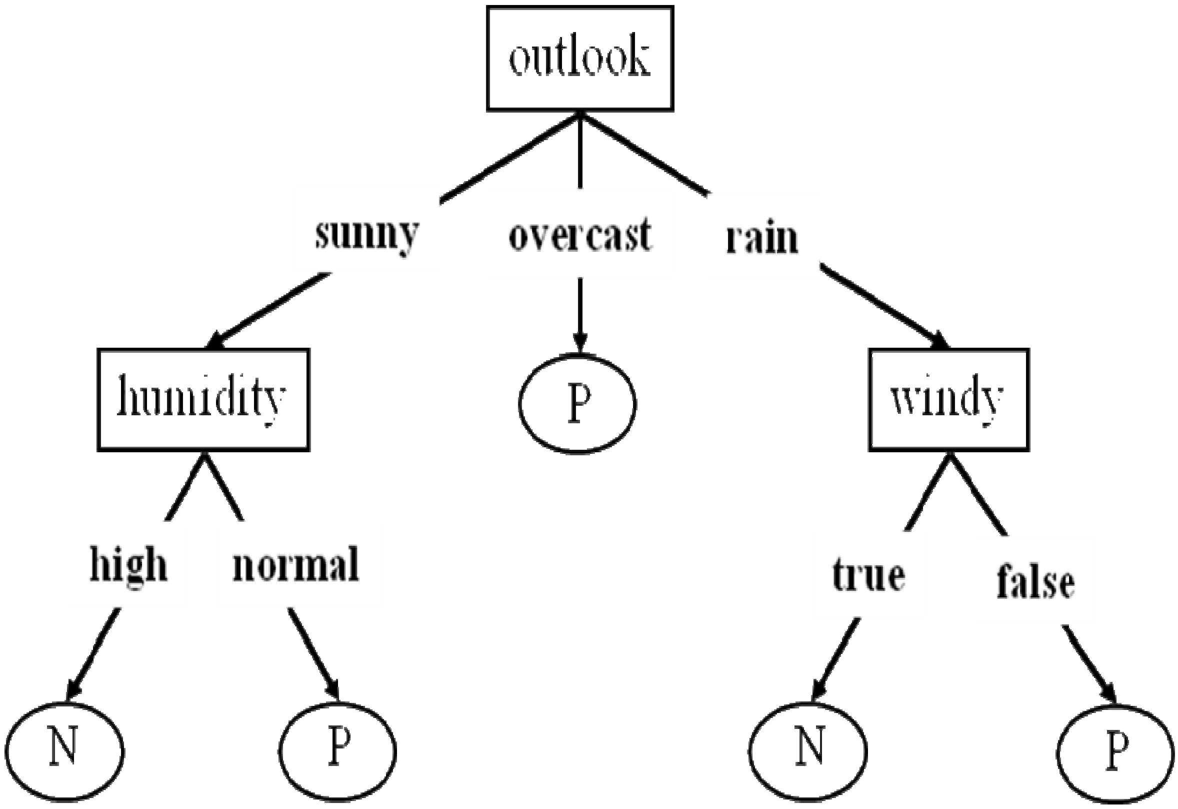
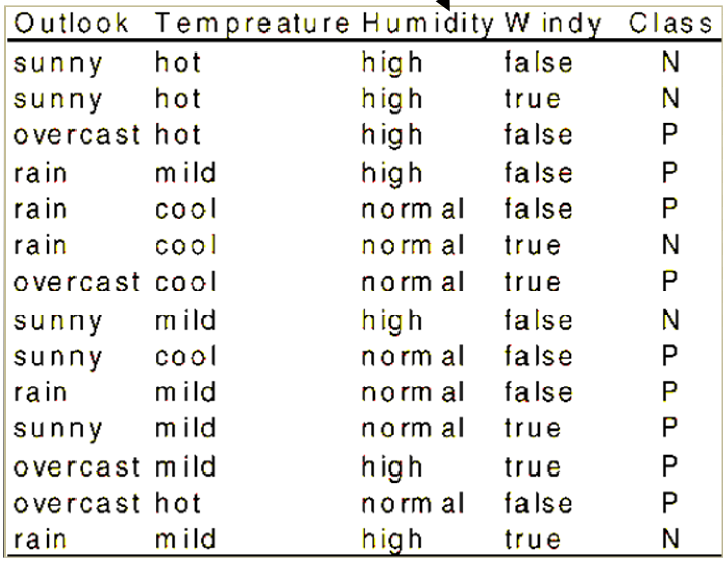


Figure 4-A Sample Decision Tree

- **Example: Whether to go for play or not?**
  - a set of attributes and their possible values:

➢ outlook	sunny, overcast, rain
➢ temperature	cool, mild, hot
➢ humidity	high, normal
➢ windy	true, false

Each instance in the training set is associated with a class. "N" stands for not play and "Y" stands for play.



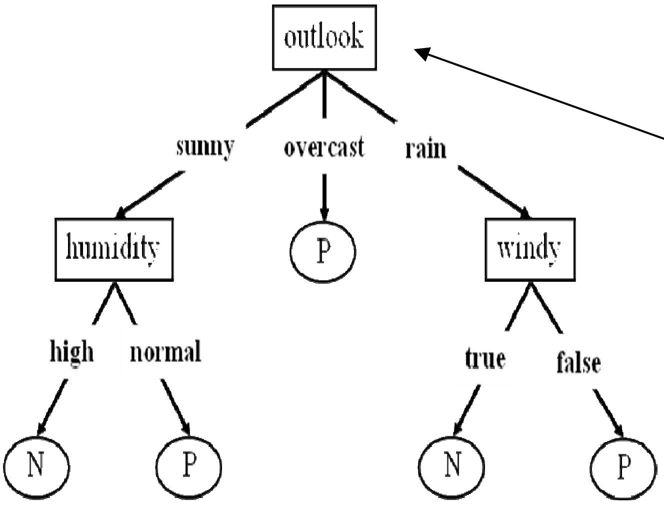
Outlook	Tempreature	Humidity	W indy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	norm al	false	P
rain	cool	norm al	true	N
overcast	cool	norm al	true	P
sunny	mild	high	false	N
sunny	cool	norm al	false	P
rain	mild	norm al	false	P
sunny	mild	norm al	true	P
overcast	mild	high	true	P
overcast	hot	norm al	false	P
rain	mild	high	true	N

Figure 5-A Sample Training Data

2.2.2 Using Decision Trees for Classification [21]

Example:

A new instance with class missing looks like: (rainy, hot, normal, true, ?)



Traverse the decision tree from top to bottom and test each attribute value that comes in the way against that of a new instance

So a new instance: (rainy, hot, normal, true, ?) Is classified as "N" i.e. no play

Figure 6-Example-Using Decision Tree for classification

`

2.2.3 Types of Decision Trees

There are two types of Decision Trees:

- **Classification tree** is used to predict a class for a new instance
- **Regression tree** is used to predict a real number.

2.2.4 Concepts

Decision Tree Algorithms use either of the two concepts or metrics or heuristic functions given below. These heuristic functions define an attribute selection criterion.

2.2.4.1 Gini Impurity

It is a very popular criterion for attribute selection. [11] Let estimated probability that a random item is in class Ci be P(Ci).Thus the estimated probability of misclassification under this rule is Gini Index [11] which is given as:

$$GINI(T)=\sum_{j=1} \sum_{i=1\neq j} P(Cj)*P(Ci)$$

It can also be written as:

$$GINI(T)=1-\sum_{i=1}^{2} P(Ci)$$

According to the formula, the attribute with best split has the minimum value for the Gini Index.

2.2.4.2 Information Entropy

Information entropy is the phenomena to calculate the disorderness in probability distribution..Information entropy is widely used as the measure of disorderness. It is given as

$$E=-\sum_{i} pi*log(pi)$$

Example: A Fair Dice

Probability of each outcome of a dice is same:

P(1)=P(2)=P(3)=P(4)=P(5)=P(6)=1/6

`

This is the case of uniform probability distribution and maximum uncertainty. Maximum uncertainty occurs when probability distribution is uniform.

So for this case Entropy is equal to 1.

Another Example: Unfair Dice

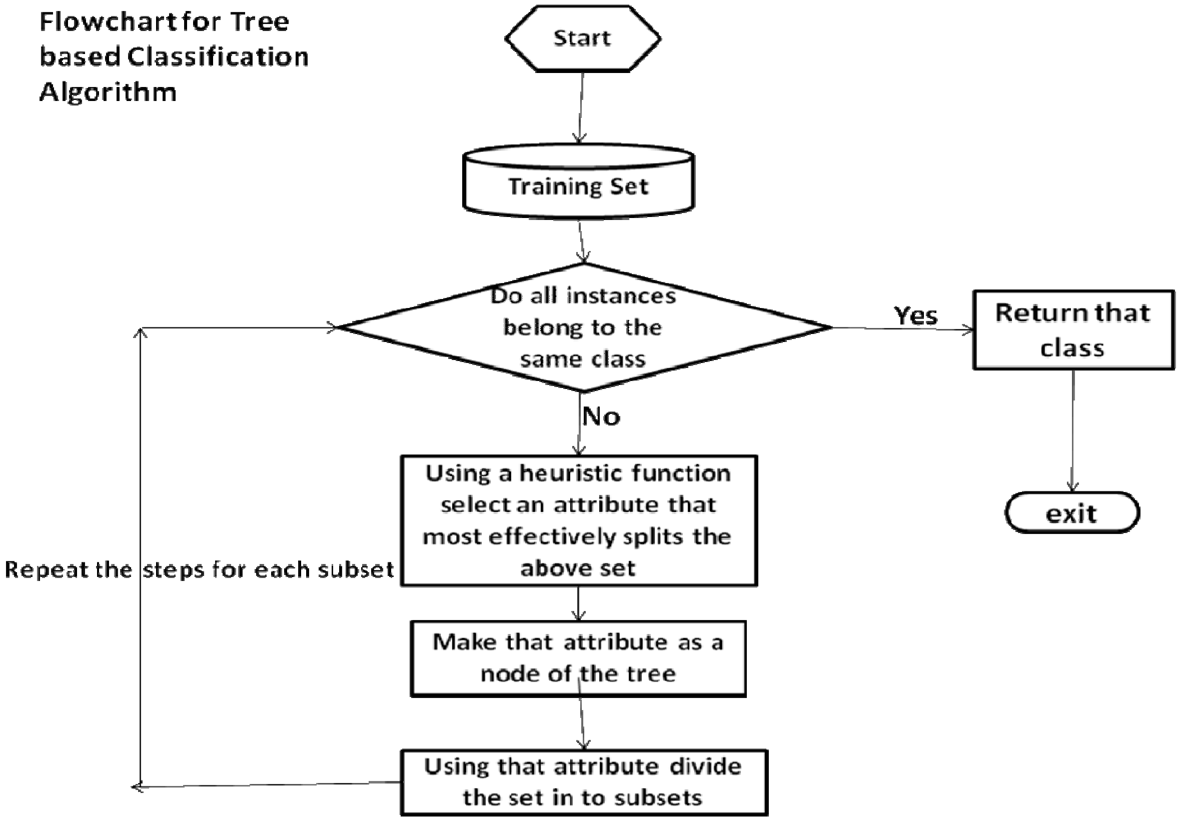
P(1)=1, p(2) =P(3)=P(4)=P(5)=P(6)=0

For this case Entropy is equal to 0

In this way the concept of information entropy is used to measure the degree of disorderness. For more uniform probability distribution, entropy is high and for non uniform probability distribution, entropy is low. The attribute with best split has the minimum value for the Information Entropy.

2.2.5 Decision Tree Algorithms

There are many tree based classification algorithms which we will discuss in this section. Some basic steps of a generic tree based classification algorithm are shown in a flow chart given below:



Flowchart for Tree based Classification Algorithm
Start
Training Set
Do all instances belong to the same class
Yes
Return that class
exit
No
Using a heuristic function select an attribute that most effectively splits the above set
Repeat the steps for each subset
Make that attribute as a node of the tree
Using that attribute divide the set in to subsets

Figure 7- Flow Chart for a Generic Tree Based Classification Algorithm

`

2.2.5.1 Early approaches for Decision Tree Algorithms

2.2.5.1.1 CART

CART is based on the concept of Gini Impurity. Classification and regression trees (CART) is a tree based classification algorithm that either produces classification tree or regression tree depending upon weather the class attribute is a discrete attribute or continuous attribute. If the classifying attribute is a discrete attribute, it produces classification tree and is classifying attribute is a continuous attribute, it produces regression tree. In the process of building a tree, it looks for the best split for the attributes [12].

Decision trees are formed as:

- Rules based on attribute values are selected to get the best split
- Based on the rule i.e selected we divide the given training set in to subsets and move to another level of tree.
- Then we will repeat the same steps on each subset and apply the algorithm recursively. Splitting stops when CART detects terminal node.
- Each branch of the tree ends in a terminal node. Each observation falls into exactly one terminal node which uniquely defined by a set of rules

For a case with a missing attribute value the question is whether this case should be send left or right [12]. The system computes alternative splits that approximate the best split in the sense that the cases are sent the same way. The best split according to the predictive measure [12] is called a surrogate split. If there is not a good candidate among surrogate splits (all of them have not sufficient values of the predictive measure), the case is sent to the child with the largest relative frequency [12].

2.2.5.1.2 ID3 Algorithm

ID3 Algorithm is based on the assumption that a smaller Decision tree with less no. of branches and/or less height or depth has higher prediction accuracy. That is why algorithm emphasizes on building a much smaller decision tree. In an effort to build a small decision tree, it selects that attribute which brings us much closer to the final classification. According to the algorithm the attribute that results in a maximum disordered state is the best attribute, the one that brings us closest to the final classification. To find out that attribute, it uses the concept of information entropy (or information gain)

The ID3 algorithm is as follows:

1. Take each attribute and calculate information entropy or information gain corresponding to it.
2. Select that attribute for which information entropy is minimum or information gain is maximum

`

3. Make node containing that attribute and on the basis of that attribute divide the given training set in to subsets and move to another level of tree.
4. Repeat the same steps on each subset until the set contains instances of the same class.

The ID3 algorithm in pseudo code [5] is as follows

```
ID3 (Examples, Target_Attribute, Attributes)
Start
Create a root node for the tree
If all examples are positive, Return the single-node tree Root, with label = +.
If all examples are negative, Return the single-node tree Root, with label =-.
If number of predicting attributes is empty, then Return the single node tree Root, with label = most common value of the target attribute in the examples.
Otherwise Begin
A = The Attribute that best classifies examples.
Decision Tree attribute for Root = A.
For each possible value, vi, of A, Add a new tree branch below Root, corresponding to the
Test A = vi.
Let Examples (vi), be the subset of examples that have the value vi for A
If Examples (vi) is empty Then below this new branch add a leaf node with label =Most common target value in the examples
Else below this new branch add the sub tree
ID3(Examples (vi), Target_Attribute, Attributes - {A})
Return Root
End
```

The ID3 metrics

Information Entropy

It is defined as the measure of disorderness. Information entropy is used by ID3 algorithm to find out how much disorderness or non uniformness an attribute is producing. It is given as:

$$E(V)=-\sum_{i} p(vi).\log(p(vi))$$

Where:

- E(V) is the information entropy of an attribute V. ;
- vi is the ith value of an attribute V. V can have n values.
- P(vi) is the occurrence probability of V=vi in a set T.

The idea is to select an attribute with best spilt. The attribute with best split has the minimum value for the Information Entropy.

`

Gain

The gain produced by a split over an attribute:

$$I(C,V)=E(C)-\sum_{i} p(vi)*E(C_{vi})$$

Where:

- I(C,V) is the information gain i.e produced by spilting the data set on basis of attribute V.
- E(C) is the entropy of information on set T.
- vi is the ith value of an attribute V. V can have n values.
- P(vi) is the occurrence probability of V=vi in a set T.
- E(Cvi) is the entropy of information on set T(V=vi).

The attribute with best split has maximum information gain.

2.2.5.2 C4.5 – A Very Popular Decision Tree Algorithm Used For Classification

2.2.5.2.1 Exploring C4.5 Algorithm

C4.5 is used to generate a decision tree which is used for classification (a pre-processing step of data mining).It is a statistical classifier based on the concept of information entropy. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets. Its criterion is the normalized information gain ratio in choosing an attribute for splitting the data. Let T be the training set consisting of various instances, |T| be the no. of examples in T and (c1,c2,c3...cn) be the set of classes or categories for an attribute whose value has to be predicted.

2.2.5.2.1.1 Algorithm Detailed Explanation (With Flowchart)

Step 1. Select any attribute V except that attribute whose value has to be predicted. Let v1,v2,v3..vn are the values of V. Let Vn be the subset of T such that V=vn and |Vn| be the no. of instances in Vn.

Step 2. Calculate Information Gain ratio for that attribute. Steps for calculating the value of **Heuristic Function** (information Gain ratio) are shown below:

I. Occurrence probability of category Cj is:

P(Cj)=| Cj| / | T |

II. Occurrence probability of property V = vi is

P(vi) =| Ti | / | T |

III. Conditional probability of category Cj given V = vi is:

P(Cj| vi) =| Cjv| / | Ti |

`

IV. Entropy of information:

$$E(C)=-\sum_{i}P(Cj).log(P(Cj)$$

V. Conditional entropy:

$$E(C/V)=-\sum_{i}p(vi)\sum_{j}(p(Cj/vi).log(p(Cj/vi))$$

VI. Information gain

$$I(C,V)=E(C)-E(C/V)$$

VII. Entropy of attribute:

$$E(V)=-\sum_{i} p(vi).log(p(vi))$$

VIII. Information gain ratio:

$$\text{Gain ratio } (v) = I(C,V)/E(V)$$

Step 3. Repeat steps 1 and 2 for each attribute.

Step 4. Then select an attribute for which entropy is minimum (or, equivalently, information gain ratio is maximum)

Step 5. Make node containing that attribute.

Step 6. Then on the basis of that attribute, divide the given training set in to subsets.

Step 7. Then recursively apply the algorithm on each subset until the set contains instances of the same class. If the set contains instances of the same class, then return that class.

Flow chart for heuristic function used in this algorithm is given as.

`

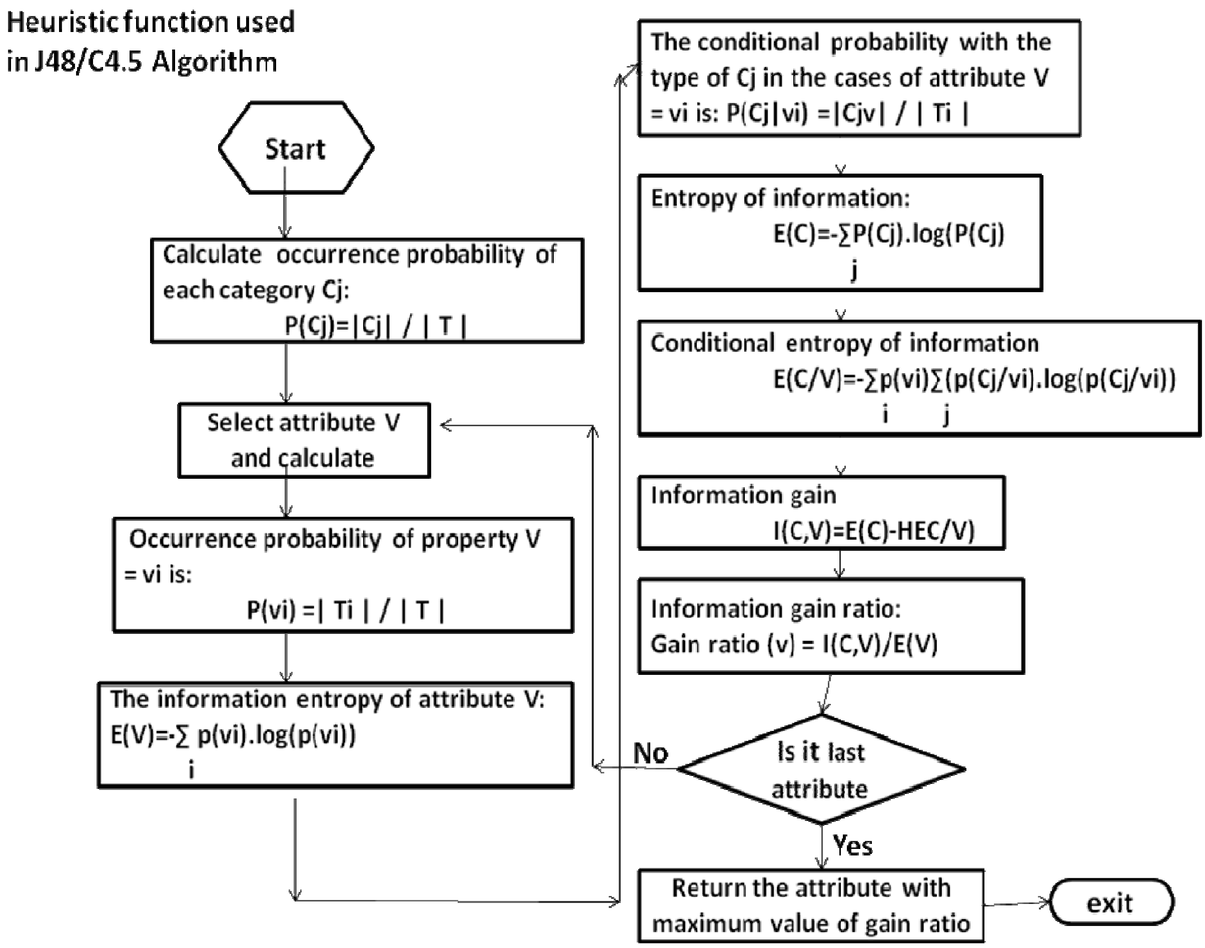

Heuristic function used in J48/C4.5 Algorithm
Start
Calculate occurrence probability of each category Cj: P(Cj)=|Cj| / | T |
Select attribute V and calculate
Occurrence probability of property V = vi is: P(vi) =| Ti | / | T |
The information entropy of attribute V: E(V)=-∑ p(vi).log(p(vi)) i
The conditional probability with the type of Cj in the cases of attribute V = vi is: P(Cj|vi) =|Cjv| / | Ti |
Entropy of information: E(C)=-∑P(Cj).log(P(Cj) j
Conditional entropy of information E(C/V)=-∑p(vi)∑(p(Cj/vi).log(p(Cj/vi)) i j
Information gain I(C,V)=E(C)-HEC/V)
Information gain ratio: Gain ratio (v) = I(C,V)/E(V)
Is it last attribute
No
Yes
Return the attribute with maximum value of gain ratio
exit

Figure 8-Flow Chart for a Heuristic Function Used in C4.5 Algorithm

2.2.5.2.1.2 Example-A case study (To explain the Algorithm)

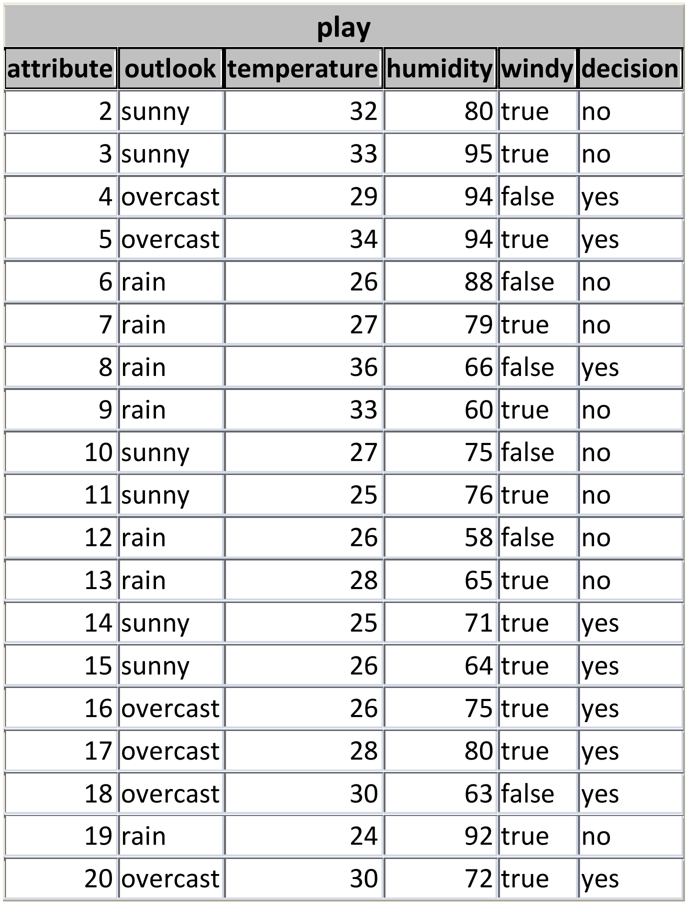
Table 1 below contains a data set of related indicators according to the weather conditions to decide whether to go for play or not. There are total 4 attributes outlook, temperature, humidity and windy. These attributes decide the value of the 5th attribute i.e. decision. Decision attribute can take either of the 2 values “yes” which means go for play and “No” which means don’t go for play.

Make the sample data set as the training set now and construct a decision tree using an algorithm given below.

Table 1- UCI weather Data set- Example to explain C4.5 Algorithm

play					
attribute	outlook	temperature	humidity	windy	decision
1	sunny	35	75	false	no

`



play					
attribute	outlook	temperature	humidity	windy	decision
2	sunny	32	80	true	no
3	sunny	33	95	true	no
4	overcast	29	94	false	yes
5	overcast	34	94	true	yes
6	rain	26	88	false	no
7	rain	27	79	true	no
8	rain	36	66	false	yes
9	rain	33	60	true	no
10	sunny	27	75	false	no
11	sunny	25	76	true	no
12	rain	26	58	false	no
13	rain	28	65	true	no
14	sunny	25	71	true	yes
15	sunny	26	64	true	yes
16	overcast	26	75	true	yes
17	overcast	28	80	true	yes
18	overcast	30	63	false	yes
19	rain	24	92	true	no
20	overcast	30	72	true	yes

Step 1. To calculate the occurrence probability for each category and then entropy of information

Pc[yes]= 0.45 pc[No]= 0.55 E[C]= 0.68

Step 2. To calculate the information gain ratio for each attribute

i. Outlook

- Pv[sunny]=0.35
- pv[overcast]= 0.3
- pv[rain]= 0.35
- E[v]= 1.096067
- E(C/V)= 0.3529
- I(C,V)= 0.335203746

`

- Gain Ratio(V)= 0.30582404

ii. Temperature

- Pv[<]=0.7
- pv[>]= 0.3
- E[v]= 0.6108
- E(C/V)= 0.676
- I(C,V)= 0.011
- Gain Ratio(V)= 0.01961

iii. Humidity

- Pv[<]=0.7
- pv[>]= 0.3
- E[v]= 0.673
- E(C/V)= 0.680
- I(C,V)= 0.0076
- Gain Ratio(V)= 0.01132

iv. Windy

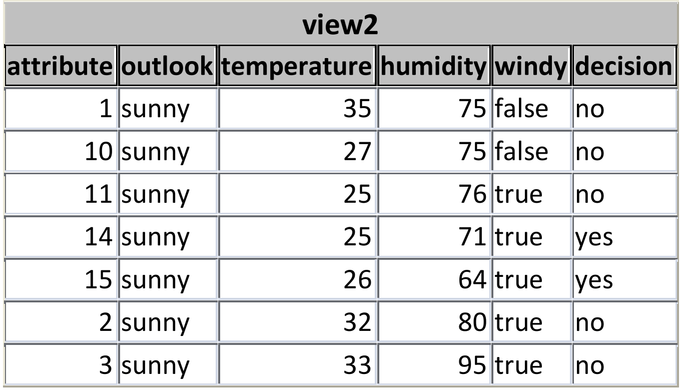
- Pv[true]= 0.65
- pv[false]= 0.35
- E[v]= 0.647
- E(C/V)= 0.68763
- I(C,V)= 5.00E
- Gain Ratio(V)= 7.7270870839E

Step 3. Then we will select that attribute for which information ratio is maximum. Clearly information gain is maximum for **outlook** attribute.

`

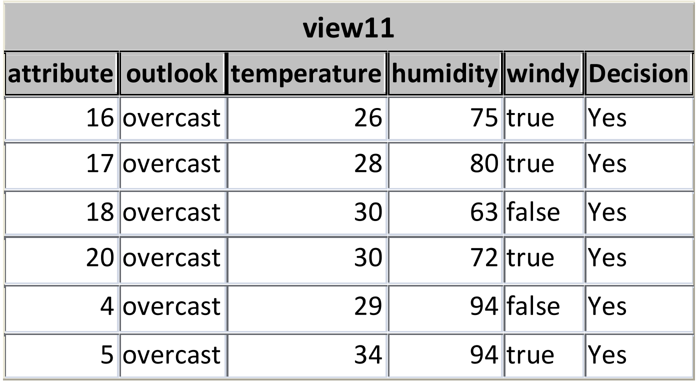
Step 4. Then on the basis of that attribute we will divide the given training set in to subsets and move to another level of tree.

Table 2- Subset outlook=”sunny” (Produced from Table 1)



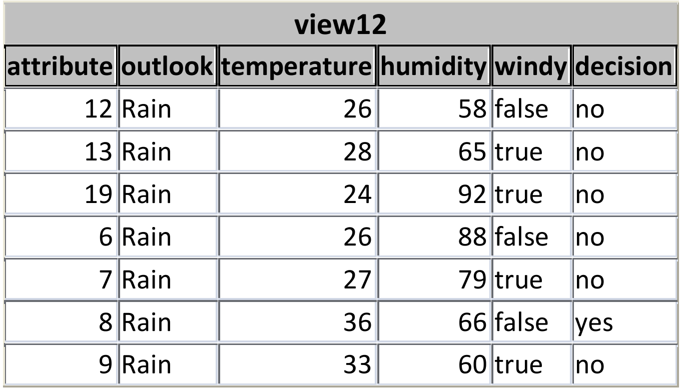
view2					
attribute	outlook	temperature	humidity	windy	decision
1	sunny	35	75	false	no
10	sunny	27	75	false	no
11	sunny	25	76	true	no
14	sunny	25	71	true	yes
15	sunny	26	64	true	yes
2	sunny	32	80	true	no
3	sunny	33	95	true	no

Table 3- Subset outlook=”overcast” (Produced from Table 1)



view11					
attribute	outlook	temperature	humidity	windy	Decision
16	overcast	26	75	true	Yes
17	overcast	28	80	true	Yes
18	overcast	30	63	false	Yes
20	overcast	30	72	true	Yes
4	overcast	29	94	false	Yes
5	overcast	34	94	true	Yes

Table 4- Subset outlook=”rain” (Produced from Table 1)



view12					
attribute	outlook	temperature	humidity	windy	decision
12	Rain	26	58	false	no
13	Rain	28	65	true	no
19	Rain	24	92	true	no
6	Rain	26	88	false	no
7	Rain	27	79	true	no
8	Rain	36	66	false	yes
9	Rain	33	60	true	no

`

Step 5. Then we will repeat the same steps on each subset and apply the algorithm recursively

In this way the final tree generated looks like

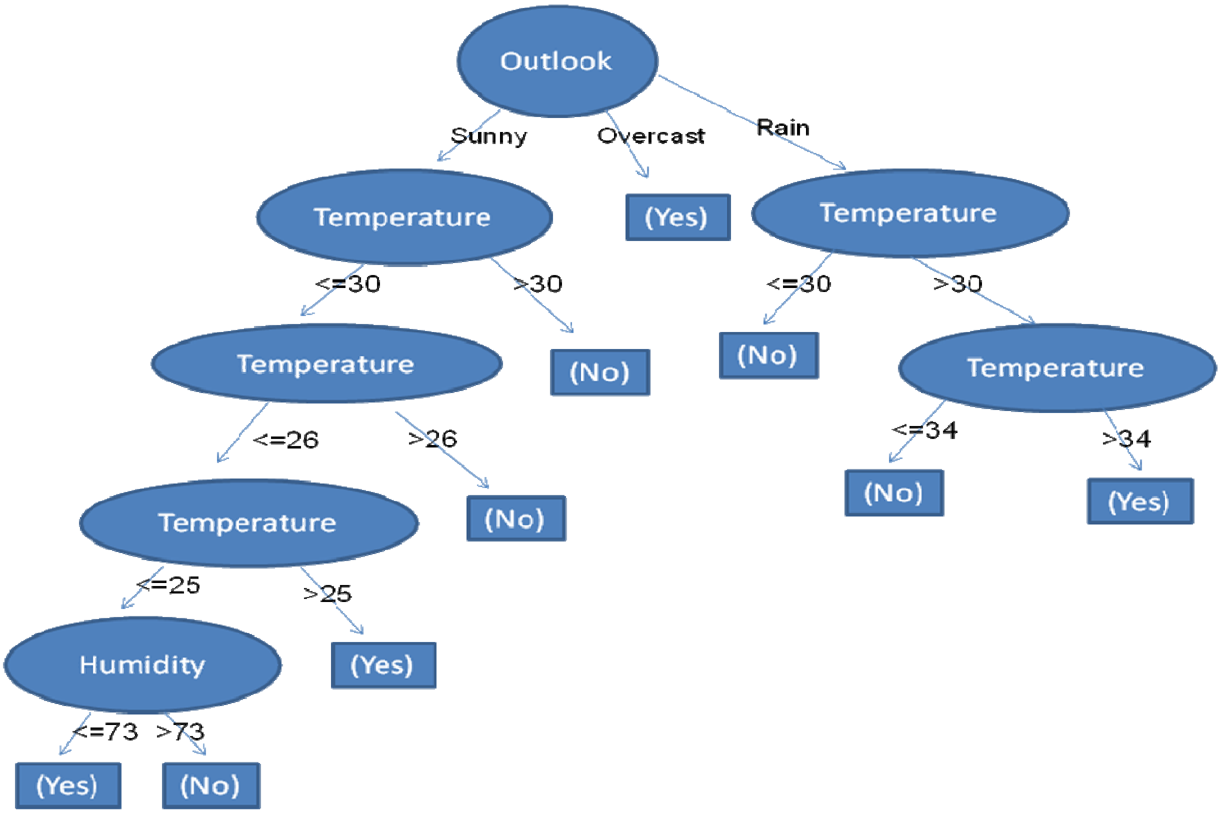


Figure 9-Decision Tree Generated on applying C4.5 Algorithm on Weather Training data

2.2.5.2.2 Improvements made in C4.5 from ID3

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm.

Quinlan himself proposed some improvements over an ID3 algorithm [3,22]. Firstly he proposed an improved method to handle continuous attributes [3, 22]. He offered to select a threshold on the values of attributes and then divide the given set of training examples in to two subsets [3,22]. One set contains attribute values greater than the threshold and other set contains attributes values less than or equal to the threshold [3,22]. Secondly he suggested to not to use missing attribute values in the calculation of information gain ratio [22]. Thirdly he proposed the method of pruning [22]. It is the process of removing unwanted sections of the tree which are generated due to noise or too small set of training data or large no. of parameters/attributes [22]. This is done to improve prediction accuracy [22].

`

2.2.5.2.2.1 Dealing with continuous variables

Several attempts have been made by researchers to improve the existing C4.5 algorithm. Quinlan himself proposed some improvements over an ID3 algorithm [3,22]. Firstly he proposed an improved method to handle continuous attributes [3,22]. He offered to select a threshold on the values of attributes and then divide the given set of training examples in to two subsets [3]. One set contains attribute values greater than the threshold and other set contains attributes values less than or equal to the threshold [3,22].

- **Partition continuous attribute into a fixed set of intervals**
  - Sort the data corresponding to the continuous attribute **A**.
  - Identify adjacent data with different classification.
  - generate a set of candidate thresholds midway
  - Disadvantage of this scheme is, it will create too many intervals
- **Another Solution:**
  - Take a threshold M i.e. the no. of partitions.
  - Divide the data in to M no. of partitions.
  - Then find out the majority class of each partition.
  - Then merge adjacent partitions with the same majority class

M=3

Temperature	54	55	58	59	60	61	62	62	65	65	70	71	73	75
Decision	yes	no	yes	yes	yes	no	no	yes	yes	yes	no	Yes	yes	no

After merging same majority class…Final mapping comes out to be

temperature <= 67.5 ==> “yes”; temperature > 67.5 ==> “no”

`

2.2.5.2.2.2 Improving on Information Gain

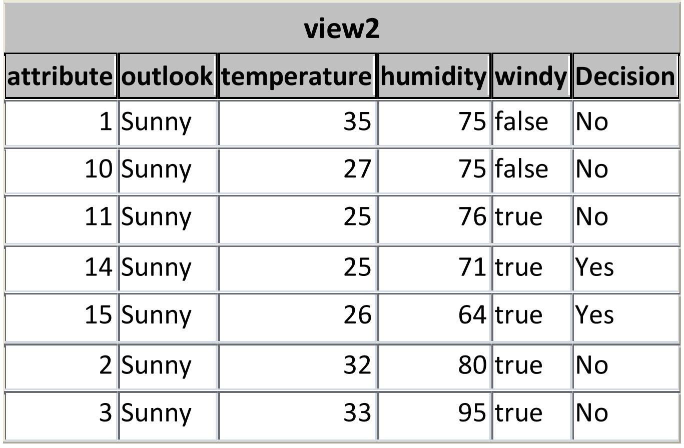
- **Info. Gain comes out to be more for attributes with a large number of values [9,23]**
    - larger distribution ==> lower entropy ==> larger Gain
- **Quinlan suggests using Gain Ratio [9,23]**
    - penalize for large number of values

The information gain ratio is given as

Gain ratio (v) = I(C,V)/E(V)

Example: Information Gain and Gain ratio calculations for the attribute Outlook

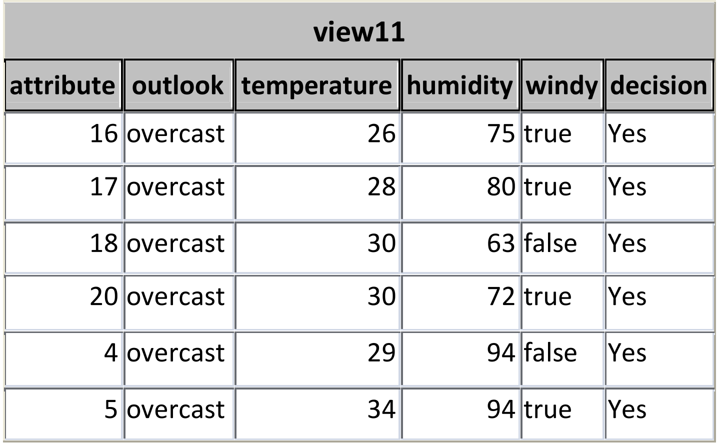
Table 5- Subset outlook="sunny" (Produced from Table 1)



view2

attribute	outlook	temperature	humidity	windy	Decision
1	Sunny	35	75	false	No
10	Sunny	27	75	false	No
11	Sunny	25	76	true	No
14	Sunny	25	71	true	Yes
15	Sunny	26	64	true	Yes
2	Sunny	32	80	true	No
3	Sunny	33	95	true	No

Table 6- Subset outlook="overcast" (Produced from Table 1)

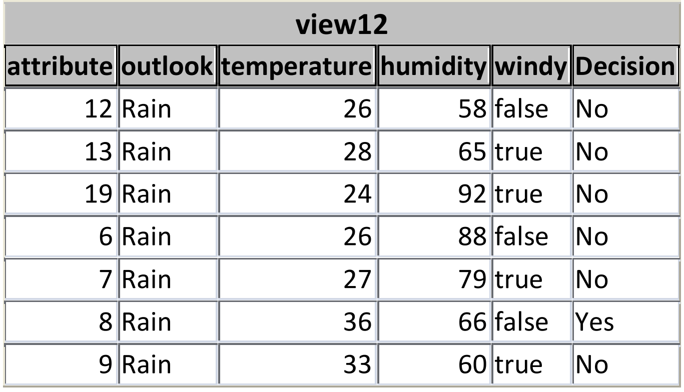


view11

attribute	outlook	temperature	humidity	windy	decision
16	overcast	26	75	true	Yes
17	overcast	28	80	true	Yes
18	overcast	30	63	false	Yes
20	overcast	30	72	true	Yes
4	overcast	29	94	false	Yes
5	overcast	34	94	true	Yes

`

Table 7- Subset outlook=”rain” (Produced from Table 1)



view12

attribute	outlook	temperature	humidity	windy	Decision
12	Rain	26	58	false	No
13	Rain	28	65	true	No
19	Rain	24	92	true	No
6	Rain	26	88	false	No
7	Rain	27	79	true	No
8	Rain	36	66	false	Yes
9	Rain	33	60	true	No

- Pv[sunny]=0.35
- pv[overcast]= 0.3
- pv[rain]= 0.35
- E[v]= 1.096067
- E(C/V)= 0.3529
- **I(C,V)**= 0.335203746
- **Gain Ratio(V)**= 0.30582404

2.2.5.2.2.3 Decision Tree Pruning

- A tree generated may over-fit the training examples due to noise or too small a set of training data. This is called over fitting in classification
- Two approaches to avoid over-fitting:
  - (Stop earlier): Stop growing the tree earlier
  - (Post-prune): Allow over-fit and then post-prune the tree
- Approaches to determine the correct final tree size:
  - Separate training and testing sets
  - or use cross-validation

`

Quinlan proposed the method of pruning [22]. It is the process of removing unwanted sections of the tree which are generated due to noise or too small set of training data or large no. of parameters/attributes [22]. This is done to improve prediction accuracy [22].The C4.5 algorithm gives several options related to tree pruning. Many algorithms attempt to "prune", or simplify, their results. Pruning produces fewer, more easily interpreted results.[10] More importantly, pruning can be used as a tool to correct for potential over fitting. The overall concept is to gradually generalize a decision tree until it gains a balance of flexibility and accuracy.[10]

C4.5 employs two pruning methods. [10]

Bottom-up Pruning

The first is also known as subtree replacement. This means that nodes in a decision tree may be replaced with a leaf -- basically reducing the number of tests along a certain path. This process starts from the leaves of the fully formed tree, and works backwards toward the root. [10]

Top Down Pruning

The second type of pruning used in C4.5 is also termed as subtree raising. In this case, a node may be moved upwards towards the root of the tree, replacing other nodes along the way. Subtree raising often has a negligible effect on decision tree models. There is often no clear way to predict the utility of the option, though it may be advisable to try turning it off if the induction process is taking a long time. This is due to the fact that subtree raising can be somewhat computationally complex.[10]

2.2.5.3 Very Recent Approaches

This section contains vary recent improvements on C4.5 Algorithm proposed by researchers which include improved Decision Tree Based Weights, Improvement by introducing a Balanced Coefficient, Weighted and simplified Entropy and Improvement of Information Gain Formula.

2.2.5.3.1 Improved Decision Tree Based Weights

The intuition for the algorithm proposed in [5] paper is to have different initial weights for every instance to reflect its importance in future prediction. The method is designed to construct decision tree model with the consideration of the weights between different instances. The author tried to show that the algorithm is more advantageous to achieve better classification results, in term of classification accuracy. Algorithm is based on the fact that many real world data is imbalanced in nature [5, 27, 28, 29]. In this paper [5], they introduced a new improved decision tree based weights, which considers imbalanced weights between different instances, to address the class imbalanced problems. They have compared the proposed algorithm with C4.5. If decision tree considers the weight of an instance to reflect its importance in the training and test data sets, the class with high weight is less likely to be misclassification and better classification results will be achieved as a consequence [5, 27, 28, 29]. Therefore, they have introduced a

`

decision tree algorithm based weights by considering unequal distribution between different instances and the instance with high weight will be classified accurately.[5]

Information Gain (IG) is one of the most popular selection criterions to address this issue. The decision tree algorithm usually selects the attribute which has the maximum Information Gain (IG) values until every node of the decision tree contains the instances belonging to only a single class or satisfying some stopping criterions. Generally speaking, the Information Gain (IG) measure, designed as (1), is widely used to select the attribute with the maximum Information Gain (IG) value.

$$IG(A,T) = IE(T) - \sum_{i} (|Ti|/|T|) * IE(Ti) \ldots\ldots\ldots\ldots\ldots\ldots\ldots 1$$

where T is all the training data sets, Ti is the i-th value of attribute A, and |T|stands for the number of instances in data set T and |Ti| represents the number of data set Ti. IE(T) is used to measure the Information Entropy (IE) for attribute A. Assume that pi is the percentage of instances belonging to class i in the data set T. So the Information Entropy (IE) for attribute A is defined as below.

$$IE(T)=-\sum_{i} Pi*\log(pi) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2$$

Gain Ratio is given as:

$$GR(A, T) =IG(A,T)/SI(A,T) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 3$$

where SI (A,T) stands for the Splitting Information (SI) value for attribute A and is defined as (4). Suppose that attribute A has k values, which are denoted as S1, S2, …,Sk . And |Si| stands for the number of instances whose values are Si, i = 1,…, k.

$$SI(A,T)= \sum_{i} (|Si|/|T|)*\log(|Si|/|T|) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 4$$

In this research they have changed the gain ratio

Here, assume that there are n class labels C1, C2 ,…, Cn and m training instances in the imbalanced data sets. mi stands for the number of the instances which belongs to class Ci, where mi=m and i=1, 2,….,n. The weight vector for instances from the imbalanced data is defined as W= (w1, w2…wm). Intuitively, the larger the proportion of certain class in all data sets is, the more possible it is that the instance belongs to this special class. Based this intuition, the weight wj for the j-th instance should be a monotonically increasing variable with the number of this special class, where j = 1,2,…,m. Suppose that the k-th instance belongs to Class Cr and the t-th instance belongs to Class Cs, where k, t=1…….,m and r, s= 1,2….. So, the weight variable should satisfy the constraints listed as below.

`

a) 0<=wj<=1

b) If mr<=ms then wk<=wt

Here, suppose that the j-th instance belongs to Class Ci, and then the weight variable wj for the j-th instance is defined as (5).

$$wj=mi/m\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5$$

After the weight of every instance is determined, the Information Gain without incorporating the weights between different instances will be replaced by the Weight Information Gain (WIG), which is given as (6).

$$WIG(A,T)=WIE(T)-\left(\sum_{i}\left(\sum_{x=ci}Wx/\sum_{j=1}Wj\right)\right)*WIE(Ti)\ldots\ldots\ldots 6$$

2.2.5.3.2 Improvement by introducing a Balanced Coefficient

Another improvement was given by LI Rui, WEI Xianmei and YU Xue-wei [4]. Since C4.5 uses divide and conquer strategy and searches the best node locally, they suggested a method to improve optimality of C4.5 algorithm by proposing a balanced coefficient [4]. Users can decide the value of this balanced coefficient according to the situation and by using their intellectual and domain knowledge [4, 24, 25, 26]. It's an unclear concept and only artificially improves the efficiency of the algorithm.

V, VI, VII, VIII equations of step 2 of the above C4.5 algorithm are shown as:

v. Conditional entropy:

$$E(C/V)=-\sum_{i}p(vi)\sum_{j}(p(Cj/vi).log(p(Cj/vi))$$

vi. Information gain

$$I(C,V)=E(C)-E(C/V)$$

vii. Entropy of attribute:

$$E(V)=-\sum_{i} p(vi).log(p(vi))$$

viii. Information gain ratio:

$$\text{Gain ratio } (v) = I(C,V)/E(V)$$

`

This research proposes a degree of balance coefficient ($0<\pounds<1$), It is a vague concept, and its size is determined by the decision-makers based on a priori knowledge or domain knowledge [4]. The improved C4.5 algorithm aims at the method of rule generation that is attribute selection criteria to improve C4.5 algorithm. By introducing a degree of balance coefficient in formulas (2) and (4), reducing information entropy of certain attributes, and increasing the information entropy of the other attributes accordingly. The end, the decision tree created in a specific environment has a higher accuracy. Now design the degree of balance coefficient of a certain attribute as £ , and then, the formulas V, VI, VII, VIII are modified

v. Conditional entropy:

$$E(C/V_{\pounds})=-\sum_{i}(p(vi)+ \pounds)\sum_{j}(p(Cj/vi).\log(p(Cj/vi))$$

vi. Information gain

$$I(C,V_{\pounds})=E(C)-E(C/V)$$

vii. Entropy of attribute:

$$E(V_{\pounds})=-\sum_{i} (p(vi)+ \pounds).\log(p(vi)+ \pounds)$$

viii. Information gain ratio:

$$\text{Gain ratio } (v) = I(C,V_{\pounds})/E(V_{\pounds})$$

2.2.5.3.3 Weighted and simplified Entropy

The article [2,18,19,20] put forward specific solution for the problems of property value vacancy, multiple-valued property selection, property selection criteria , propose to introduce weighted and simplified entropy into decision tree algorithm so as to achieve the improvement of ID3 algorithm. The experimental results show that the improved algorithm is better than widely used ID3 algorithm at present on overall performance.

Improvement can be shown with the help of an example:

Let I(p,n) be the entropy of a set of instances *I*, containing *p* positive and *n* negative examples

`

$$E(A)=\sum_{i}^{V}\frac{p_i+n_i}{P+N}I(p_i,n_i)$$

$$I(p_i,n_i)=\frac{p_i}{p_i+n_i}\log_2\frac{p_i}{p_i+n_i}-\frac{p_i}{p_i+n_i}\log_2\frac{p_i}{p_i+n_i}$$

And put the above formula into E (A), then,

$$E(A)=\sum_{i}^{V}\frac{p_i+n_i}{P+N}\left(-\frac{p_i}{p_i+n_i}\log_2\frac{p_i}{p_i+n_i}-\frac{p_i}{p_i+n_i}\log_2\frac{p_i}{p_i+n_i}\right)$$

$$=\sum_{i}^{V}\frac{1}{(P+N)\ln 2}\left(-p_i\ln\frac{p_i}{p_i+n_i}-n_i\ln\frac{n_i}{p_i+n_i}\right)$$

Because (P+N) In2 is a constant in training data set, so we could suppose a function S(A) to meet the following formula:

$$S(A)=\sum_{i}^{V}\left(-p_i\ln\frac{p_i}{p_i+n_i}-n_i\ln\frac{n_i}{p_i+n_i}\right)$$

S(A) just contains add, multiply, division operations, so operation time is certainly shot than that of E(A) which contains multiple logarithmic terms.

2.2.5.3.4 Improvement of Information Gain Formula

Another attempt to improve C4.5 algorithm was done by Zhu Xiaoliang, Wang Jian, YanHongcan, WuShangzhu [1]. They analyzed C4.5 algorithm and gave further research to improve its time complexity. Using some laws of mathematics they tried to simplify the formula of calculation of information entropy [1, 15, 16, 17]. They claimed improvement over an existing C4.5 algorithm in terms of time complexity but didn't give any solid justification to improve the prediction accuracy of the algorithm.

Improvement can be shown with the help of an example:

Let I(p,n) be the entropy of a set of instances *I*, containing *p* positive and *n* negative examples

Continuing from the above improvement, the formula could simply to

$$\mathit{Info}(s)=\frac{1}{(n+p)\mathit{In}2}\sum_{i=1}^{n}\left(-n_i\mathit{In}\,\frac{n_i}{n_i+p_i}-p_i\mathit{In}\,\frac{p_i}{n_i+p_i}\right)$$

According to the theory of equivalent infinitesimal of mathematic, if *X* is small, then In (1 + x) approximately = X, so

`

$$In\ \frac{n_i}{n_i+p_i} = In(1-\frac{p_i}{n_i+p_i}) \approx -\frac{p_i}{n_i+p_i}$$

$$In\ \frac{p_i}{n_i+p_i} = In(1-\frac{n_i}{n_i+p_i}) \approx -\frac{n_i}{n_i+p_i}$$

so it could simplify the calculation

$$Info(s) \approx \frac{1}{(n+p)In2}\sum_{i=1}^{n} \frac{2n_i p_i}{n_i+p_i}$$

This greatly reduces the no. of operations.

2.2.6 Advantages of Decision Trees

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

- Decision trees have the ability to handle discrete attributes as well as continuous attributes.
- Unlike other classification techniques which require data normalization, removal of missing attribute values etc, they just require a simple data preparation.
- Decision trees are very easy to understand and visualize.
- Decision tree model can easily be tested and validated. In other words statistical test cases can easily be applied on the decision tree model to check the reliability of the model.
- Decision trees have separate to handle missing attributes.
- Decision trees work well for large amount of data. Large amount of data can be analyzed in lesser time. They are known for their good performance against larger data sets.

2.2.7 Limitations of Decision Trees

- Practical decision-tree learning is based on greedy algorithm where locally optimal decisions are made at each level of the tree. Such algorithms cannot guarantee to return the globally optimal solutions. That is why the problem of learning with Decision tree is NP Completes.
- A tree generated may over-fit the training examples due to noise or too small a set of training data. This is called over fitting in classification.
- There are some limitations of decision tree algorithms proposed by quinlan. First, C4.5 uses a divided strategy, It is the best local search when structure the internal nodes of trees [4] Therefore, the final outcome is still less than optimal results in spite of having been very high accuracy. Second, C4.5 decision-making evaluation is most based on the

`

error rate of decision tree [4]. The depth of the tree and the number of nodes are not considered, and the average depth of the tree correspond directly to the speed of the decision tree forecast, the number of tree nodes represent the size of the tree. Third, while structuring the decision tree, while evaluation. Following structure of the decision tree, it is difficult to re-adjust the structure and content of the tree, it is very difficult to improve the performance of decision trees [4]. Fourth, C4.5 has each test while grouping attribute values without a mechanism for the use of heuristic search and the efficiency of division is lower [4].

2.2.8 Tools and software for Decision Tree classification

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

Weka workbench provides two very common decision tree construction algorithms: ID3 and C.45 (also called j48 classifier).

ID3 is Inductive Logic Programming methods, developed by Quinlan [5], it is an attribute based machine-learning algorithm that creates a decision tree on a training set of data and an entropy measure to build the leaves of the tree.

C4.5 algorithm is based on the ID3, with supplementary programming to address ID3 problems.

2.2.9 Conclusion

The Most of the improvements like weighted and simplified entropy, improvement of information gain formula etc that are discussed above are mathematical improvements where the researchers have tried to simplify the formula of information gain by using some approximations and laws of mathematics. Such improvements do not have any impact on prediction accuracy and also they marginally improve the time complexity of the algorithm. The Miscellaneous improvements like improvement by introducing balanced coefficient, tree based weights etc that are discussed above also have very less impact on time complexity and prediction accuracy. So we conclude that the most of the improvements done on C4.5 Algorithm in the past are marginal improvements in terms of prediction accuracy or time. So I propose two conceptual improvements which have a huge impact on time complexity and prediction accuracy. Next section contains the proposed taxonomy of tree based classification algorithms i.e. introduction two new heuristic functions to form improved algorithms. These heuristic functions are better than the one used by C4.5 Algorithm. First one is better in terms of time taken to build the tree and second one is better in terms of prediction accuracy. Proposed algorithms are analysed in greater depth and there performance is compared with C4.5 algorithm.

`

3. PROPOSED TAXANOMY OF TREE BASED CLASSIFICATION ALGORITHM

3.1 Introduction of New Heuristic Functions to form Improved Algorithms

Some basic steps of a generic tree based classification algorithm are shown in a flow chart given below:

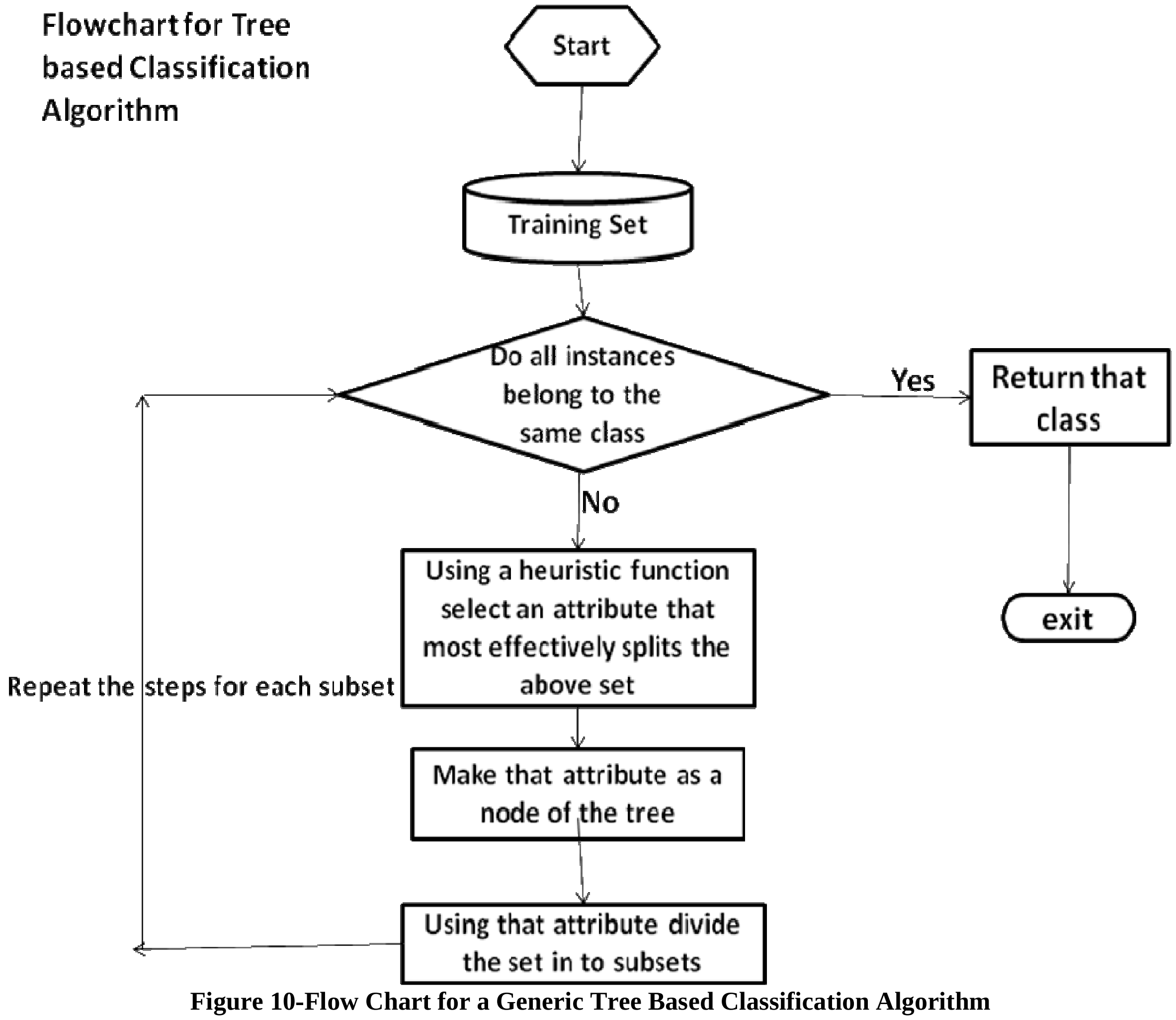

Flowchart for Tree based Classification Algorithm
Start
Training Set
Do all instances belong to the same class
Yes
Return that class
exit
No
Using a heuristic function select an attribute that most effectively splits the above set
Repeat the steps for each subset
Make that attribute as a node of the tree
Using that attribute divide the set in to subsets

Figure 10-Flow Chart for a Generic Tree Based Classification Algorithm

`

The performance of a tree based classification algorithm largely depends upon the heuristic function used in it. There are two critical factors always associated with the heuristic function

- Time Complexity
- Prediction accuracy

In this research I propose two new heuristic functions which are better than the one used by C4.5 algorithm by some way or the other. They are given as:

3.1.1 Simplified Measure of Disorderness to reduce the Time Complexity (Algorithm 1)

The Heuristic Function used in C4.5 algorithm is based on concept of Information entropy. Information entropy is the phenomena to calculate the disorderness in probability distribution..Information entropy is widely used as the measure of disorderness. It is given as

$$E=-\sum_{i} pi*\log(pi)$$

Example: A Fair Dice

Probability of each outcome of a dice is same:

P(1)=P(2)=P(3)=P(4)=P(5)=P(6)=1/6

This is the case of uniform probability distribution and maximum uncertainty. Maximum uncertainty occurs when probability distribution is uniform.

So for this case Entropy is equal to 1.

Another Example: Unfair Dice

P(1)=1, p(2) =P(3)=P(4)=P(5)=P(6)=0

For this case Entropy is equal to 0

In this way the concept of information entropy is used to measure the degree of disorderness. For more uniform probability distribution, entropy is high and for non uniform probability distribution, entropy is low.

ID3/C4.5 Algorithm is based on some assumptions

- A smaller Decision tree with less no. of branches and/or less height or depth has higher prediction accuracy. That is why algorithm emphasizes on building a much smaller

`

decision tree. In an effort to build a small decision tree, it selects that attribute which brings us much closer to the final classification.

- Another assumption is, the attribute that results in a maximum disordered state is the best attribute, the one that brings us closest to the final classification. To find out that attribute, it uses the concept of information entropy.

The heuristic function proposed by me is also based on the same assumptions but the difference is in the measure of disorderness. C4.5 uses information entropy as the measure of disorderness. I am using a much simplified measure i.e calculating the maximum probability in the distribution. It can be considered as the approximate measure of disorderness. So introduction of this Heuristic function gives rise to an improved Algorithm. This Algorithm is better than C4.5 in terns of time complexity.

3.1.1.1 Algorithm Explanation (with flow chart)

Steps 2 and 4 of the above C4.5 algorithm are modified and the other steps remain the same.

Step 2. Steps for calculating the value of heuristic function are given as

2.1 For each V=vi Calculate

2.1.1 No. occurrences of each class/ category.

2.1.2 Occurrence Probability of each class/category.

2.1.3 Maximum of above calculated Occurrence probabilities. This the measure of disorderness for V=vi.

2.2 **H(attribute)=** Compute the overall disorderness for the attribute V by computing the average of disorderness for each V=vi.

Step 4. Select an attribute for which the value of heuristic function is maximum.

Flow chart for the Heuristic Function used in this algorithm is given as:

`

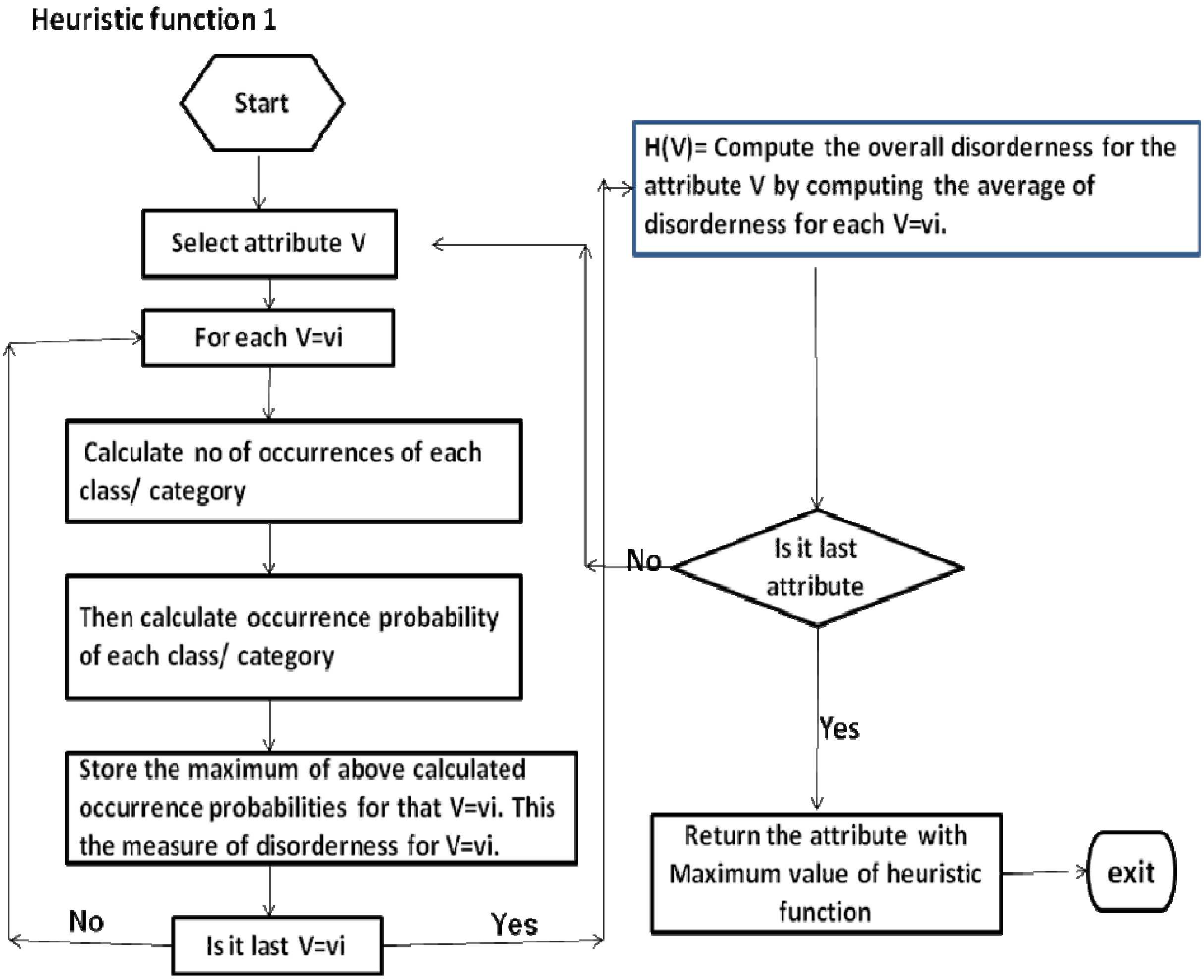

Heuristic function 1
Start
Select attribute V
For each V=vi
Calculate no of occurrences of each class/ category
Then calculate occurrence probability of each class/ category
Store the maximum of above calculated occurrence probabilities for that V=vi. This the measure of disorderness for V=vi.
Is it last V=vi
No
Yes
H(V)= Compute the overall disorderness for the attribute V by computing the average of disorderness for each V=vi.
Is it last attribute
No
Yes
Return the attribute with Maximum value of heuristic function
exit

Figure 11-Flow Chart for a Heuristic Function Used in Algorithm 1

3.1.1.2 Example-A case study (To explain the Algorithm)

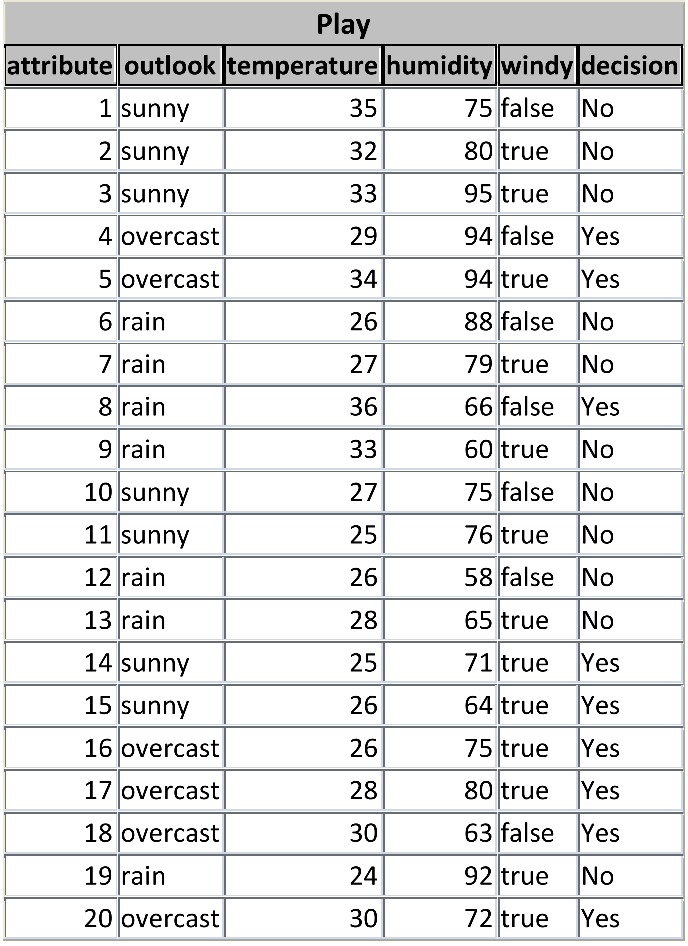
We will use the same example/case study to explain the above algorithm which we used in case of C4.5.

Table 8 below contains a data set of related indicators according to the weather conditions to decide whether to go for play or not. There are total 4 attributes outlook, temperature, humidity and windy. These attributes decide the value of the 5th attribute i.e. decision. Decision attribute can take either of the 2 values "yes" which means go for play and "No" which means don't go for play.

Make the sample data set as the training set now and construct a decision tree using an algorithm given below.

`

Table 8- UCI weather Data set- Example to explain Algorithm 1



Play					
attribute	**outlook**	**temperature**	**humidity**	**windy**	**decision**
1	sunny	35	75	false	No
2	sunny	32	80	true	No
3	sunny	33	95	true	No
4	overcast	29	94	false	Yes
5	overcast	34	94	true	Yes
6	rain	26	88	false	No
7	rain	27	79	true	No
8	rain	36	66	false	Yes
9	rain	33	60	true	No
10	sunny	27	75	false	No
11	sunny	25	76	true	No
12	rain	26	58	false	No
13	rain	28	65	true	No
14	sunny	25	71	true	Yes
15	sunny	26	64	true	Yes
16	overcast	26	75	true	Yes
17	overcast	28	80	true	Yes
18	overcast	30	63	false	Yes
19	rain	24	92	true	No
20	overcast	30	72	true	Yes

Step 1. Compute the Heuristic function value (Disorderness) for each attribute

- For attribute: Outlook

  ➢ For Oulook=sunny

    i. Total no. Of instances=7

    ii. No. Of yes's=2

    iii. No. of no’s=5

    iv. Occurrence Probability of No. of yes’s Py=2/7

    v. Occurrence Probability of No. of no’s Pn=5/7

    vi. Max(Py,Pn)=5/7

`

- For Oulook=overcast
  i. Total no. Of instances=6
  ii. No. Of yes's=6
  iii. No. of no’s=0
  iv. Occurrence Probability of No. of yes’s Py=6/6=1
  v. Occurrence Probability of No. of no’s Pn=0/6=0
  vi. Max(Py,Pn)=1
- For Outlook=rain
  i. Total no. Of instances=7
  ii. No. Of yes's=1
  iii. No. of no’s=6
  iv. Occurrence Probability of No. of yes’s Py=1/7
  v. Occurrence Probability of No. of no’s Pn=6/7
  vi. Max(Py,Pn)=6/7
- H(Outlook)=(5/7+1+6/7)/3=6/7=0.857

- For attribute: Temperature(Threshold value is 30)
  - For Temperature>30
    i. Total no. Of instances=6
    ii. No. Of yes's=2
    iii. No. of no’s=4
    iv. Occurrence Probability of No. of yes’s Py=2/6=1/3
    v. Occurrence Probability of No. of no’s Pn=4/6=2/3
    vi. Max(Py,Pn)=2/3
  - For Temperature<=30

`

i. Total no. Of instances=14
ii. No. Of yes's=7
iii. No. of no’s=7
iv. Occurrence Probability of No. of yes’s Py=7/14=1/2
v. Occurrence Probability of No. of no’s Pn=7/14=1/2
vi. Max(Py,Pn)=1/2

- H(Temperature)=(2/3+1/2)/2=7/12=0.58

- For attribute: Humidity(Threshold value is 76)
  - For Humidity>76
    i. Total no. Of instances=8
    ii. No. Of yes's=3
    iii. No. of no’s=5
    iv. Occurrence Probability of No. of yes’s Py=3/8
    v. Occurrence Probability of No. of no’s Pn=5/8
    vi. Max(Py,Pn)=5/8
  - For Humidity<=76
    i. Total no. Of instances=12
    ii. No. Of yes's=6
    iii. No. of no’s=6
    iv. Occurrence Probability of No. of yes’s Py=6/12=1/2
    v. Occurrence Probability of No. of no’s Pn=6/12=1/2
    vi. Max(Py,Pn)=1/2
  - H(Humidity)=(5/8+1/2)/2=9/16=0.56
- For attribute: Windy

`

- For Windy=true

  i. Total no. Of instances=13

  ii. No. Of yes's=6

  iii. No. of no’s=7

  iv. Occurrence Probability of No. of yes’s Py=6/13

  v. Occurrence Probability of No. of no’s Pn=7/13

  vi. Max(Py,Pn)=7/13

- For Windy=false

  i. Total no. Of instances=7

  ii. No. Of yes's=3

  iii. No. of no’s=4

  iv. Occurrence Probability of No. of yes’s Py=3/7

  v. Occurrence Probability of No. of no’s Pn=4/7

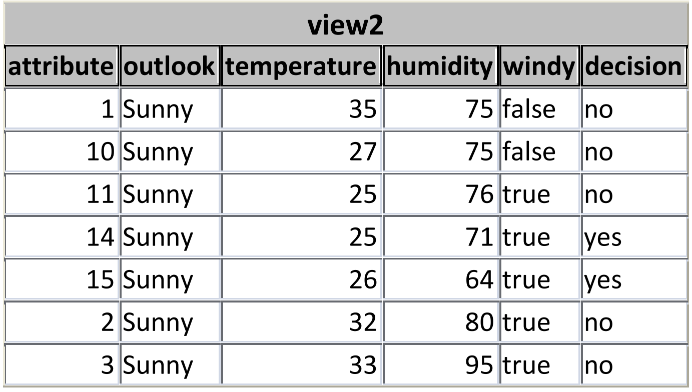
  vi. Max(Py,Pn)=4/7

- H(Windy)=(7/13+4/7)/2=101/182=0.554

Step 2. Then we will select that attribute for which Heuristic Function value is maximum. Clearly it is maximum for **outlook** attribute.

Step 3. Then on the basis of that attribute we will divide the given training set in to subsets and move to another level of tree.

Table 9- Subset outlook=”sunny” (Produced from Table 8)



view2					
attribute	outlook	temperature	humidity	windy	decision
1	Sunny	35	75	false	no
10	Sunny	27	75	false	no
11	Sunny	25	76	true	no
14	Sunny	25	71	true	yes
15	Sunny	26	64	true	yes
2	Sunny	32	80	true	no
3	Sunny	33	95	true	no

`

Table 10- Subset outlook="overcast" (Produced from Table 8)

view11					
attribute	outlook	temperature	humidity	windy	decision
16	overcast	26	75	true	yes
17	overcast	28	80	true	yes
18	overcast	30	63	false	yes
20	overcast	30	72	true	yes
4	overcast	29	94	false	yes
5	overcast	34	94	true	yes

Table 11- Subset outlook="rain" (Produced from Table 8)

view12					
attribute	outlook	temperature	humidity	windy	decision
12	Rain	26	58	false	no
13	Rain	28	65	true	no
19	Rain	24	92	true	no
6	Rain	26	88	false	no
7	rain	27	79	true	no
8	rain	36	66	false	yes
9	rain	33	60	true	no

Step 4. Then we will repeat the same steps on each subset and apply the algorithm recursively

In this way the final tree generated looks like

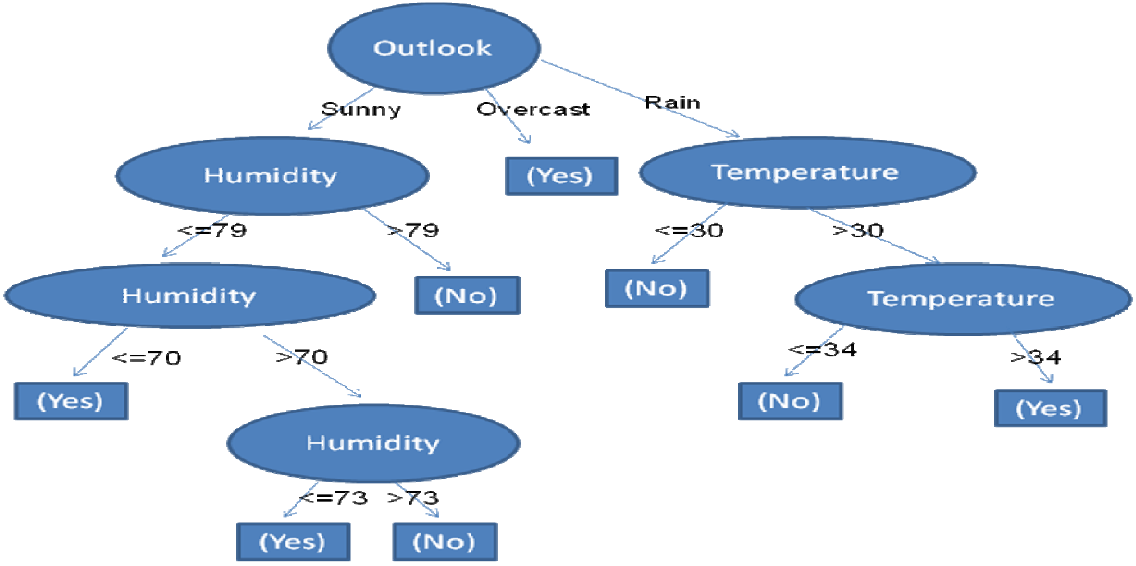


Figure 12-Decision Tree Generated on applying Algorithm 1 on Weather Training data

`

3.1.1.3 Comparison with C4.5 Algorithm-To prove better Time Complexity

This heuristic function is a simplified form of the one used by C4.5. From the Table-12 it is clear that the no of calculations performed in this heuristic function are less as compare to the entropy calculations performed in C4.5 algorithm. So the execution time is less and the output decision tree is more or less similar to the one generated by C4.5 classifier in terms of either structure or prediction accuracy or both. We will be using four examples below to illustrate our point.

Table 12- Comparison of no. of calculation/operations performed in C4.5 algorithm and algorithm 1

Heuristic Functions	**No. of additions**	**No. of Multiplications**	**No. of Divisions**	**Time Complexity in Big O notation**
C4.5 Heuristic Function	(c-1)+u(c-1)+(u-1)	c+uc+u	c+u+uc+1	O(u*n*c)
Algorithm 1 Heuristic Function	(u-1)	0	0	O(u*n*c)

Note:*c is the no. of categories/classes of the decision attribute, u is the no. of unique values of an attribute V and n is the no. of instances in the training set. The figures shown in table 1 are for calculating the value of heuristic function for a particular attribute V. Ignore the calculations of occurrence probability of each attribute values.

3.1.2 A More Realistic Heuristic Function that leads to better prediction accuracy (Algorithm 2)

Few Important points of this Heuristic Function

- Rather than finding out the attribute that brings us closer to the classification, here we emphasize on measuring the importance of an attribute in decision making. This heuristic function calculates how much important that attribute is in decision making.

- Unlike C4.5 this Heuristic Function is not based on the assumption that a smaller decision tree is a better decision tree in terms of prediction accuracy. It is based on the fact that a decision tree with realistic attributes at the top is a much better tree in terms of prediction accuracy.

- Sometimes an attribute individually is not that much important in decision making but when combined with other attributes becomes important in decision making. Unlike C4.5, this Heuristic function takes all the combinations of attributes while measuring the

`

importance of an attribute in decision making. The value of Information gain ratio (Heuristic function of C4.5) computed for any attribute depends only that attribute and the decision attribute. But in this case since Heuristic function is taking in to account all the combinations of attributes while measuring the importance of attribute in decision making, so the value depends upon the other attributes as well. This nature of Heuristic function results in more Real Decision Tree and thus gives more accurate and reasonable results.

3.1.2.1 Algorithm Explanation (with flow chart)

Steps 2 and 4 of the above C4.5 algorithm are modified and the other steps remain the same.

Step 2. Heuristic function is given as:

H(attribute)=∑(No of unique values of that attribute for that decision value)/(Total no

(For each decision value) of unique values)

+

∑

(For each combination of attributes other than current attribute)

∑((No of unique combination of values covered by the current attribute and the

(For each decision value) given combination for that decision value)/(Total no. Of unique combinations of values))

Note: If attribute domain consists of range rather than unique values, then heuristic functions is modified. We will divide the range in to some fixed no of ranges and we will be calculating the no of ranges rather than no of unique values.

Step 4. Select an attribute for which the value of heuristic function is minimum.

`

Flow chart for heuristic function used in this algorithm is given as.

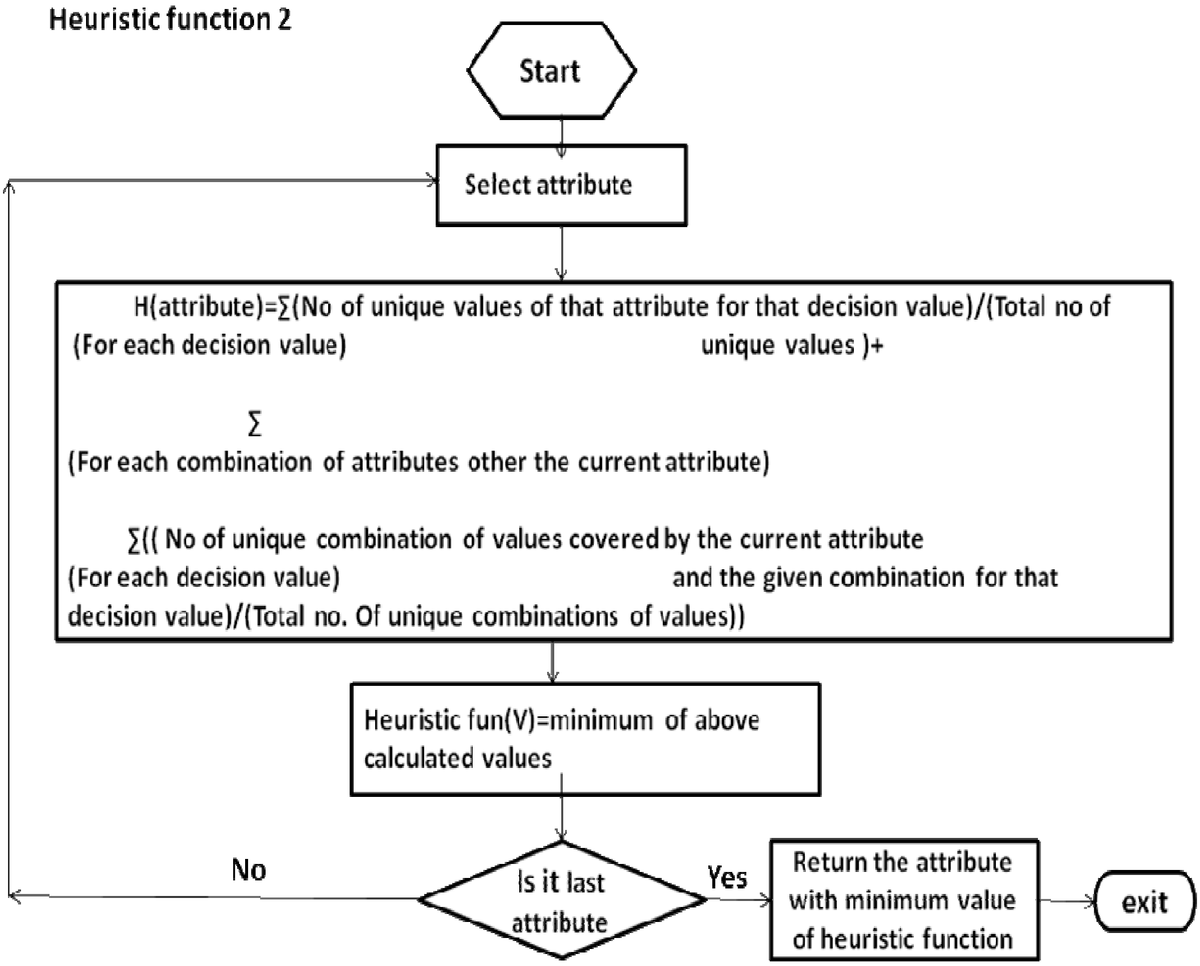

Heuristic function 2
Start
Select attribute
H(attribute)=∑(No of unique values of that attribute for that decision value)/(Total no of
(For each decision value) unique values)+
∑
(For each combination of attributes other the current attribute)
∑((No of unique combination of values covered by the current attribute
(For each decision value) and the given combination for that
decision value)/(Total no. Of unique combinations of values))
Heuristic fun(V)=minimum of above calculated values
Is it last attribute
No
Yes
Return the attribute with minimum value of heuristic function
exit

Figure 13-Flow Chart for a Heuristic Function Used in Algorithm 2

3.1.2.2 Example-A case study (To explain the Algorithm)

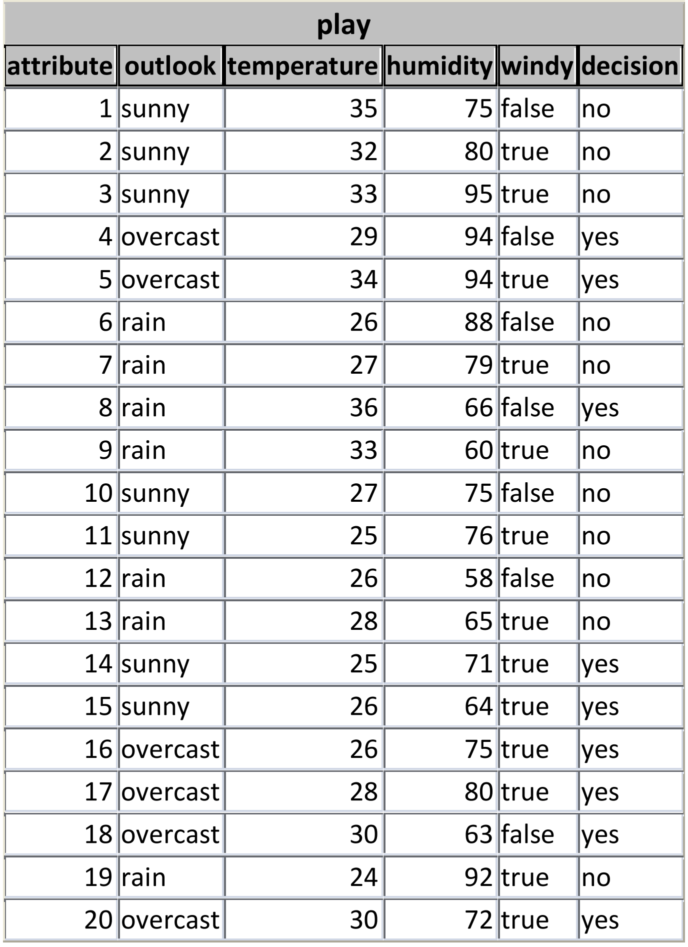
We will use the same example/case study to explain the above algorithm which we used in case of C4.5.

Table 13 below contains a data set of related indicators according to the weather conditions to decide whether to go for play or not. There are total 4 attributes outlook, temperature, humidity and windy. These attributes decide the value of the 5th attribute i.e. decision. Decision attribute can take either of the 2 values “yes” which means go for play and “No” which means don’t go for play.

Make the sample data set as the training set now and construct a decision tree using an algorithm given below.

`

Table 13- UCI weather Data set- Example to explain Algorithm 2



play					
attribute	outlook	temperature	humidity	windy	decision
1	sunny	35	75	false	no
2	sunny	32	80	true	no
3	sunny	33	95	true	no
4	overcast	29	94	false	yes
5	overcast	34	94	true	yes
6	rain	26	88	false	no
7	rain	27	79	true	no
8	rain	36	66	false	yes
9	rain	33	60	true	no
10	sunny	27	75	false	no
11	sunny	25	76	true	no
12	rain	26	58	false	no
13	rain	28	65	true	no
14	sunny	25	71	true	yes
15	sunny	26	64	true	yes
16	overcast	26	75	true	yes
17	overcast	28	80	true	yes
18	overcast	30	63	false	yes
19	rain	24	92	true	no
20	overcast	30	72	true	yes

Step 1. **Calculate the value of H(outlook)**

- For Outlook
  - Total no. Of Unique Outlook values in the training set t=3
  - For “no”
    1. No. Of unique Outlook values n=2
    2. H(outlook)+=n/t=2/3
  - For “yes”
    1. No. Of unique Outlook values n=3
    2. H(outlook)+=n/t=3/3=1
- For (Outlook, Temperature)
  - Total no. Of Unique (Outlook, Temperature) combinations in the training set t=17
  - For “no”
    1. No. Of unique (Outlook, temperature) combinations n=10
    2. H(outlook)+=n/t=10/17
  - For “yes”
    1. No. Of unique (Outlook, temperature) combinations n=8

`

        ii. H(outlook)+=n/t=8/17

- For(Outlook, Temperature, Humidity)
    - Total no. Of Unique (Outlook, Temperature, Humidity) combinations in the training set t=20
    - For “no”
        i. No. Of unique (Outlook, temperature, humidity) combinations n=11
        ii. H(outlook)+=n/t=11/20
    - For “yes”
        i. No. Of unique (Outlook, temperature, humidity) combinations n=9
        ii. H(outlook)+=n/t=9/20
- For(Outlook, Temperature, Humidity, windy)
    - Total no. Of Unique (Outlook, Temperature, Humidity) combinations in the training set t=20
    - For “no”
        i. No. Of unique (Outlook, temperature, humidity, windy) combinations n=11
        ii. H(outlook)+=n/t=11/20
    - For “yes”
        i. No. Of unique (Outlook, temperature, humidity, windy) combinations n=9
        ii. H(outlook)+=n/t=9/20
- For(Outlook, Temperature, Windy)
    - Total no. Of Unique (Outlook, Temperature, Windy) combinations in the training set t=18
    - For “no”
        i. No. Of unique (Outlook, temperature, Windy) combinations n=10
        ii. H(outlook)+=n/t=10/18
    - For “yes”
        i. No. Of unique (Outlook, temperature, Windy) combinations n=9
        ii. H(outlook)+=n/t=9/18
- For (Outlook, Humidity)
    - Total no. Of Unique (Outlook, Humidity) combinations in the training set t=18
    - For “no”
        i. No. Of unique (Outlook, Humidity) combinations n=10
        ii. H(outlook)+=n/t=10/18
    - For “yes”
        i. No. Of unique (Outlook, Humidity) combinations n=8
        ii. H(outlook)+=n/t=8/18
- For (Outlook, Humidity, windy)
    - Total no. Of Unique (Outlook, Humidity, windy) combinations in the training set t=19
    - For “no”
        i. No. Of unique (Outlook, Humidity, windy) combinations n=10
        ii. H(outlook)+=n/t=10/19
    - For “yes”
        i. No. Of unique (Outlook, Humidity, windy) combinations n=9

`

ii. H(outlook)+=n/t=9/19

- For (Outlook, Windy)
    - Total no. Of Unique (Outlook, windy) combinations in the training set t=6
    - For “no”
        i. No. Of unique (Outlook, Windy) combinations n=4
        ii. H(outlook)+=n/t=4/6=2/3
    - For “yes”
        i. No. Of unique (Outlook, Windy) combinations n=4
        ii. H(outlook)+=n/t=4/6=2/3
- In this way the final value comes out to be **H(outlook)**=9.11

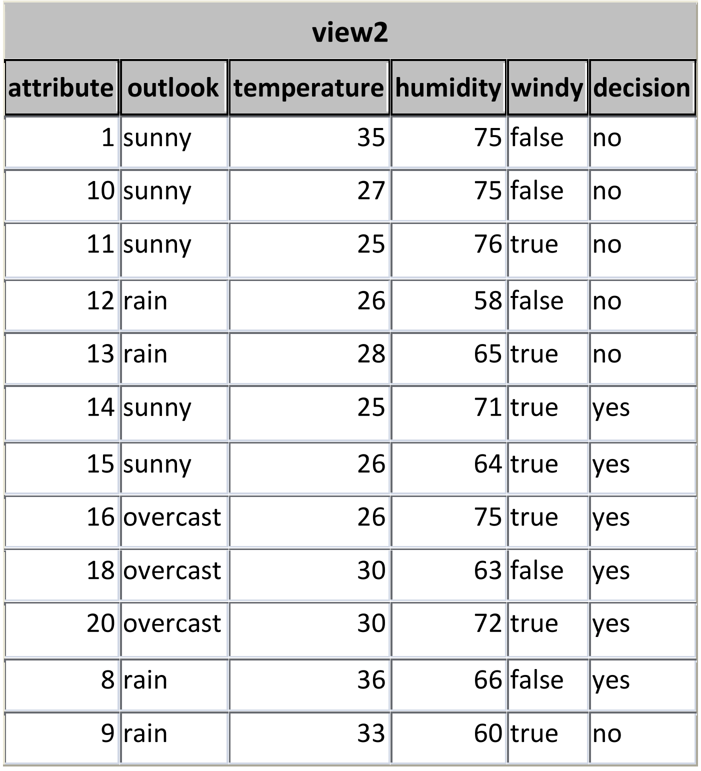
Step 2. Similarly we will calculate the Heuristic Function value for the other attributes

H(Temperature)=8.497712
H(Humidity)=8.180555
H(Windy)=9.577778

Step 3. Then we will select that attribute for which Heuristic Function value is minimum. Clearly it is minimum for **humidity** attribute.

Step 4. Then on the basis of that attribute we will divide the given training set in to subsets and move to another level of tree.

Table 14- Subset humidity<=76 (Produced from Table 13)

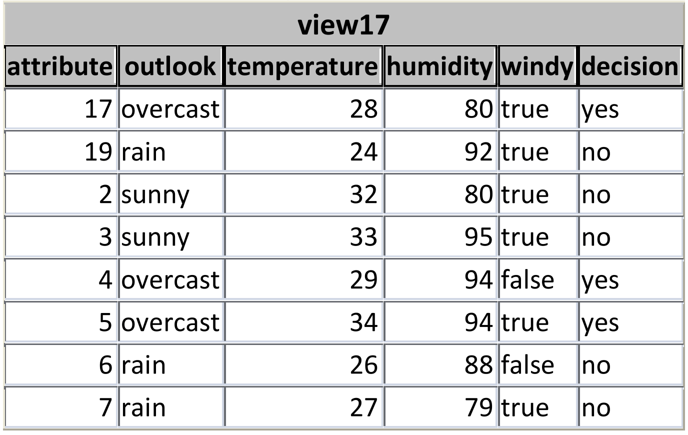


view2

attribute	outlook	temperature	humidity	windy	decision
1	sunny	35	75	false	no
10	sunny	27	75	false	no
11	sunny	25	76	true	no
12	rain	26	58	false	no
13	rain	28	65	true	no
14	sunny	25	71	true	yes
15	sunny	26	64	true	yes
16	overcast	26	75	true	yes
18	overcast	30	63	false	yes
20	overcast	30	72	true	yes
8	rain	36	66	false	yes
9	rain	33	60	true	no

`

Table 15- Subset humidity>76 (Produced from Table 13)



view17					
attribute	outlook	temperature	humidity	windy	decision
17	overcast	28	80	true	yes
19	rain	24	92	true	no
2	sunny	32	80	true	no
3	sunny	33	95	true	no
4	overcast	29	94	false	yes
5	overcast	34	94	true	yes
6	rain	26	88	false	no
7	rain	27	79	true	no

Step 5. Then we will repeat the same steps on each subset and apply the algorithm recursively

In this way the final tree generated looks like

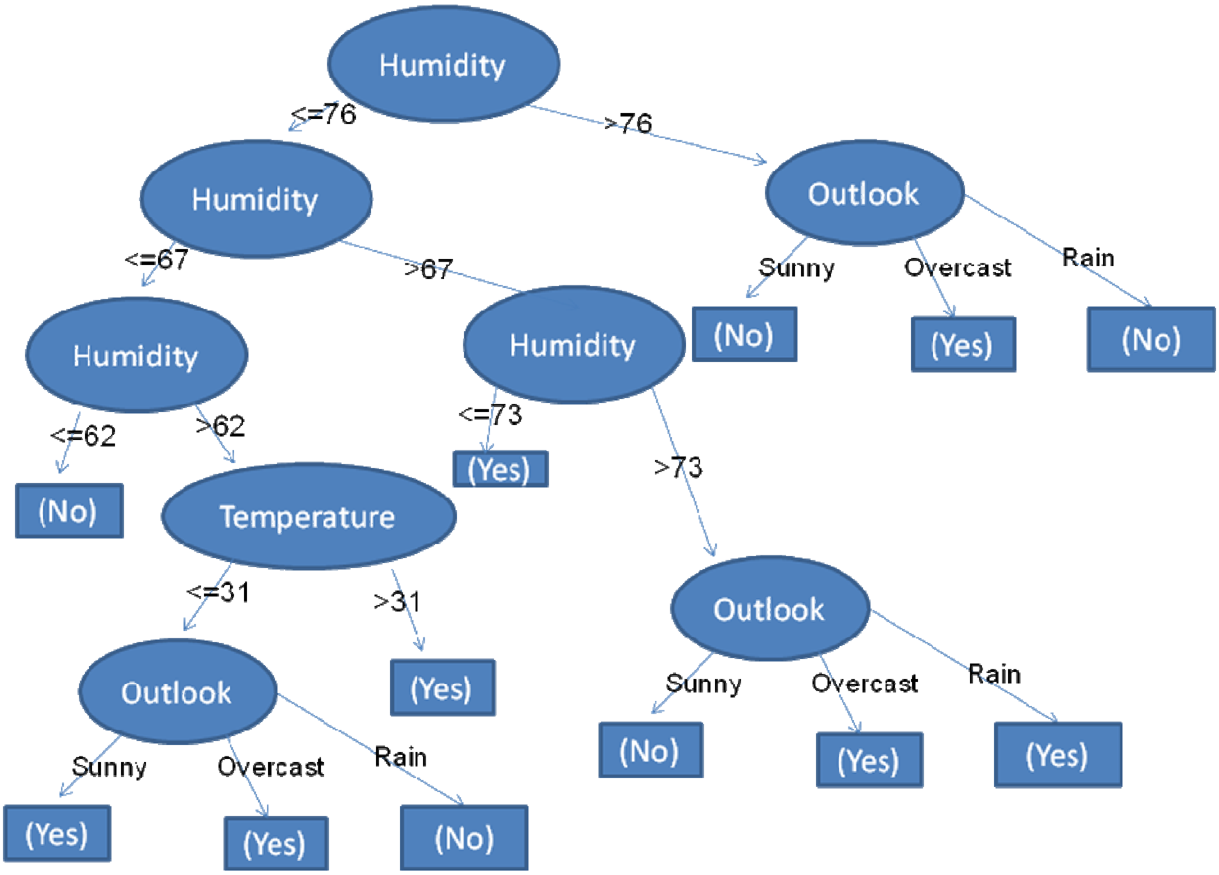


Figure 14-Decision Tree Generated on applying Algorithm 2 on Weather Training data.

`

3.1.2.3 Comparison with C4.5 Algorithm-To prove better Prediction accuracy

- This heuristic function gives more importance to realistic attributes like humidity and temperature and thus is more real and gives more accurate and reasonable results. Although it takes more time to build but prediction accuracy comes out to be more than C4.5 algorithm. We will be using four examples below to illustrate our point.

- Unlike C4.5 this Heuristic Function is not based on the assumption that a smaller decision tree is a better decision tree in terms of prediction accuracy. It is based on the fact that a decision tree with realistic attributes at the top is a much better tree in terms of prediction accuracy.

- Sometimes an attribute individually is not that much important in decision making but when combined with other attributes becomes important in decision making. Unlike C4.5, this Heuristic function takes all the combinations of attributes while measuring the importance of an attribute in decision making. The value of Information gain ratio (Heuristic function of C4.5) computed for any attribute depends only that attribute and the decision attribute. But in this case since Heuristic function is taking in to account all the combinations of attributes while measuring the importance of attribute in decision making, so the value depends upon the other attributes as well. This nature of Heuristic function results in more Real Decision Tree and thus gives more accurate and reasonable results.

`

4. CASE STUDIES AND RESULTS

I establish my claims by implementing the above algorithms on different case studies (examples). The idea behind taking more than one case study (example) is to prove that the algorithms not only work well for one type of data sets but also for varied data sets.

- First case study is a simple example with smaller data set.
- 2nd and 3rd case studies are much more complex examples.
- The last case study is based on the real AIEEE data containing instances in the range of thousands.

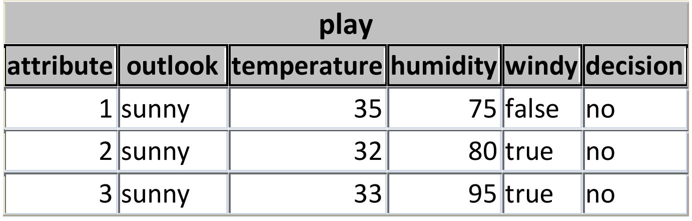
In this section I have tried to validate the proposed algorithms by applying them on different case studies. I have tried to prove that algorithms not only work well for simpler data sets but also for complex data sets.

4.1 A case study on weather

4.1.1 Explanation

It's a very popular example used to compare different classification algorithms. Best thing about this example is its simplicity and ease to understand. Table 1 below contains a data [1] set of related indicators according to the weather conditions to decide whether to go for play or not. There are total 4 attributes outlook, temperature, humidity and windy. These attributes decide the value of the 5th attribute i.e. decision. Decision attribute can take either of the two values "yes" which means go for play and "No" which means don't go for play. Make the sample data set [4] (UCI Data Set) as the training set now and construct a decision tree using algorithms given above. It is a test of the all the above algorithms on a smaller data set. A test on a larger and more complex data set is done using other case studies which are explained in the subsequent sections.

Table 16- UCI Weather data set (A case study-To compare performance of algorithms)



play					
attribute	outlook	temperature	humidity	windy	decision
1	sunny	35	75	false	no
2	sunny	32	80	true	no
3	sunny	33	95	true	no

`

play					
attribute	outlook	temperature	humidity	windy	decision
4	overcast	29	94	false	yes
5	overcast	34	94	true	yes
6	Rain	26	88	false	no
7	Rain	27	79	true	no
8	Rain	36	66	false	yes
9	Rain	33	60	true	no
10	sunny	27	75	false	no
11	sunny	25	76	true	no
12	Rain	26	58	false	no
13	Rain	28	65	true	no
14	sunny	25	71	true	yes
15	sunny	26	64	true	yes
16	overcast	26	75	true	yes
17	overcast	28	80	true	yes
18	overcast	30	63	false	yes
19	Rain	24	92	true	no
20	Overcast	30	72	true	yes

4.1.2 Application of Algorithms on the case study

4.1.2.1 Running C4.5 Algorithm on the Training Data

Decision Tree Generated looks like

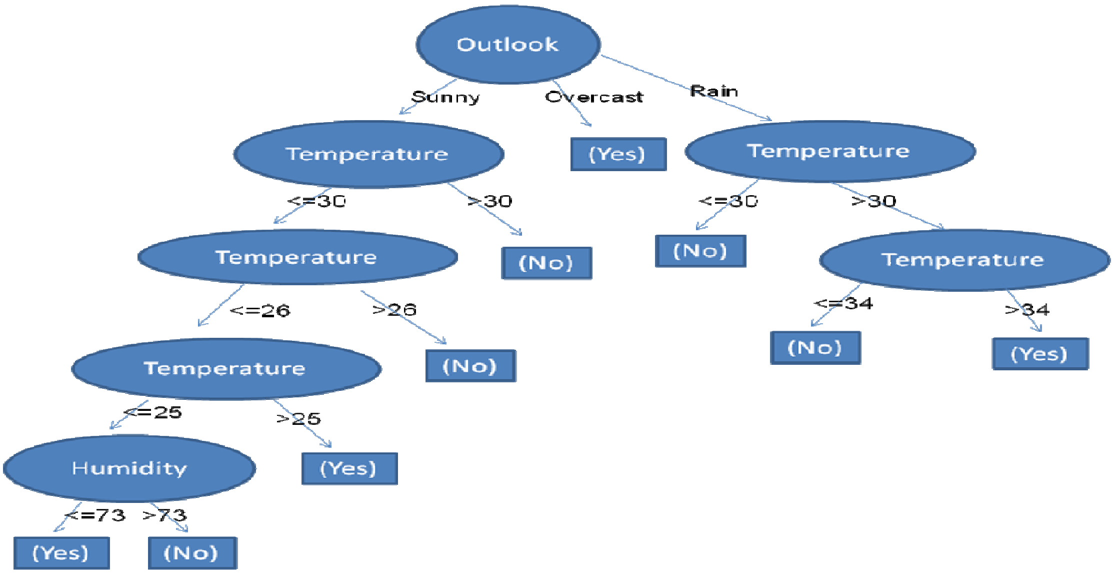


Figure 15-Decision Tree Generated on applying C4.5 Algorithm on weather training data

`

Algorithm Statistics:

Time to build: 1235 mille seconds

Prediction accuracy on training Set: 100%

Prediction accuracy on Supplied Set: 72%

Prediction accuracy (Stratified Cross Validation):

Fold1: 50%

Fold2: 80%

Average: 65%

Note: These statistics have been generated by implementing the algorithm using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown in the subsequent sections.

4.1.2.2 Running Algorithm 1 on the Training Data

Decision Tree Generated looks like

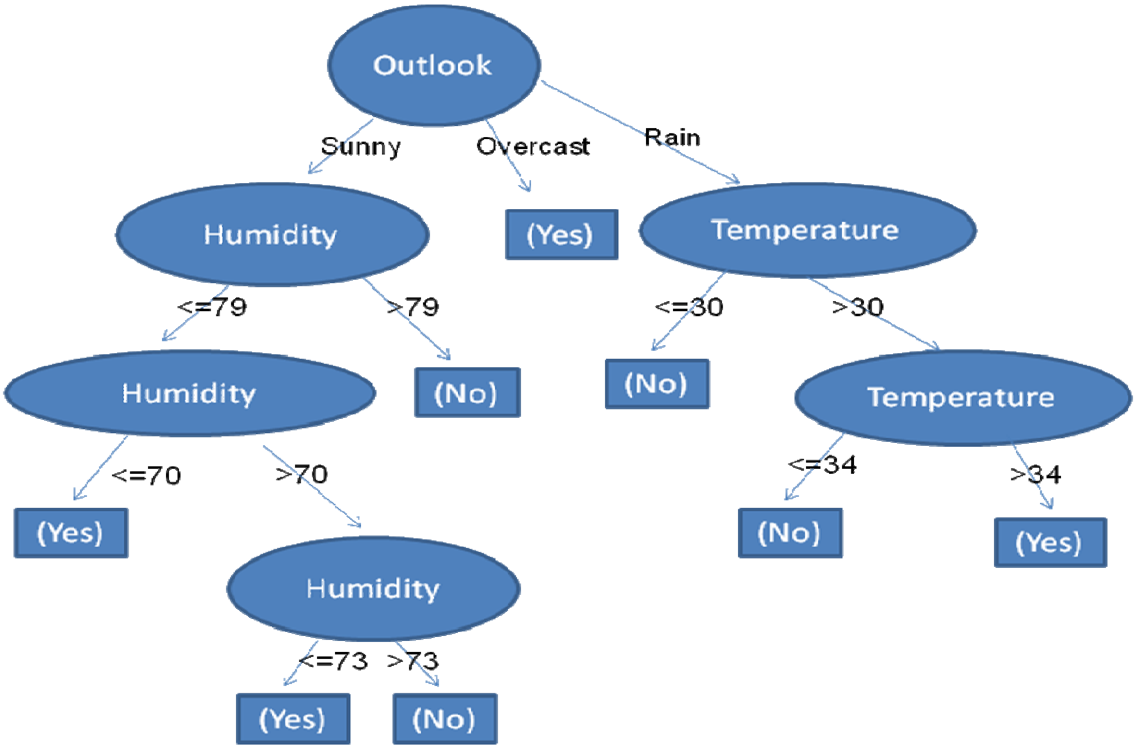


Figure 16-Decision Tree Generated on applying Algorithm 1 on weather training data

`

Algorithm Statistics:

Time to build: 968 mille seconds

Prediction accuracy on training Set: 100%

Prediction accuracy on Supplied Set: 72%

Prediction accuracy (Stratified Cross Validation):

Fold1: 50%

Fold2: 80%

Average: 65%

Note: These statistics have been generated by implementing the algorithm using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown in the subsequent sections.

4.1.2.3 Running Algorithm 2 on the Training Data

Decision Tree Generated looks like

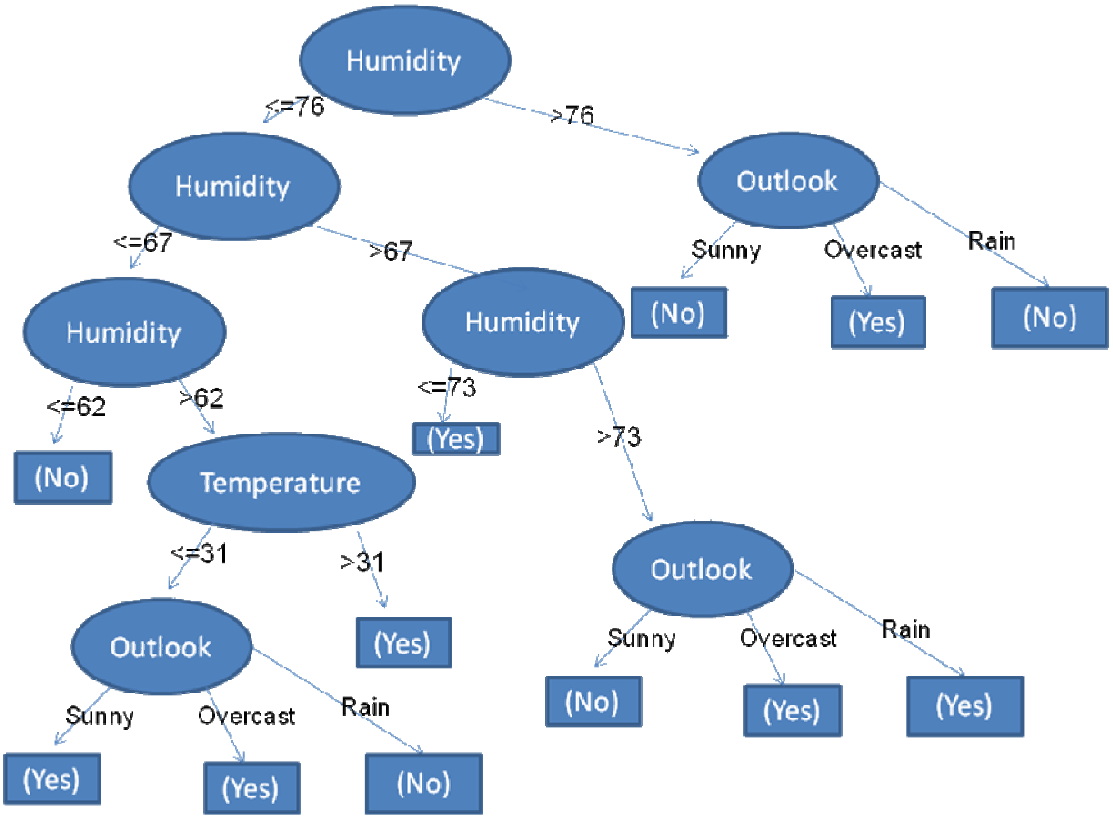


Figure 17-Decision Tree Generated on applying Algorithm 2 on weather training data

`

Algorithm Statistics:

Time to build: 2688 mille seconds

Prediction accuracy on training Set: 100%

Prediction accuracy on Supplied Set: 79%

Prediction accuracy (Stratified Cross Validation):

Fold1: 70%

Fold2: 80%

Average: 75%

Note: These statistics have been generated by implementing the algorithm using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown in the subsequent sections.

4.1.3 Analysis and Comparison of Results (With Snap Shots)

If we compare C4.5 and algorithm1, it is clear from the table -17 that the decision tree generated by both the algorithms are very much the same in terms of structure as well as the prediction accuracy (on training set, supplied set, stratified cross-validation) but there is a big difference in time taken to generate the tree.Algorithm1 takes lesser time than C4.5 to generate the decision tree. So Algorithm1 is better than C4.5 in term of time taken to build the tree.

If we compare C4.5 and algorithm2, the decision tree structure is different. Algorithm 2 gives more importance to realistic attributes like humidity and temperature than outlook and thus is more real and gives more accurate and reasonable results. If we compare the statistics C4.5 and algorithm2 given in table-17, it is clear that the algorithm2 has better prediction accuracy (on training set, supplied set and stratified cross validation) as compare to C4.5.

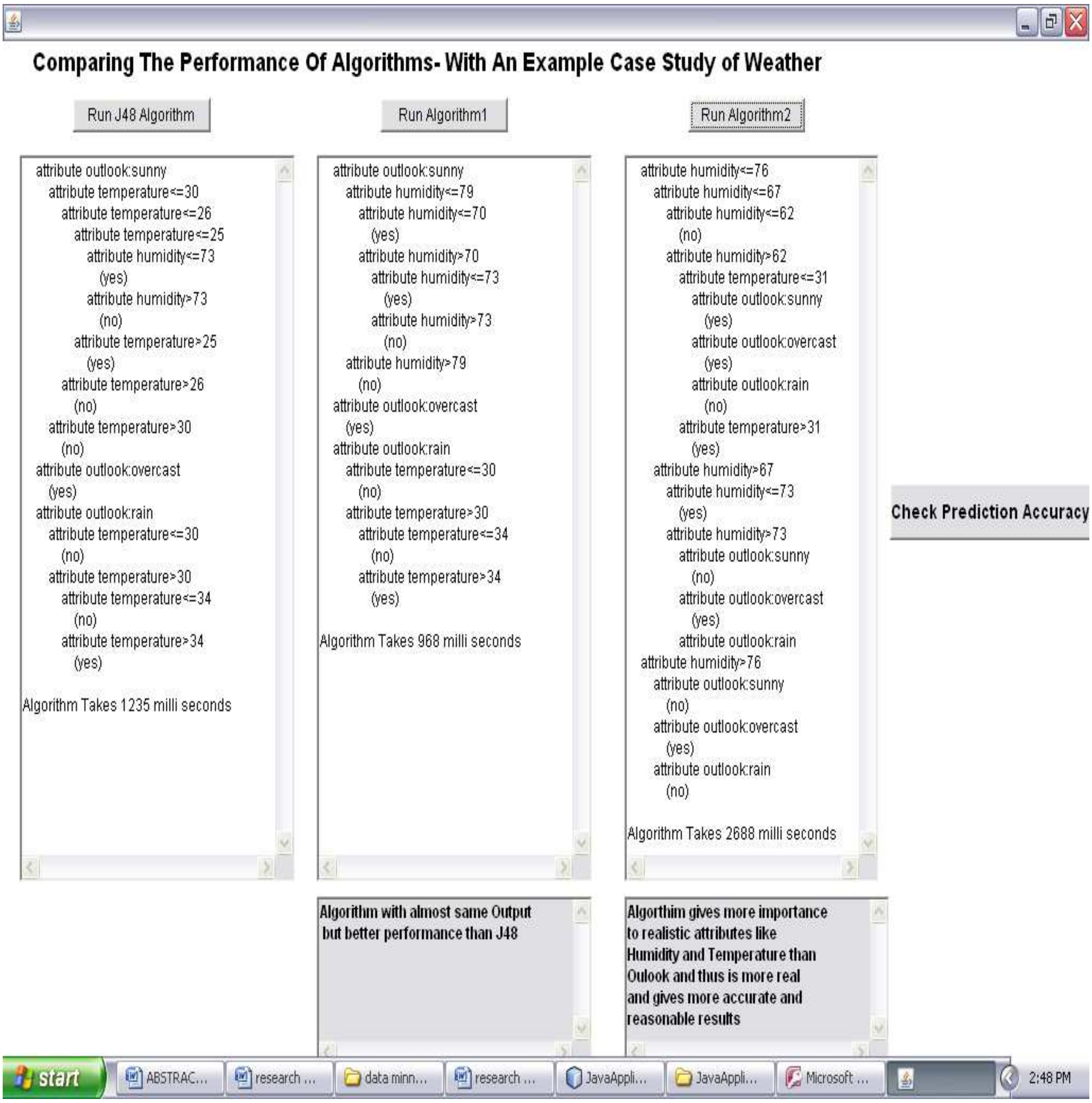
Table-17-Comparing build time and prediction accuracy (Weather Case study)

	Build Time (in ms)	**Prediction accuracy (Training Set)**	**Prediction accuracy (Supplied Set)**	**Prediction accuracy (Cross-Validation Using 2 folds)**
C4.5	1235	100%	72%	65%
Algo1	968	100%	72%	65%
Algo2	2688	100%	79%	75%

`

These statistics have been generated by implementing the algorithms using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown as

Snapshot-Time to build the tree



```
attribute outlook:sunny
  attribute temperature<=30
    attribute temperature<=26
      attribute temperature<=25
        attribute humidity<=73
          (yes)
        attribute humidity>73
          (no)
      attribute temperature>25
        (yes)
    attribute temperature>26
      (no)
  attribute temperature>30
    (no)
attribute outlook:overcast
  (yes)
attribute outlook:rain
  attribute temperature<=30
    (no)
  attribute temperature>30
    attribute temperature<=34
      (no)
    attribute temperature>34
      (yes)

Algorithm Takes 1235 milli seconds
```

```
attribute outlook:sunny
  attribute humidity<=79
    attribute humidity<=70
      (yes)
    attribute humidity>70
      attribute humidity<=73
        (yes)
      attribute humidity>73
        (no)
  attribute humidity>79
    (no)
attribute outlook:overcast
  (yes)
attribute outlook:rain
  attribute temperature<=30
    (no)
  attribute temperature>30
    attribute temperature<=34
      (no)
    attribute temperature>34
      (yes)

Algorithm Takes 968 milli seconds
```

```
attribute humidity<=76
  attribute humidity<=67
    attribute humidity<=62
      (no)
    attribute humidity>62
      attribute temperature<=31
        attribute outlook:sunny
          (yes)
        attribute outlook:overcast
          (yes)
        attribute outlook:rain
          (no)
      attribute temperature>31
        (yes)
  attribute humidity>67
    attribute humidity<=73
      (yes)
    attribute humidity>73
      attribute outlook:sunny
        (no)
      attribute outlook:overcast
        (yes)
      attribute outlook:rain
attribute humidity>76
  attribute outlook:sunny
    (no)
  attribute outlook:overcast
    (yes)
  attribute outlook:rain
    (no)

Algorithm Takes 2688 milli seconds
```

Algorithm with almost same Output but better performance than J48

Algorthim gives more importance to realistic attributes like Humidity and Temperature than Oulook and thus is more real and gives more accurate and reasonable results

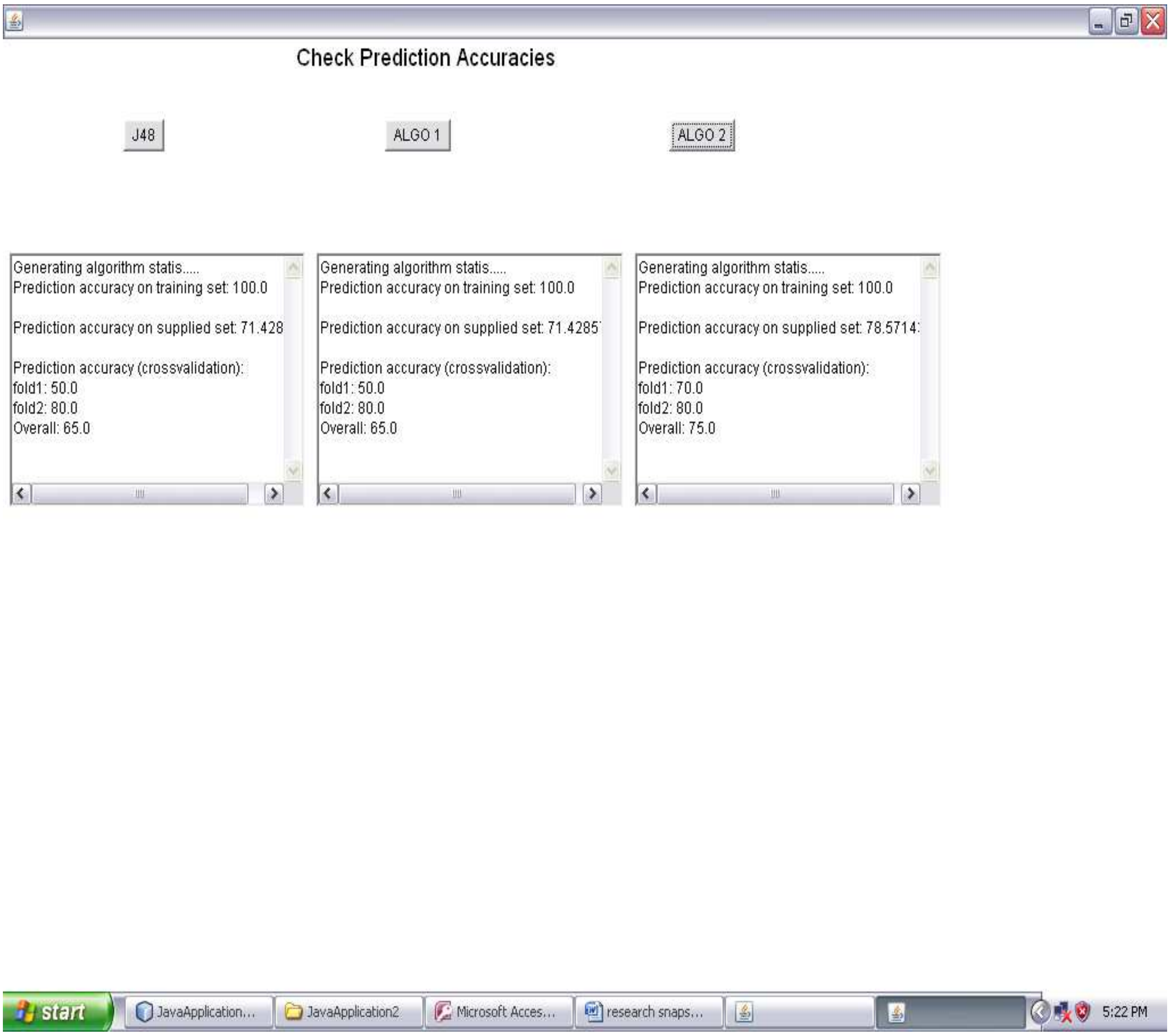
Figure 18-Snapshot- comparing the execution time of algorithms-with an example/case study of weather

In the above snapshot there are three buttons “Run J48 Algorithm”, “Run Algorithm 1”, “Run Algorithm2” and underneath each button there is a Text Area. When we click on button “Run J48 Algorithm”, J48 Tree is displayed in the corresponding Text Area and the time to generate the decision tree is also displayed. Similarly when we click on button “Run Algorithm 1”, a

`

decision tree is displayed in the corresponding Text Area and the time to generate the decision tree is also displayed. In the same way when we click on button “Run Algorithm 1”, a decision tree is displayed in the corresponding Text Area and the time to generate the decision tree is also displayed.

Snapshot-Prediction accuracy (On Training Set, On Supplied Set, Using CROSS VALIDATION-2 FOLD)



```
Generating algorithm statis.....
Prediction accuracy on training set: 100.0

Prediction accuracy on supplied set: 71.428

Prediction accuracy (crossvalidation):
fold1: 50.0
fold2: 80.0
Overall: 65.0
```

```
Generating algorithm statis.....
Prediction accuracy on training set: 100.0

Prediction accuracy on supplied set: 71.4285

Prediction accuracy (crossvalidation):
fold1: 50.0
fold2: 80.0
Overall: 65.0
```

```
Generating algorithm statis.....
Prediction accuracy on training set: 100.0

Prediction accuracy on supplied set: 78.5714

Prediction accuracy (crossvalidation):
fold1: 70.0
fold2: 80.0
Overall: 75.0
```

Figure 19- Snapshot-Comparing the prediction accuracies of algorithms-with an example/case study of weather

In the above snapshot, there are three buttons, “J48”, “ALGO1”, “ALGO2” and underneath each button there is a Text Area. When we click on button “J48”, its Prediction accuracy(On Training set, supplied set, 2- fold cross-validation) are displayed in the corresponding Text area. Similarly when we click on button “ALGO1”, its Prediction accuracy(On Training set, supplied set, 2- fold

`

cross-validation) are displayed in the corresponding Text area. In the same way when we click on button “ALGO2”, its Prediction accuracy(On Training set, supplied set, 2- fold cross-validation) are displayed in the corresponding Text area.

4.2 Predicting and Analyzing Student's behavior in Education using Decision Tree Algorithm (Counseling Help)

4.2.1 Explanation

We did research on how data mining can be used in education. Here we are calculating student’s satisfaction level for a particular engineering branch i.e. offered to him to find out whether the branch suits him or not. We collected the data through a survey, conducted on 100 students of Delhi technological university. Using this data we calculate the satisfaction level. We take the student attributes and combine them with branch attributes to find out the satisfaction level of student for that branch. Student attributes and branch attributes are as follows

Student Attributes

- Students Perspective (further studies/placements)
- Family Pressure (1 to 5)
- Marks (Physics) (1 to 100)
- Marks (Chemistry) (1 to 100)
- Marks (Math’s) (1 to 100)
- Favorite Subject (physics, chemistry, math’s, biology, computers)
- Family Background (graduate, post graduate, not graduate)
- Father Profession (technical, non-technical, business)
- Worker (hard worker, effective worker)
- Creativity (1 to 5)
- Satisfaction Level (1 to 10)

Branch Attributes

- Branch Name (cs, it, ece, me, ce, ps, bt)
- Future Scope (1 to 5)

`

- Placement (1 to 5)
- Work Pressure (1 to 5)
- Branch Type (core, subsidiary, hybrid)
- Further Study Scope (1 to 5)
- Syllabus(ancient, old, new)
- Work Type (practical, theory, mixed)
- Faculty & Facility(1 to 5)
- Familiarity (1 to 5)

This one forms more complex example with larger no. of parameters and larger data set. This complex training set formed by conducting the survey on Delhi technological university

4.2.2 Application of Algorithms on the case study

4.2.2.1 Running C4.5 Algorithm on the Training Data

Decision Tree Generated is shown in section 1 of Appendix A.

Algorithm Statistics:

Time to build: 206406 mille seconds

Prediction accuracy (Stratified Cross-Validation using 10 folds):

Fold1: 30%

Fold2: 10%

Fold3: 10%

Fold4: 30%

Fold5: 0%

Fold6: 20%

Fold7: 40%

Fold8: 20%

`

Fold9: 30%

Fold10: 0%

Average: 20%

Note: These statistics have been generated by implementing the algorithm using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown in the subsequent sections.

4.2.2.2 Running Algorithm 1 on the Training Data

Decision Tree Generated is shown in section 2 of Appendix A.

Algorithm Statistics:

Time to build: 77922 mille seconds

Prediction accuracy (Stratified Cross-Validation using 10 folds):

Fold1: 30%

Fold2: 20%

Fold3: 10%

Fold4: 40%

Fold5: 10%

Fold6: 40%

Fold7: 10%

Fold8: 20%

Fold9: 30%

Fold10: 0%

Average: 21%

Note: These statistics have been generated by implementing the algorithm using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown in the subsequent sections.

4.2.2.3 Running Algorithm 2 on the Training Data

`

Decision Tree Generated is shown in section 3 of Appendix A.

Algorithm Statistics:

Time to build: 415453 mille seconds

Prediction accuracy (Stratified Cross-Validation using 10 folds):

Fold1: 30%

Fold2: 30%

Fold3: 20%

Fold4: 30%

Fold5: 20%

Fold6: 30%

Fold7: 40%

Fold8: 0%

Fold9: 30%

Fold10: 25%

Average: 26%

Note: These statistics have been generated by implementing the algorithm using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown in the next section.

4.2.3 Analysis and Comparison of Results (With Snap Shots)

If we compare C4.5 and algorithm1, it is clear from table-18 that the decision tree generated by both the algorithms are same in terms of prediction accuracy (through stratified cross-validation using 10 folds) but there is a big difference in time taken to generate the tree.Algorithm1 takes lesser time than C4.5 to generate the decision tree. So Algorithm1 is better than C4.5 in term of time taken to build the tree.

If we compare C4.5 and algorithm2, the decision tree structure is different .Algorithm 2 gives more importance to realistic attributes like marks in subjects (like math's, physics, chemistry), branch name , family pressure etc and thus is more real and gives more accurate and reasonable results. If we compare the statistics of C4.5 and algorithm2 given in table-18, it is clear that the

`

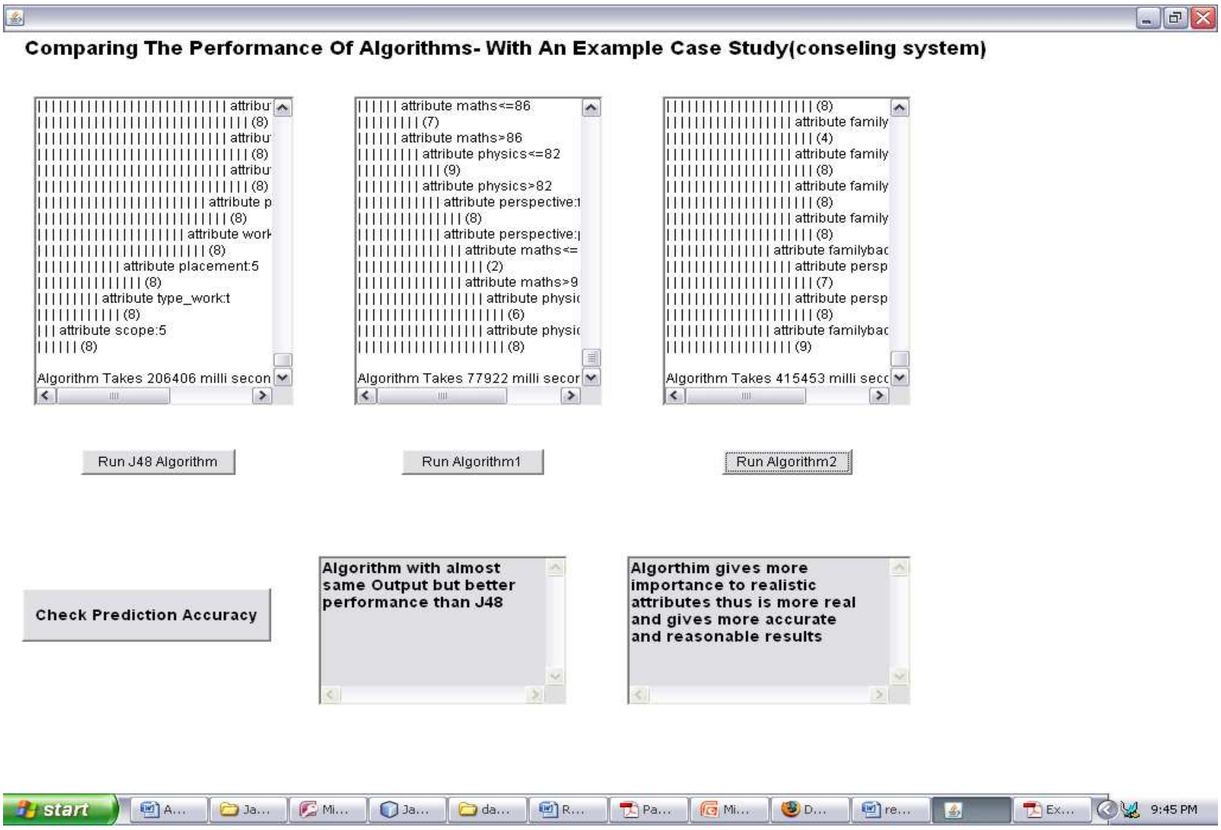
algorithm2 has better prediction accuracy (through stratified cross-validation using 10 folds) as compare to C4.5.

Table-18-Comparing build time and prediction accuracy (Counseling system)

	Build Time	**Prediction accuracy(Cross-Validation Using 10 folds)**
C4.5	88203 msec	20%
Algorithm 1	42344 msec	20%
Algorithm 2	133734 msec	26%

These statistics have been generated by implementing the algorithms using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown as

Snapshot-Time to build the tree



Comparing The Performance Of Algorithms- With An Example Case Study(conseling system)

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribu
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribu
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribu
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute p
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute work
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute placement:5
| | | | | | | | | | | | | | | (8)
| | | | | | | | | attribute type_work:t
| | | | | | | | | | | | (8)
| | | attribute scope:5
| | | | | | (8)

Algorithm Takes 206406 milli secon
```

```
| | | | | | attribute maths<=86
| | | | | | | | | (7)
| | | | | | attribute maths>86
| | | | | | | | | attribute physics<=82
| | | | | | | | | | | | (9)
| | | | | | | | | attribute physics>82
| | | | | | | | | | | | attribute perspective:1
| | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute perspective:|
| | | | | | | | | | | | | | | attribute maths<=
| | | | | | | | | | | | | | | | | | (2)
| | | | | | | | | | | | | | | attribute maths>9
| | | | | | | | | | | | | | | | | | attribute physic
| | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | attribute physic
| | | | | | | | | | | | | | | | | | | | | (8)

Algorithm Takes 77922 milli secor
```

```
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute family
| | | | | | | | | | | | | | | | | | | | | (4)
| | | | | | | | | | | | | | | | | | attribute family
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute family
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute family
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute familybac
| | | | | | | | | | | | | | | | | | attribute persp
| | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | attribute persp
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute familybac
| | | | | | | | | | | | | | | | | | (9)

Algorithm Takes 415453 milli secc
```

Algorithm with almost same Output but better performance than J48

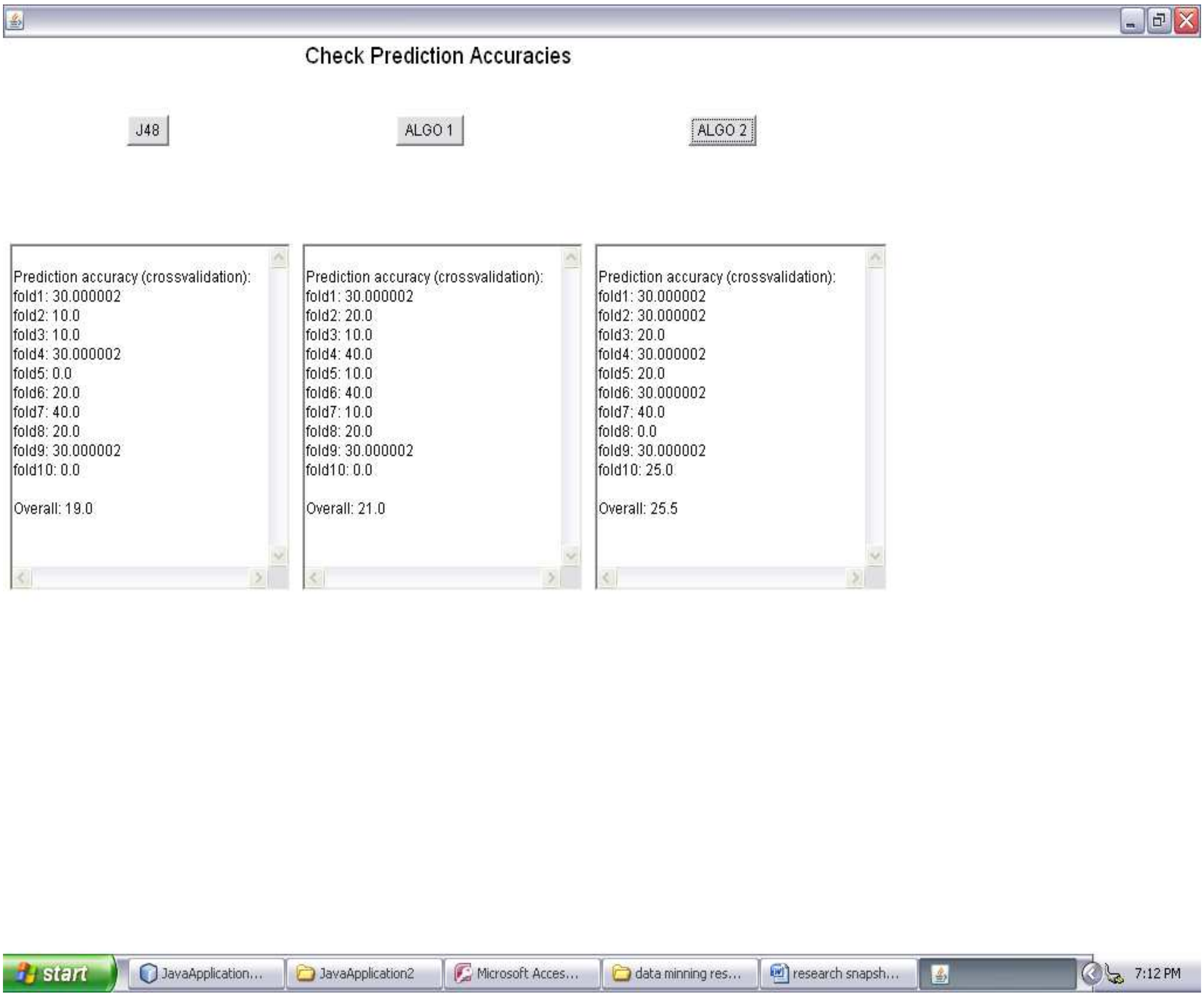
Algorthim gives more importance to realistic attributes thus is more real and gives more accurate and reasonable results

Figure 20- Snapshot-Comparing the execution time of algorithms-with an example/case study (counseling system)

`

In the above snapshot there are three buttons “Run J48 Algorithm”, “Run Algorithm 1”, “Run Algorithm2” and above each button there is a Text Area. When we click on button “Run J48 Algorithm”, J48 Tree is displayed in the corresponding Text Area and the time to generate the decision tree is also displayed. Similarly when we click on button “Run Algorithm 1”, a decision tree is displayed in the corresponding Text Area and the time to generate the decision tree is also displayed. In the same way when we click on button “Run Algorithm 1”, a decision tree is displayed in the corresponding Text Area and the time to generate the decision tree is also displayed.

Snapshot-Prediction accuracy (CROSS VALIDATION-10 FOLD)



```
Prediction accuracy (crossvalidation):
fold1: 30.000002
fold2: 10.0
fold3: 10.0
fold4: 30.000002
fold5: 0.0
fold6: 20.0
fold7: 40.0
fold8: 20.0
fold9: 30.000002
fold10: 0.0

Overall: 19.0
```

```
Prediction accuracy (crossvalidation):
fold1: 30.000002
fold2: 20.0
fold3: 10.0
fold4: 40.0
fold5: 10.0
fold6: 40.0
fold7: 10.0
fold8: 20.0
fold9: 30.000002
fold10: 0.0

Overall: 21.0
```

```
Prediction accuracy (crossvalidation):
fold1: 30.000002
fold2: 30.000002
fold3: 20.0
fold4: 30.000002
fold5: 20.0
fold6: 30.000002
fold7: 40.0
fold8: 0.0
fold9: 30.000002
fold10: 25.0

Overall: 25.5
```

Figure 21- Snapshot-Comparing the prediction accuracies of algorithms-with an example/case study (counseling system)

`

In the above snapshot, there are three buttons, “J48”, “ALGO1”, “ALGO2” and underneath each button there is a Text Area. When we click on button “J48”, its Prediction accuracies (10- fold cross-validation) are displayed in the corresponding Text area. Similarly when we click on button “ALGO1”, its Prediction accuracies (10- fold cross-validation) are displayed in the corresponding Text area. In the same way when we click on button “ALGO2”, its Prediction accuracies (10- fold cross-validation) are displayed in the corresponding Text area.

4.3 Predicting the level of students and monitoring their performance using Decision Tree Algorithm (Grading System)

4.3.1 Explanation

This particular case study is developed by us. We are doing research on this case study separately. Here we are computing the grade of a student in a class for a particular subject. The system actually combines student attributes and subject attributes and based on the historical data, grade of the student for a particular subject is calculated. The system predicts the grade of a student by constructing a decision tree on the training set through J48 Classifier. The grade can be A,B, C or D. The attributes used for decision making are

Student Attributes

1. Branch Name(cs, it)
2. Mid-Term Marks(out of 30) in this subject
3. Participation in tec. Events related to this subject(1 to 5)
4. Interaction with subject teacher(1 to 5)
5. Library Visit for this subject(1 to 5)
6. Tutorials attend (1 to 5)
7. Lectures attend (1 to 5)
8. Labs attend (1 to 5)
9. Understanding of theory(rate 1 to 5)
10. Understanding of practical(rate 1 to 5)
11. Creativity and effectiveness for this subject (1 to 5)
12. Grade (A,B,C,D)

`

Subject Attributes

1. Subject Name(dbms, data structures, software eng., computer n/w, operating system)
2. Subject scope (placement wise)(1 to 5)
3. Subject scope (Further studies wise)(1to 5)
4. Subject type(core/noncore)
5. Work type(theory/practical/mixed)
6. Faculty(1 to 5)
7. Familiarity(1 to 5)

This one forms more complex example with larger no. of parameters and larger data set. This complex training set formed by conducting the survey on Delhi technological university

4.3.2 Application of Algorithms on the case study

4.3.2.1 Running C4.5 Algorithm on the Training Data

Decision Tree Generated is shown in section 1 of Appendix B.

Algorithm Statistics:

Time to build: 66172 mille seconds

Prediction accuracy (Stratified Cross-Validation using 10 folds):

Fold1: 50%

Fold2: 50%

Fold3: 58.33%

Fold4: 50%

Fold5: 16.66%

Fold6: 41.66%

Fold7: 41.66%

Fold8: 58.33%

Fold9: 66.64%

`

Fold10: 70.58%

Average: 51%

Note: These statistics have been generated by implementing the algorithm using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown in the subsequent sections.

4.3.2.2 Running Algorithm 1 on the Training Data

Decision Tree Generated is shown in section 2 of Appendix B.

Algorithm Statistics:

Time to build: 43203 mille seconds

Prediction accuracy (Stratified Cross-Validation using 10 folds):

Fold1: 75%

Fold2: 58%

Fold3: 41.66%

Fold4: 66.66%

Fold5: 33.33%

Fold6: 58.33%

Fold7: 33.33%

Fold8: 41.66%

Fold9: 75%

Fold10: 70%

Average: 55%

Note: These statistics have been generated by implementing the algorithms using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown in the subsequent sections.

4.3.2.3 Running Algorithm 2 on the Training Data

Decision Tree Generated is shown in section 3 of Appendix B.

`

Algorithm Statistics:

Time to build: 115297 mille seconds

Prediction accuracy (Stratified Cross-Validation using 10 folds):

Fold1: 50%

Fold2: 41%

Fold3: 50%

Fold4: 75%

Fold5: 50%

Fold6: 50%

Fold7: 75%

Fold8: 75%

Fold9: 75%

Fold10: 70%

Average: 62%

Note: These statistics have been generated by implementing the algorithm using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown in the next section.

4.3.3 Analysis and Comparison of Results (With Snap Shots)

If we compare C4.5 and algorithm1, it is clear from table-19 that the decision tree generated by both the algorithms are same in terms of prediction accuracy (through stratified cross-validation using 10 folds) but there is a big difference in time taken to generate the tree.Algorithm1 takes lesser time than C4.5 to generate the decision tree. So Algorithm1 is better than C4.5 in term of time taken to build the tree.

If we compare C4.5 and algorithm2, the decision tree structure is different .Algorithm 2 gives more importance to realistic attributes like midterm marks, subject name, branch name etc and thus is more real and gives more accurate and reasonable results. If we compare the statistics C4.5 and algorithm2 given in table-19, it is clear that the algorithm2 has better prediction accuracy (through stratified cross-validation using 10 folds) as compare to C4.5.

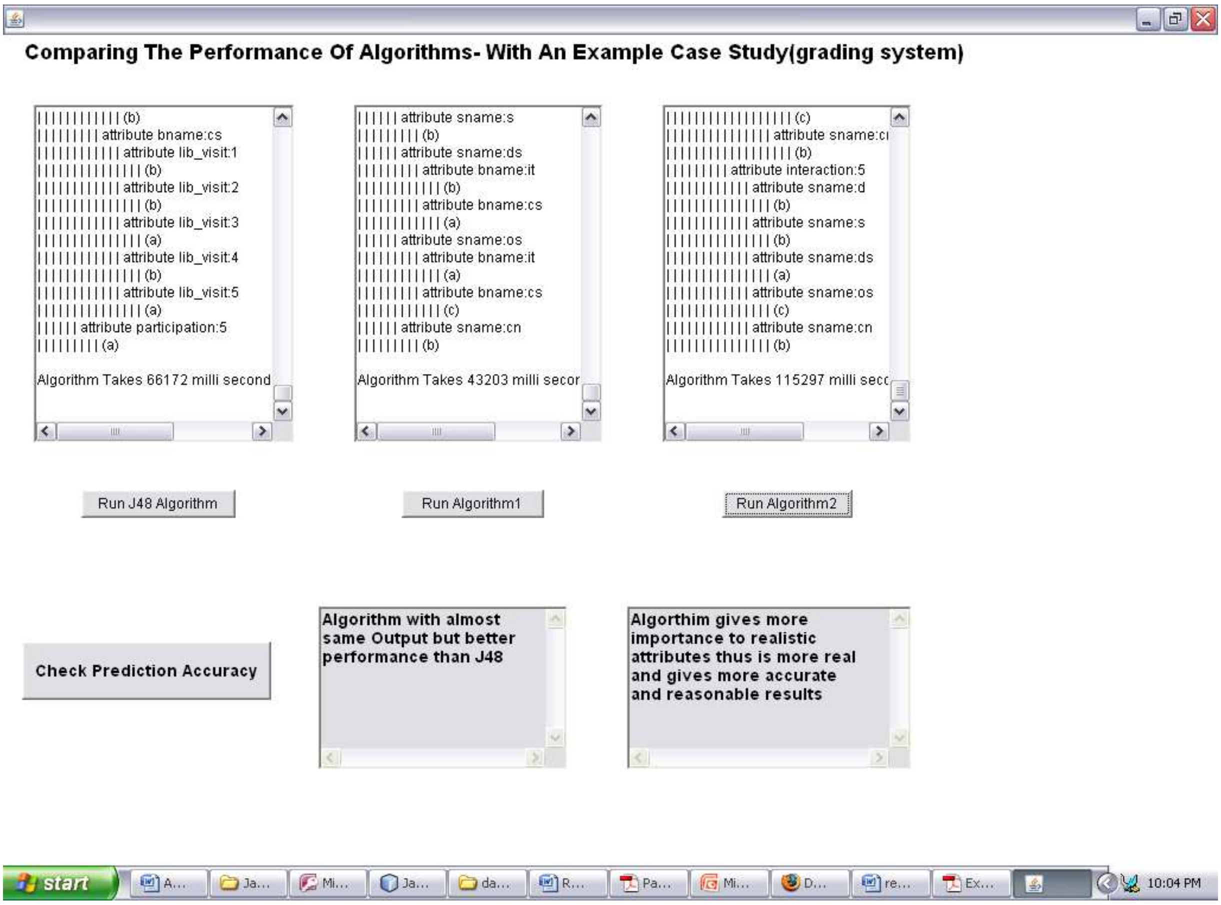
`

Table-19-Comparing build time and prediction accuracy (Grading system)

	Build Time	Prediction accuracy(Cross-Validation Using 10 folds)
C4.5	66172 msec	51%
Algorithm 1	43203 msec	55%
Algorithm 2	1115297 msec	62%

These statistics have been generated by implementing the algorithms using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown as

Snapshot-Time to build the tree



Comparing The Performance Of Algorithms- With An Example Case Study(grading system)

```
| | | | | | | | | | | | (b)
| | | | | | | | | attribute bname:cs
| | | | | | | | | | | | attribute lib_visit:1
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute lib_visit:2
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute lib_visit:3
| | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | attribute lib_visit:4
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute lib_visit:5
| | | | | | | | | | | | | | | (a)
| | | | | | attribute participation:5
| | | | | | | | | (a)

Algorithm Takes 66172 milli second
```

```
| | | | | | attribute sname:s
| | | | | | | | | (b)
| | | | | | attribute sname:ds
| | | | | | | | | attribute bname:it
| | | | | | | | | | | | (b)
| | | | | | | | | attribute bname:cs
| | | | | | | | | | | | (a)
| | | | | | attribute sname:os
| | | | | | | | | attribute bname:it
| | | | | | | | | | | | (a)
| | | | | | | | | attribute bname:cs
| | | | | | | | | | | | (c)
| | | | | | attribute sname:cn
| | | | | | | | | (b)

Algorithm Takes 43203 milli secor
```

```
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute sname:cı
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | attribute interaction:5
| | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | (b)

Algorithm Takes 115297 milli secc
```

Algorithm with almost same Output but better performance than J48

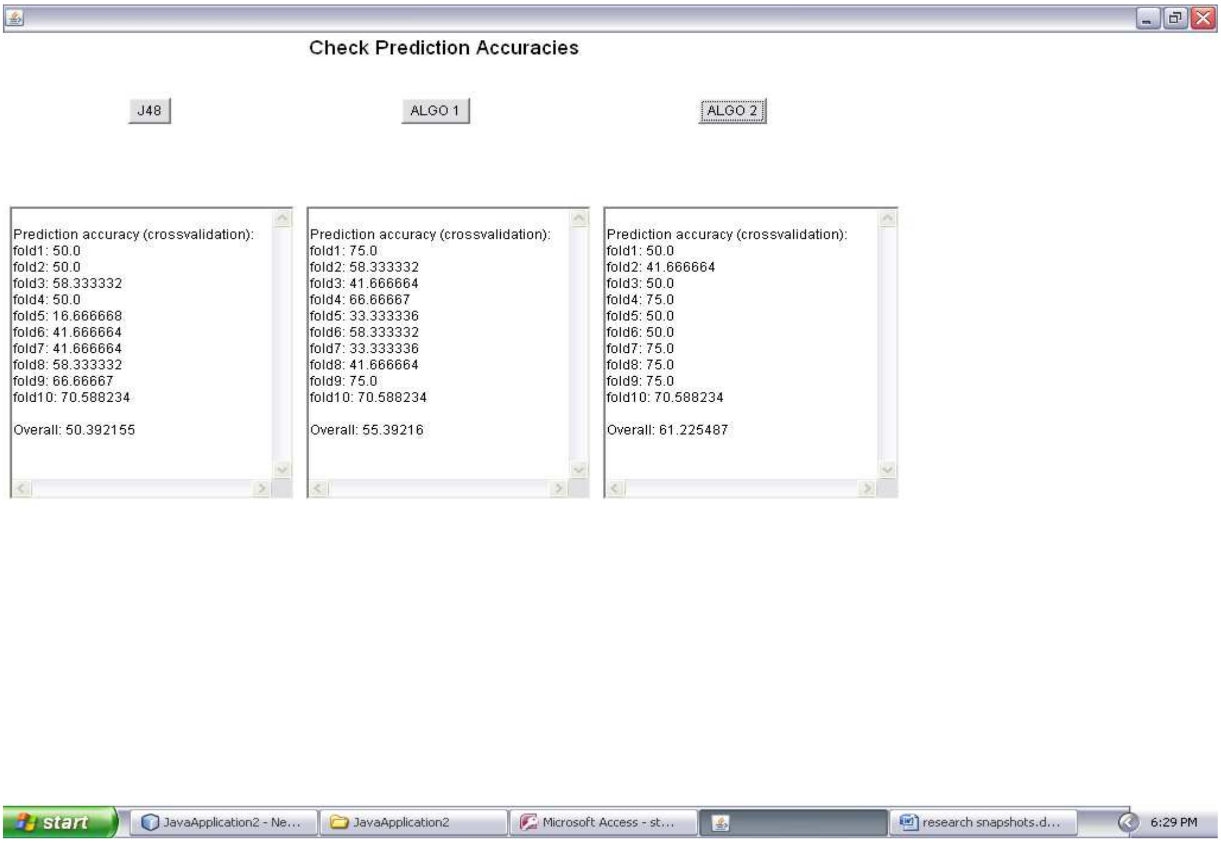
Algorthim gives more importance to realistic attributes thus is more real and gives more accurate and reasonable results

Figure 22- Snapshot-Comparing the execution time of algorithms-with an example/case study (grading system)

`

In the above snapshot there are three buttons “Run J48 Algorithm”, “Run Algorithm 1”, “Run Algorithm2” and above each button there is a Text Area. When we click on button “Run J48 Algorithm”, J48 Tree is displayed in the corresponding Text Area and the time to generate the decision tree is also displayed. Similarly when we click on button “Run Algorithm 1”, a decision tree is displayed in the corresponding Text Area and the time to generate the decision tree is also displayed. In the same way when we click on button “Run Algorithm 1”, a decision tree is displayed in the corresponding Text Area and the time to generate the decision tree is also displayed.

Snapshot-Prediction accuracy (CROSS VALIDATION-10 FOLD)



```
Prediction accuracy (crossvalidation):
fold1: 50.0
fold2: 50.0
fold3: 58.333332
fold4: 50.0
fold5: 16.666668
fold6: 41.666664
fold7: 41.666664
fold8: 58.333332
fold9: 66.66667
fold10: 70.588234

Overall: 50.392155
```

```
Prediction accuracy (crossvalidation):
fold1: 75.0
fold2: 58.333332
fold3: 41.666664
fold4: 66.66667
fold5: 33.333336
fold6: 58.333332
fold7: 33.333336
fold8: 41.666664
fold9: 75.0
fold10: 70.588234

Overall: 55.39216
```

```
Prediction accuracy (crossvalidation):
fold1: 50.0
fold2: 41.666664
fold3: 50.0
fold4: 75.0
fold5: 50.0
fold6: 50.0
fold7: 75.0
fold8: 75.0
fold9: 75.0
fold10: 70.588234

Overall: 61.225487
```

Figure 23- Snapshot-Comparing the prediction accuracies of algorithms-with an example/case study (grading system)

In the above snapshot, there are three buttons, “J48”, “ALGO1”, “ALGO2” and underneath each button there is a Text Area. When we click on button “J48”, its Prediction accuracies (10- fold cross-validation) are displayed in the corresponding Text area. Similarly when we click on button “ALGO1”, its Prediction accuracies (10- fold cross-validation) are displayed in the corresponding Text area. In the same way when we click on button “ALGO2”, its Prediction accuracies (10-fold cross-validation) are displayed in the corresponding Text area.

`

4.4 Predicting the eligibility of students regarding AIEEE Counseling (Based on Real AIEEE data)

4.4.1 Explanation

The sole purpose of this case study is to test the proposed algorithms. This case study is based on real AIEEE data. Here we are predicting student’s eligibility to attend AIEEE counselling i.e. whether student is eligible to attend AIEEE Counselling or not. The Decision regarding student’s eligibility is estimated based on the following attributes.

- Course(BTECH/BARCH)
- Category(SC/ST/OBC/GEN/PH)
- Birth_year (86 to 90)
- Gender(M/F)
- Marks in Test(1 to 250)
- All India Rank (1 to 3 lakh)
- State Rank(1 to 20k)
- Decision(ELIGIBLE/ NOT ELIGIBLE/ ELIGIBLE-CATEGORY SPECIFIC)

The above attributes are extracted from Real AIEEE Data which I got from NIC (National Informatics centre). The Data set is very large complex containing data in the range of thousands.

4.4.2 Application of Algorithms on the case study

4.4.2.1 Running C4.5 Algorithm on the Training Data

Decision Tree Generated is shown in section 1 of Appendix C.

Algorithm Statistics:

Time to build: 30172 mille seconds

Prediction accuracy (Stratified Cross-Validation using 10 folds):

Fold1: 99%

Fold2: 95%

Fold3: 97%

Fold4: 97%

Fold5: 90%

Fold6: 94%

`

Fold7: 97%

Fold8: 99%

Fold9: 99%

Fold10: 100%

Average: 96.7%

Note: These statistics have been generated by implementing the algorithm using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown in the subsequent sections.

4.4.2.2 Running Algorithm 1 on the Training Data

Decision Tree Generated is shown in section 2 of Appendix C.

Algorithm Statistics:

Time to build: 27578 mille seconds

Prediction accuracy (Stratified Cross-Validation using 10 folds):

Fold1: 100%

Fold2: 96%

Fold3: 97%

Fold4: 95%

Fold5: 98%

Fold6: 92%

Fold7: 98%

Fold8: 98%

Fold9: 98%

Fold10: 100%

Average: 97.2%

`

Note: These statistics have been generated by implementing the algorithm using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown in the subsequent sections.

4.4.2.3 Running Algorithm 2 on the Training Data

Decision Tree Generated is shown in section 3 of Appendix C.

Algorithm Statistics:

Time to build: 108484 mille seconds

Prediction accuracy (Stratified Cross-Validation using 10 folds):

Fold1: 98%

Fold2: 95%

Fold3: 98%

Fold4: 95%

Fold5: 98%

Fold6: 99%

Fold7: 98%

Fold8: 98%

Fold9: 100%

Fold10: 98.20%

Average: 97.8%

Note: These statistics have been generated by implementing the algorithm using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown in the next section.

4.4.3 Analysis and Comparison of Results (With Snap Shots)

If we compare C4.5 and algorithm1, it is clear from table-20 that the decision tree generated by both the algorithms are same in terms of prediction accuracy (through stratified cross-validation using 10 folds) but there is a big difference in time taken to generate the

`

tree.Algorithm1 takes lesser time than C4.5 to generate the decision tree. So Algorithm1 is better than C4.5 in term of time taken to build the tree.

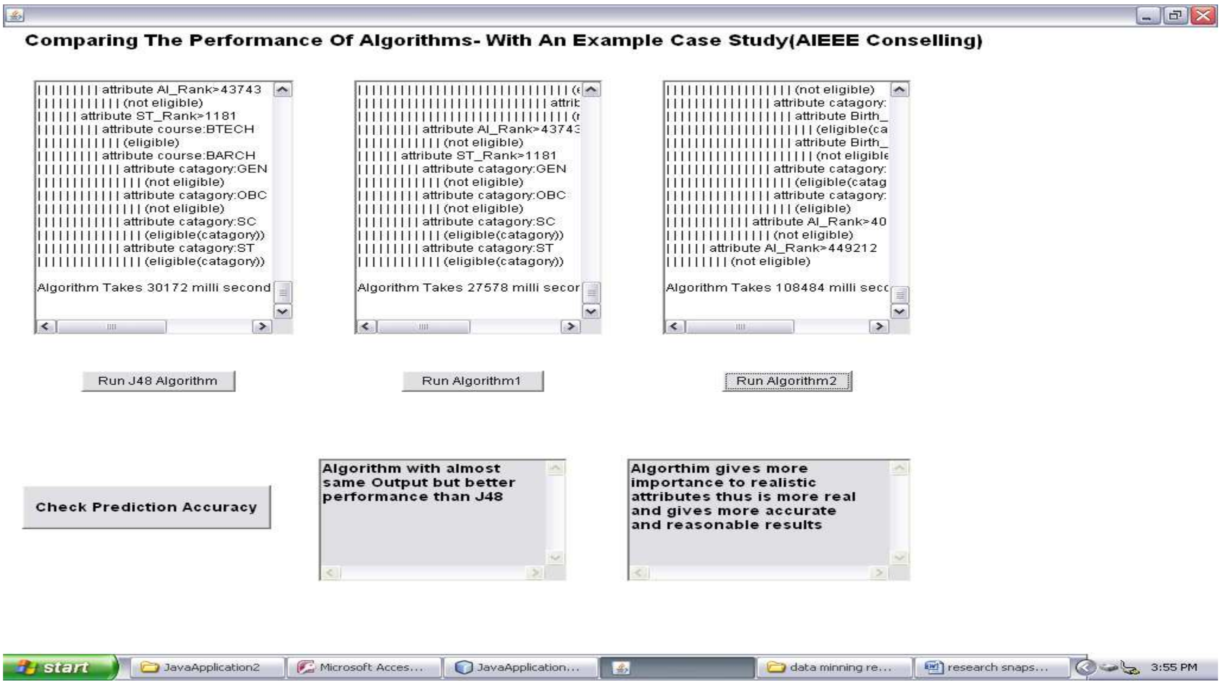
If we compare C4.5 and algorithm2, the decision tree structure is different .Algorithm 2 gives more importance to realistic attributes like All India Rank, marks in test and category and thus is more real and gives more accurate and reasonable results. If we compare the statistics C4.5 and algorithm2 given in table-20, it is clear that the algorithm2 has better prediction accuracy (through stratified cross-validation using 10 folds) as compare to C4.5.

Table-20-Comparing build time and prediction accuracy (AIEEE Counseling)

	Build Time	**Prediction accuracy(Cross-Validation Using 10 folds)**
C4.5	30172 msec	96.7%
Algorithm 1	27578 msec	97.2%
Algorithm 2	108484 msec	97.8%

These statistics have been generated by implementing the algorithms using JAVA and without using any data mining tool like WEKA. Snapshots (outputs) of implementation are shown as

Snapshot-Time to build the tree



Comparing The Performance Of Algorithms- With An Example Case Study(AIEEE Conselling)

```
||||||||| attribute AI_Rank>43743
|||||||||||| (not eligible)
|||||| attribute ST_Rank>1181
||||||||| attribute course:BTECH
|||||||||||| (eligible)
||||||||| attribute course:BARCH
|||||||||||| attribute catagory:GEN
||||||||||||||| (not eligible)
|||||||||||| attribute catagory:OBC
||||||||||||||| (not eligible)
|||||||||||| attribute catagory:SC
||||||||||||||| (eligible(catagory))
|||||||||||| attribute catagory:ST
||||||||||||||| (eligible(catagory))

Algorithm Takes 30172 milli second
```

```
|||||||||||||||||||||||||||||| (e
|||||||||||||||||||||||||||| attrib
|||||||||||||||||||||||||||||| (r
||||||||| attribute AI_Rank>43743
|||||||||||| (not eligible)
|||||| attribute ST_Rank>1181
||||||||| attribute catagory:GEN
|||||||||||| (not eligible)
||||||||| attribute catagory:OBC
|||||||||||| (not eligible)
||||||||| attribute catagory:SC
|||||||||||| (eligible(catagory))
||||||||| attribute catagory:ST
|||||||||||| (eligible(catagory))

Algorithm Takes 27578 milli secor
```

```
||||||||||||||||||| (not eligible)
|||||||||||||||| attribute catagory:
||||||||||||||||||| attribute Birth_
||||||||||||||||||||||| (eligible(ca
||||||||||||||||||| attribute Birth_
||||||||||||||||||||||| (not eligible
|||||||||||||||| attribute catagory:
||||||||||||||||||| (eligible(catag
|||||||||||||||| attribute catagory:
||||||||||||||||||| (eligible)
||||||||||||| attribute AI_Rank>40
|||||||||||||||| (not eligible)
|||||| attribute AI_Rank>449212
|||||||||| (not eligible)

Algorithm Takes 108484 milli secc
```

Algorithm with almost same Output but better performance than J48

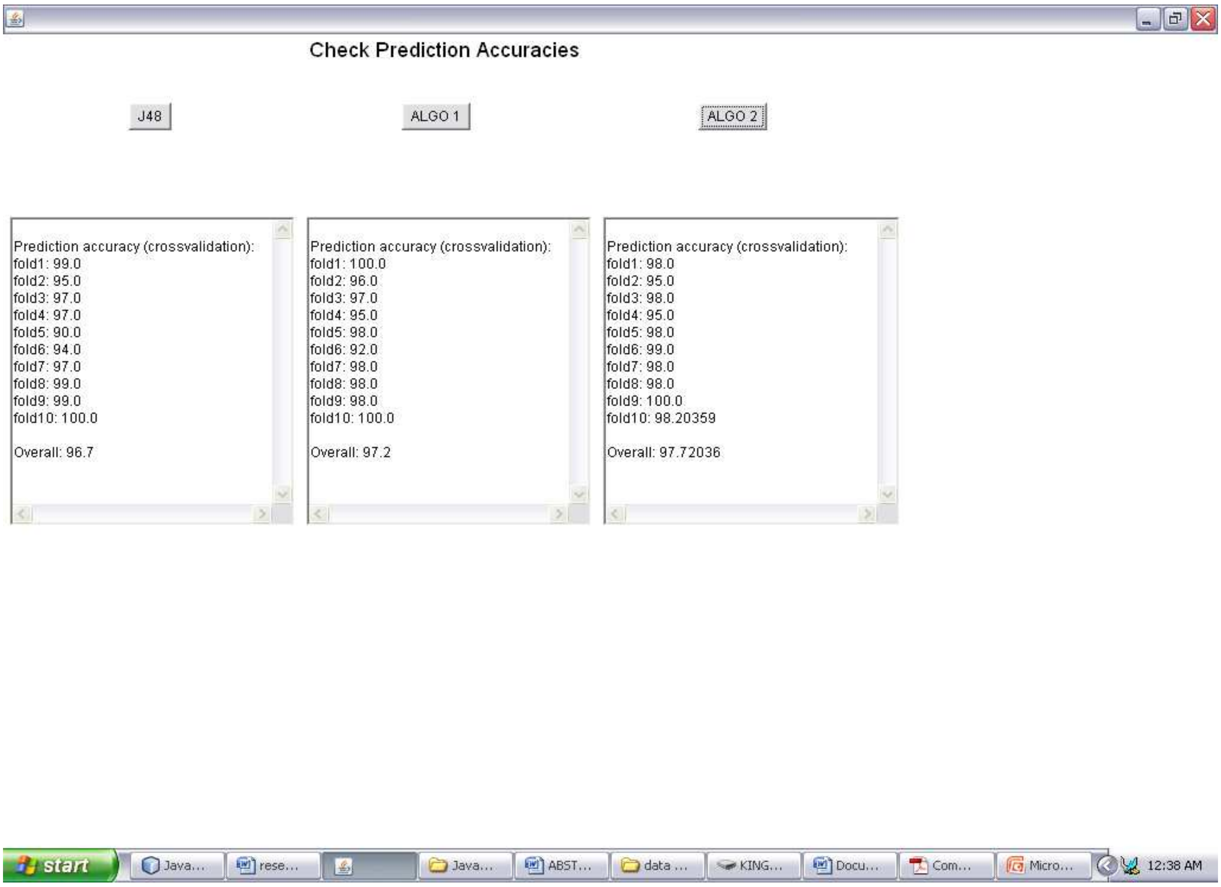
Algorthim gives more importance to realistic attributes thus is more real and gives more accurate and reasonable results

Figure 24- Snapshot-Comparing the execution time of algorithms-with an example/case study (AIEEE Counseling)

`

In the above snapshot there are three buttons “Run J48 Algorithm”, “Run Algorithm 1”, “Run Algorithm2” and above each button there is a Text Area. When we click on button “Run J48 Algorithm”, J48 Tree is displayed in the corresponding Text Area and the time to generate the decision tree is also displayed. Similarly when we click on button “Run Algorithm 1”, a decision tree is displayed in the corresponding Text Area and the time to generate the decision tree is also displayed. In the same way when we click on button “Run Algorithm 1”, a decision tree is displayed in the corresponding Text Area and the time to generate the decision tree is also displayed.

Snapshot-Prediction accuracy (CROSS VALIDATION-10 FOLD)



```
Prediction accuracy (crossvalidation):
fold1: 99.0
fold2: 95.0
fold3: 97.0
fold4: 97.0
fold5: 90.0
fold6: 94.0
fold7: 97.0
fold8: 99.0
fold9: 99.0
fold10: 100.0

Overall: 96.7
```

```
Prediction accuracy (crossvalidation):
fold1: 100.0
fold2: 96.0
fold3: 97.0
fold4: 95.0
fold5: 98.0
fold6: 92.0
fold7: 98.0
fold8: 98.0
fold9: 98.0
fold10: 100.0

Overall: 97.2
```

```
Prediction accuracy (crossvalidation):
fold1: 98.0
fold2: 95.0
fold3: 98.0
fold4: 95.0
fold5: 98.0
fold6: 99.0
fold7: 98.0
fold8: 98.0
fold9: 100.0
fold10: 98.20359

Overall: 97.72036
```

Figure 25- Snapshot-Comparing the prediction accuracies of algorithms-with an example/case study (AIEEE Counseling)

In the above snapshot, there are three buttons, “J48”, “ALGO1”, “ALGO2” and underneath each button there is a Text Area. When we click on button “J48”, its Prediction accuracies (10- fold

`

cross-validation) are displayed in the corresponding Text area. Similarly when we click on button “ALGO1”, its Prediction accuracies (10- fold cross-validation) are displayed in the corresponding Text area. In the same way when we click on button “ALGO2”, its Prediction accuracies (10-fold cross-validation) are displayed in the corresponding Text area.

`

5. IMPLEMENTATION

5.1 Tools and software used

- Java Platform Standard Edition 6 Development Kit (JDK 6): - JDK 6 provides a run time environment and other utilities to develop, compile, execute and edit programs written in the Java programming language. It can be downloaded from Oracle Sun website.

- NetBeans IDE 6.9.1: -NetBeans IDE(Integrated Development Environment) allows you to develop Java SE, web, and Java EE enterprise applications. The GUI Builder enables you to visually design Java desktop (AWT and Swing) applications. It includes special support for GroupLayout and the Swing Application Framework (JSR 296). It provides the general infrastructure for handling Java code in the IDE. Includes features such as syntax highlighting, error marking, code completion, code templates, refactoring, and other coding productivity features. It can be downloaded free of cost from NetBeans website.

- Microsoft Access: - Microsoft access has been used to make relations which act as training data sets. It is a very light weight database and provides all the basic database utilities .

5.2 Relations used

For each case study, one relation is used which acts as a training data for that case study/ example.

1. **For Weather case study**
   The relation used is “weather” and the attributes are
   - ID
   - outlook
   - temperature
   - humidity
   - windy
   - decision

2. **For case study-counselling system**
   The relation used is “student_branch” and the attributes are
   - ID
   - perspective
   - familypressure
   - physics
   - chem

`

- maths
- fav
- familyback
- fath_prof
- worker
- creative
- branch
- scope
- placement
- work_pressure
- type
- study_scope
- syalbus
- type_work
- facility
- familarity
- satisfaction

3. **For case study-Grading system**

   The relation used is “student_grade” and the attributes are

   - ID
   - bname
   - sname
   - midterm_marks
   - participation
   - interaction
   - lib_visit
   - tutorials
   - lectures
   - labs
   - understanding_theory
   - understanding_practical
   - creativity_and_effectiveness
   - scope_placement
   - scope_furtherstudies
   - Subject_type
   - work_type
   - faculty
   - familiarity
   - grade

4. **For case study-AIEEE Counselling**

   The relation used is “AIEEE” and the attributes are

   - ID
   - course

`

- Gender
- catagory
- Birth_year
- Marks
- AI_Rank
- ST_Rank
- Decision

There is one common relation "domain" which is used in all the case studies. Its attributes are

- ID
- Attribute name
- Attribute type
- Value 1
- Value 2
- Value 3
- Value 4
- Value 5

Note: if attribute is a discrete attribute and it has only 2 unique values then fields "value 4" and "value 5" are kept blank and 0 is inserted in field "value 3" which represents end of domain. If attribute is a continuous attribute then field "value 1" contains lowest value and "value 2" contains maximum value of attribute.

All the above relations are stored in the same Microsoft access database named student.accdb"

5.3 Data source creation

We must create an ODBC data source. Before we create the data source, we must have the ODBC drivers for Microsoft Access properly installed. The "student" database must be an ODBC data source defined in the ODBC Administrator. Additionally, the data source must be a system DSN.

1. From the Control Panel or an ODBC group, start the ODBC Administrator. When the ODBC Administrator starts, you will see the Data Sources dialog box.
2. Install the new data source by clicking on the System DSN button in the Data Sources dialog box. The System Data Sources dialog box should appear. In the System Data Sources dialog box, click on the Add button. The Add Data Source dialog box will then appear.
3. In the Add Data Source dialog box, select the Microsoft Access ODBC driver and click on the OK button. The ODBC Microsoft Access Setup dialog box should appear. This dialog box is used to set up the data source.
4. Give your data source a name by typing *student* into the Data Source Name text box. This is the name you will use to identify the database whenever you utilize an ODBC driver. This name can be anything you want and is not directly related to the name of the database.

`

5. Enter a brief description for your data source in the Description text box. This can be any text you want and has no effect on the application.
6. Establish your data source as the database you copied previously by clicking on the Select button. Now you will see a dialog box that lets you select the file for this data source. Navigate to the PROJECT folder in which you copied the database and select student.accdb. When you are done, click on the OK button. Your new data source should now be visible in the System Data Sources dialog box.

5.4 Program Architecture

Same program architecture is followed for all the case studies. Because of the space limitations it is not possible to explain the program architecture for all the case studies. So I have chosen one out of four case studies i.e. **AIEEE Counselling (Based on real AIEE Data)** to explain the program architecture.

5.4.1 Program Modules

- **Algorithm_comparison_AIEEE.java-** It's a home page. There are three buttons "Run J48 Algorithm", "Run Algorithm 1", "Run Algorithm2" and above each button there is a Text Area. There is an another button "check prediction accuracy".

- **J48_AIEEE-** This module contains the code for C4.5/J48 algorithm. It is responsible for generating J48/C4.5 decision tree and the time taken to build the tree.

- **Algo1_AIEEE**- This module contains the code for algorithm 1. It is responsible for generating algorithm1 decision tree and the time taken to build the tree.

- **Algo2_AIEEE**- This module contains the code for algorithm 2. It is responsible for generating algorithm2 decision tree and the time taken to build the tree.

- **Prediction_accuracy_AIEEE-** This page contains three buttons "J48", "ALGO1", "ALGO2" and below each button there is text area.

- **Prediction_accurracy_AIEEE_J48-** This module is responsible for computing prediction accuracies -10 fold cross validation of J48/C4.5 decision tree.

- **Prediction_accuracy_AIEEE_algo1-** This module is responsible for computing prediction accuracies -10 fold cross validation of algorithm1 decision tree.

- **Prediction_accuracy_AIEEE_algo2-** This module is responsible for computing prediction accuracies -10 fold cross validation of algorithm1 decision tree.

The Code for above modules is not mentioned in the thesis because of the size problems but it can be provided by the author on request.

`

5.4.2 Top-down design

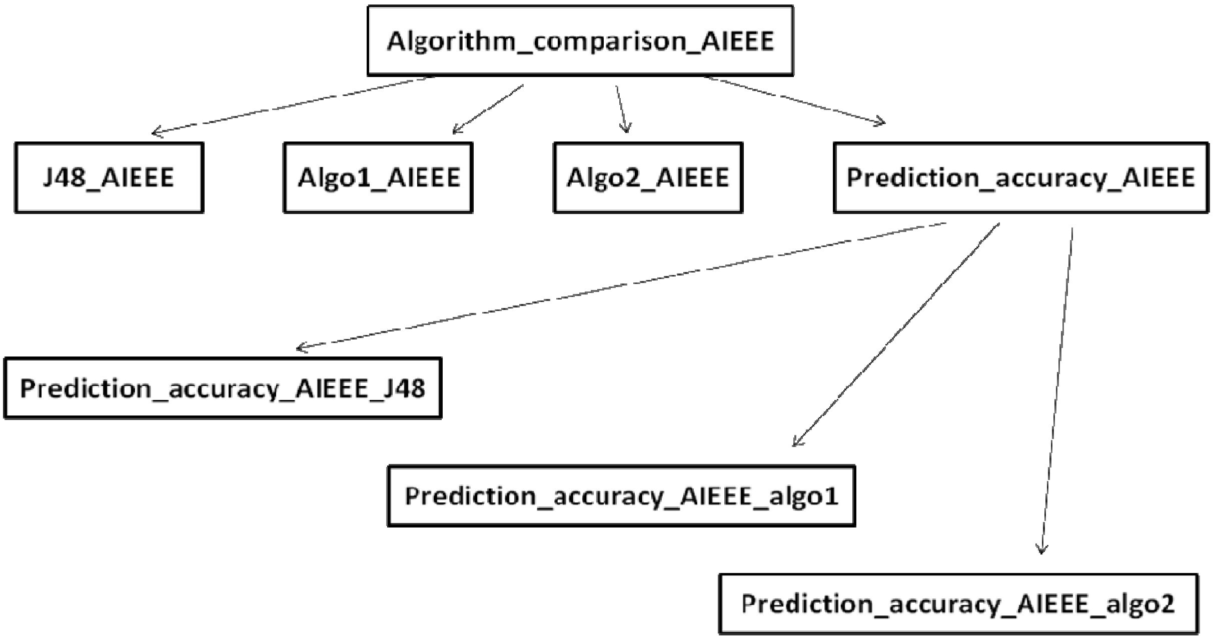

Algorithm_comparison_AIEEE
J48_AIEEE
Algo1_AIEEE
Algo2_AIEEE
Prediction_accuracy_AIEEE
Prediction_accuracy_AIEEE_J48
Prediction_accuracy_AIEEE_algo1
Prediction_accuracy_AIEEE_algo2

Figure-26-Top-down design of the application (AIEEE Counseling)

5.4.3 Step by Step Execution of Modules (With Snapshots)

We run the application by first running the main program i.e “Algorithm_comparison_AIEEE“

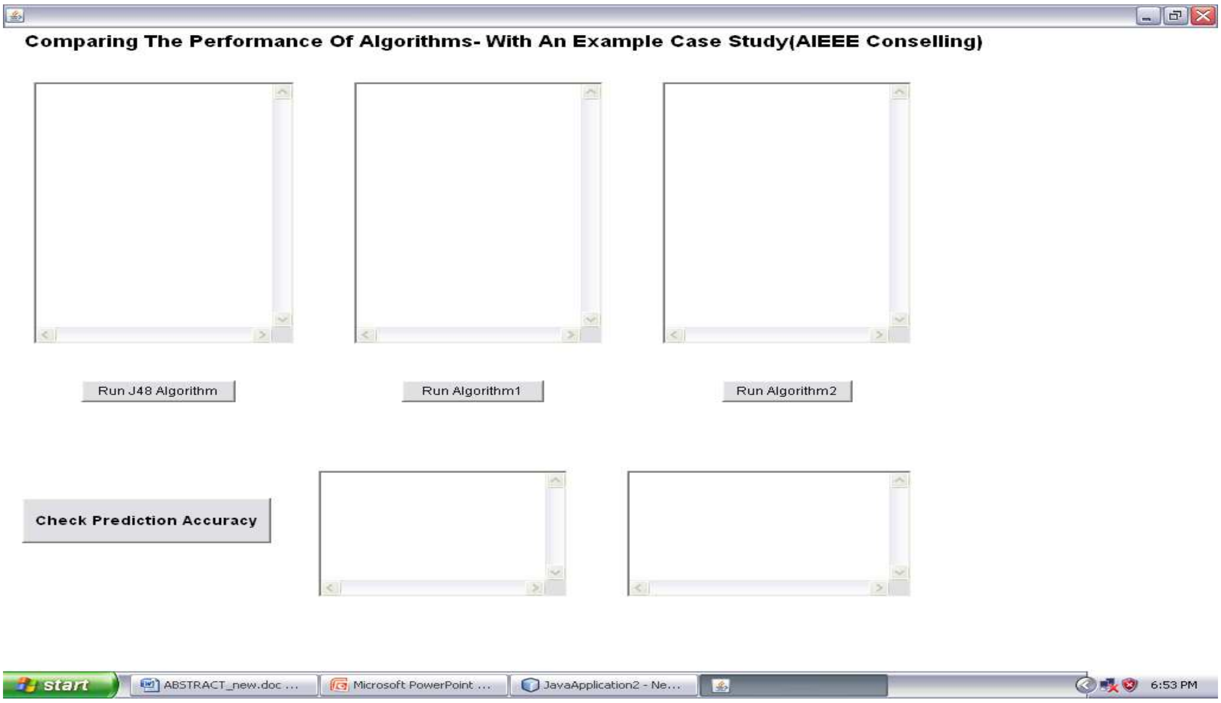


Figure-27-Snapshot 1 (AIEEE Counseling)

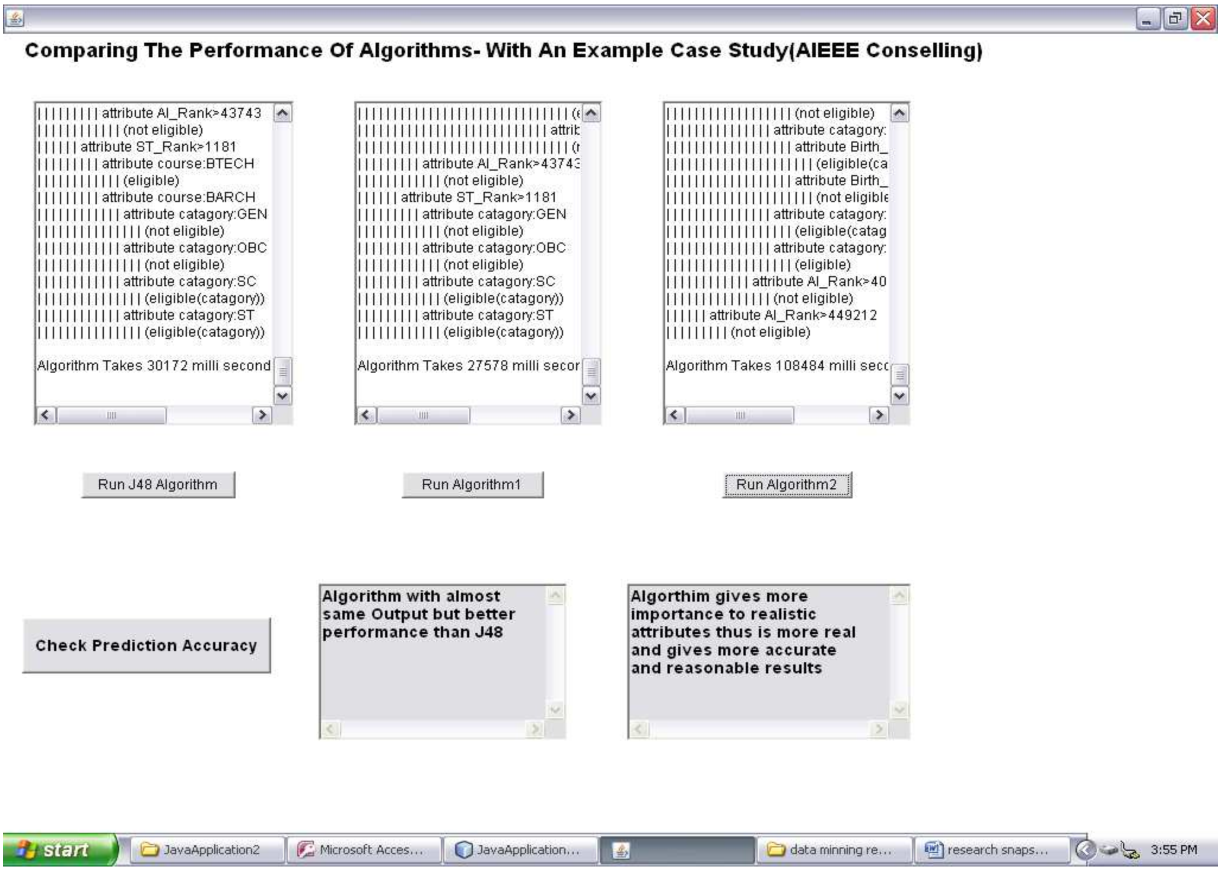
`

In the above snapshot, there are four buttons

- Run J48 Algorithm-On clicking on this button, “J48_AIEEE” module is called and J48/C4.5 tree is displayed on the above text area.
- Run Algorithm1- On clicking on this button, “algo1_AIEEE” module is called and algorithm 1 decision tree is displayed on the above text area.
- Run Algorithm2- On clicking on this button, “algo2_AIEEE” module is called and algorithm2 decision tree is displayed on the above text area

  Here is the snapshot

.



Comparing The Performance Of Algorithms- With An Example Case Study(AIEEE Conselling)

```
||||||||| attribute AI_Rank>43743
|||||||||||| (not eligible)
|||||| attribute ST_Rank>1181
||||||||| attribute course:BTECH
|||||||||||| (eligible)
||||||||| attribute course:BARCH
|||||||||||| attribute catagory:GEN
||||||||||||||| (not eligible)
|||||||||||| attribute catagory:OBC
||||||||||||||| (not eligible)
|||||||||||| attribute catagory:SC
||||||||||||||| (eligible(catagory))
|||||||||||| attribute catagory:ST
||||||||||||||| (eligible(catagory))

Algorithm Takes 30172 milli second
```

```
|||||||||||||||||||||||||||||| (e
||||||||||||||||||||||||||| attrib
|||||||||||||||||||||||||||||| (r
||||||||| attribute AI_Rank>43743
|||||||||||| (not eligible)
|||||| attribute ST_Rank>1181
||||||||| attribute catagory:GEN
|||||||||||| (not eligible)
||||||||| attribute catagory:OBC
|||||||||||| (not eligible)
||||||||| attribute catagory:SC
|||||||||||| (eligible(catagory))
||||||||| attribute catagory:ST
|||||||||||| (eligible(catagory))

Algorithm Takes 27578 milli secor
```

```
|||||||||||||||||| (not eligible)
||||||||||||||| attribute catagory:
|||||||||||||||||| attribute Birth_
||||||||||||||||||||| (eligible(ca
|||||||||||||||||| attribute Birth_
||||||||||||||||||||| (not eligible
||||||||||||||| attribute catagory:
|||||||||||||||||| (eligible(catag
||||||||||||||| attribute catagory:
|||||||||||||||||| (eligible)
|||||||||||| attribute AI_Rank>40
||||||||||||||| (not eligible)
|||||| attribute AI_Rank>449212
||||||||| (not eligible)

Algorithm Takes 108484 milli secc
```

Algorithm with almost same Output but better performance than J48

Algorthim gives more importance to realistic attributes thus is more real and gives more accurate and reasonable results

Figure-28-Snapshot 2 (AIEEE Counseling)

- Check Prediction_Accuracy- On clicking on this button, “**Prediction_accuracy_AIEEE”** module is called and a page with three buttons “J48”, “ALGO1”, “ALGO2” is opened with three text areas.

`

Here is the snapshot

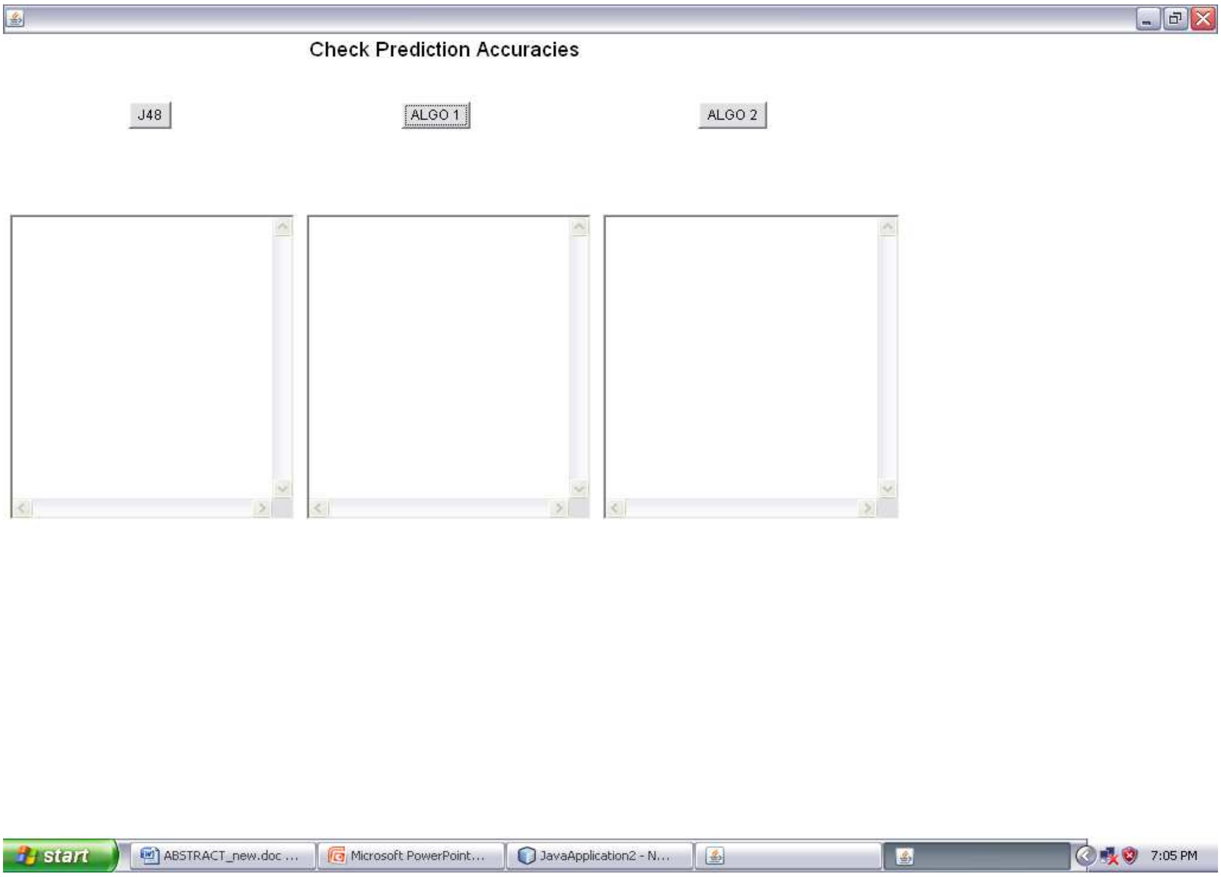


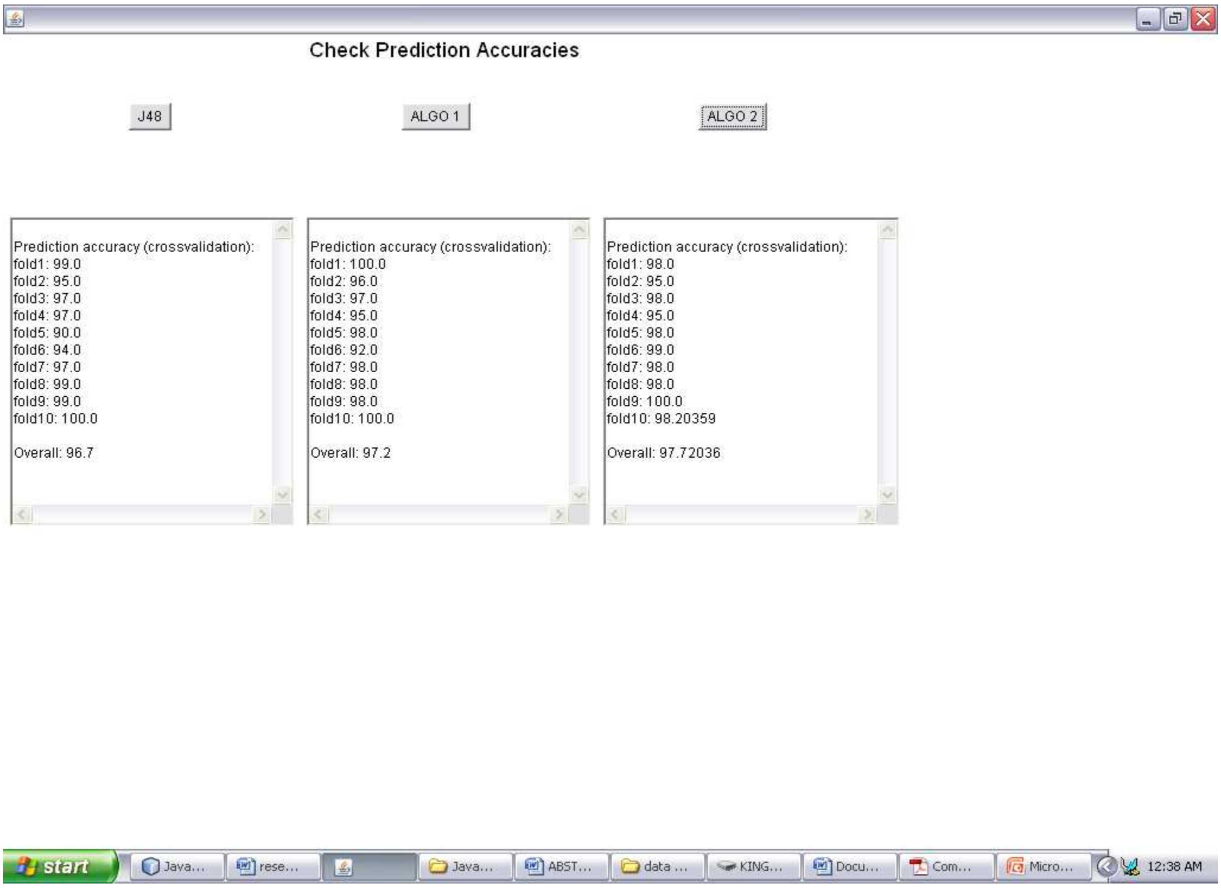
Figure-29-Snapshot 3 (AIEEE Counseling)

In the above snapshot, there are three buttons

- J48-On clicking on this button , "**Prediction_accuracy_AIEEE_J48"** module is called and prediction accuracy(10-fold cross validation) of J48/C4.5 decision tree is outputted in the corresponding text area
- ALGO1- On clicking on this button, "**Prediction_accuracy_AIEEE_algo1"** module is called and prediction accuracy(10-fold cross validation) of algorithm1 decision tree is outputted in the corresponding text area
- ALGO2- On clicking on this button , "**Prediction_accuracy_AIEEE_algo2"** module is called and prediction accuracy(10-fold cross validation) of algorithm2 decision tree is outputted in the corresponding text area

`

Here is the snapshot



```
Prediction accuracy (crossvalidation):
fold1: 99.0
fold2: 95.0
fold3: 97.0
fold4: 97.0
fold5: 90.0
fold6: 94.0
fold7: 97.0
fold8: 99.0
fold9: 99.0
fold10: 100.0

Overall: 96.7
```

```
Prediction accuracy (crossvalidation):
fold1: 100.0
fold2: 96.0
fold3: 97.0
fold4: 95.0
fold5: 98.0
fold6: 92.0
fold7: 98.0
fold8: 98.0
fold9: 98.0
fold10: 100.0

Overall: 97.2
```

```
Prediction accuracy (crossvalidation):
fold1: 98.0
fold2: 95.0
fold3: 98.0
fold4: 95.0
fold5: 98.0
fold6: 99.0
fold7: 98.0
fold8: 98.0
fold9: 100.0
fold10: 98.20359

Overall: 97.72036
```

Figure-30-Snapshot 4 (AIEEE Counseling)

`

6. CONCLUSION AND FUTURE WORK

I have analyzed C4.5 algorithm and tried to come out with the methods to improve build time and prediction accuracy. I have taken four case studies to prove that our algorithms not only show better behavior on smaller and simpler data sets but also on more complex and larger datasets.

Two conclusions can be drawn

1. Algorithm 1 has better time complexity than C4.5 Algorithm

C4.5 Algorithm is based on assumption that, the attribute that results in a maximum disordered state is the best attribute, the one that brings us closest to the final classification. To find out that attribute, it uses the concept of information entropy. The proposed heuristic function used in algorithm 1 is also based on the same assumption but the difference is in the measure of disorderness. C4.5 uses information entropy as the measure of disorderness. I am using a much simplified measure i.e calculating the maximum probability in the distribution. It can be considered as the approximate measure of disorderness. So introduction of this Heuristic function gives rise to an improved Algorithm which is better than C4.5 in terns of time complexity. The detailed reasoning is given in section 3.1.1.

Also from the case studies presented in section 4 we conclude that the decision tree generated by C4.5 and algorithm1 are very much the same in terms of structure or the prediction accuracy (on training set, supplied set, stratified cross-validation) or both but there is a big difference in time taken to generate the tree.Algorithm1takes lesser time than C4.5 to generate the decision tree.

2. Algorithm 2 has better prediction accuracy than C4.5 Algorithm

Unlike C4.5 this Heuristic Function is not based on the assumption that a smaller decision tree is a better decision tree in terms of prediction accuracy. It is based on the fact that a decision tree with realistic attributes at the top is a much better tree in terms of prediction accuracy. Sometimes an attribute individually is not that much important in decision making but when combined with other attributes becomes important in decision making. Unlike C4.5, this Heuristic function takes all the combinations of attributes while measuring the importance of an attribute in decision making. The value of Information gain ratio (Heuristic function of C4.5) computed for any attribute depends only that attribute and the decision attribute. But in this case since Heuristic function is taking in to account all the combinations of attributes while measuring the importance of attribute in decision making, so the value depends upon the other attributes as well. This nature of Heuristic function results in more Real Decision Tree and thus gives more accurate and reasonable results. The detailed reasoning is given in section 3.1.2.

`

Also from the case studies presented in section 4, if we compare C4.5 and algorithm2, although the decision tree structure is different and time taken by algorithm 2 to generate the decision tree is more than that of C4.5 but algorithm2 has better prediction accuracy (on training set, supplied set and stratified cross validation) as compare to C4.5. Algorithm 2 gives more importance to realistic attributes and thus is more real and gives more accurate and reasonable results.

Future Work

I have mainly focused on 2 performance measures 1) Time to build the tree, 2) Prediction accuracy. I have tried to improve these two things. These improvements can have a significant impact on the practical applications. We want the practical applications to be solved more efficiently and effectively. Two types of applications always exist in the real word. First type of applications is the one where results need to be generated faster e.g. real estate market prediction or share market prediction. Second ones are the critical applications where results of classification need to be more accurate e.g.: predicting student's behavior in education, medical applications where doctor needs to predict a disease based on the symptoms of the patient, business applications where a marketing professional needs complete description of customer segments to successfully launch a marketing campaign etc. So algorithms have been developed keeping in mind these two types of applications. In future, I will focus on improving the algorithms further.

`

7. REFERENCES

[1]. Zhu Xiaoliang, Wang Jian, YanHongcan, Wu Shangzhuo (2009) ,“Research and Application of the improved Algorithm C4.5 on Decision Tree”, 2009 International Conference on Test and Measurement

[2]. Linna Li, , Xuemin Zhang “,“Study of Data Mining Algorithm Based on Decision Tree “,201O International Conference On Computer Design And Appliations (ICCDA 2010)

[3]. J. R. Quinlan. “Improved use of continuous attributes inc4.5.”Journal of Artificial Intelligence Research, 4:77-90, 1996.

[4]. “The Improvement of C4.5 Algorithm and Case Study” LI Rui, WEI Xian-mei ,YU Xue-wei ,Software Institute Dalian Jiao tong “University Dalian, China,(2009).

[5]. Weizhao Guo and Jian Yin, Zhimin Yang*, Xiaobo Yang, and Li Huang, Exploring an Improved Decision Tree Based Weights, 2009 Fifth International Conference on Natural Computation

[6]. “CAKE – Classifying, Associating & Knowledge DiscovEry An Approach for Distributed Data Mining (DDM) Using PArallel Data Mining Agents (PADMAs)”,Danish Khan, 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

[7]. Yan Chen1, Ming Yang2, Lin Zhang3,” General Data Mining Model System Based on Sample Data Division”, 2009 Second International Symposium on Knowledge Acquisition and Modeling

[8]. Mouhib Al-Noukari, Wael Al-Hussan, Using Data Mining Techniques for Predicting Future Car market Demand”,DCX Case Study

[9]. Data Mining Techniques: Classification and Clustering ,Bamshad Mobasher School of CTI, DePaul University

[10]. Witten IH, Frank E. Data Mining: Practical Machine Learning Tools and Techniques. Second edition, 2005. Morgan Kaufmann. Han J, Kamber M. Data Mining: Concepts and Techniques. Second edition, 2006. Morgan Kaufmann

[11]. Privacy Preserving ID3 using Gini Index over Horizontally Partitioned Data, Saeed Samet, Ali Miri.

[12]. Rough Set and CART Approaches to Mining Incomplete Data, Jerzy W. Grzymala-Busse

[13]. Nermin Ozgulbas, Ali Serhan Koyuncugil, Developing Road Maps for Financial Decision Making by CHAID Decision Tree, 2009 International Conference on Information Management and Engineering

[14]. “Forecasting the Hourly Ontario Energy Price by Multivariate Adaptive Regression Splines”, H. Zareipour, Student Member, IEEE, K. Bhattacharya, Senior Member, IEEE, C.A. Ca˜nizares, Senior Member,I EEE.

`

[15]. Feng Shaorong:Research and Improvement of Decision Trees Algorithm .Journal of XiamenUniversity (NaturalScience), 2007,pp.498-500.

[16]. Zhang Jun ,Yang Xuebing: Decision Tree and Its Key Techniques. Computer Technology and Development.2006, pp.44-46.

[17]. Sun Chaili: Study on Data Stream M in ing Algorithm Based on Decision Tree Journal of Taiyuan University of Science and Technology.,2006, pp.269-270.

[18]. Mobasher B,Cooley R,Jaideep S,etal. Comments on decision tree. New York:IEEE Press,1999.

[19]. Shahabi C, Zarkesh A M, Adibi J, et al. Introduction of neutral network [C]. .B inningham: IEEE Press,200.

[20]. Yan T,JacobesnM,Garcia-Mo Lina H,et al Introduction of genetic algorithm. [C]. :Paris WAM Press,1999.

[21]. Quinlan J R, ”Induction of decision tree,”Machine Learning,1986, (1),pp.81-106

[22]. Quinlan J R, ”C4.5 program for machine learning,” San Marteo Morgan Kaufmann Publisher -s,1993,pp.21-301

[23]. Quinlan J R, ”Simplifying Decision Tree,” Internet Journal of Man-Machine Studies,1987,27,pp.221-234

[24]. Yang Xue-bing,Zhang Jun, ”Decision Tree Algorithm and its core technology,” Computer Technology and Development,2007, 17(1),pp.43-45.

[25]. Qu Kai-she,Wen Cheng-li,Wang Jun-hong, ”An improved algorithm of ID3 algorithm,” Computer Engineering and Applications, 2003,(25),pp.104-107.

[26]. Huang Ai-hui, ”Improvement and application of decision tree C4.5 algorithm ,“ Science Technology and Engineering,2009, (1),pp.34-37.

[27]. B. Wang and H. Zhang, “Improving the Ranking Performance of Decision Trees”, ECML, pp. 461-472, 2006.

[28]. S. Sheng and C. X. Ling, “Hybrid Cost-Sensitive Decision Tree”, PKDD, pp. 274-284, 2005.

[29]. C. X. Ling, Q. Yang, J. Wang, and ShiChao Zhang, “Decision Trees with Minimal Costs”, ICML, 2004.

[30]. “Predicting student’s behaviour in education using J48 algorithm analysis tools in WEKA Environment Authors”: Daya Gupta, Rajni Jindal, Vaibhav Verma, Dilpreet Singh Kohli, Shashi Kant Sharma,(2010).

`

APPENDIX A

Decision Trees Generated on Application of Algorithms on the CASE STUDY- Predicting and Analyzing Student's behavior in Education using Decision Tree Algorithm (Counseling Help)

Section 1. Decision Tree Generated on Running C4.5 Algorithm looks like

```
| | | attribute scope:1
| | | | | | (8)
| | | attribute scope:2
| | | | | | (8)
| | | attribute scope:3
| | | | | | attribute perspective:f
| | | | | | | | | attribute physics<=84
| | | | | | | | | | | | (3)
| | | | | | | | | attribute physics>84
| | | | | | | | | | | | (6)
| | | | | | attribute perspective:p
| | | | | | | | | (5)
| | | attribute scope:4
```

`

```
| | | | | | attribute maths<=74
| | | | | | | | | attribute chem<=62
| | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | (1)
| | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | (4)
| | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | attribute physics<=62
| | | | | | | | | | | | | | | | | | | | | (2)
| | | | | | | | | | | | | | | | | | attribute physics>62
| | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | attribute chem>62
| | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | attribute worker:e
| | | | | | | | | | | | | | | | | | (10)
```

`

```
| | | | | | | | | | | | | | | attribute worker:h
| | | | | | | | | | | | | | | | | | (7)
| | | | | | attribute maths>74
| | | | | | | | | attribute type_work:h
| | | | | | | | | | | | attribute physics<=81
| | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | attribute physics<=77
| | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | attribute physics>77
| | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | attribute physics>81
| | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | (3)
| | | | | | | | | attribute type_work:p
| | | | | | | | | | | | attribute placement:1
| | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | attribute placement:2
| | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | attribute placement:3
| | | | | | | | | | | | | | | attribute perspective:f
```

`

```
| | | | | | | | | | | | | | | | | | attribute fav:p
| | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | attribute fav:c
| | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | attribute fav:m
| | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | attribute fav:b
| | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | attribute fav:cs
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | attribute placement:4
| | | | | | | | | | | | | | | attribute creative:1
| | | | | | | | | | | | | | | | | | attribute fav:p
| | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | attribute fav:c
| | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | attribute fav:m
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute fav:b
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute fav:cs
| | | | | | | | | | | | | | | | | | | | | (6)
```

`

```
| | | | | | | | | | | | | | | attribute creative:2
| | | | | | | | | | | | | | | | | | attribute maths<=91
| | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | | | | (4)
| | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | attribute maths>91
| | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | attribute creative:3
| | | | | | | | | | | | | | | | | | attribute fath_prof:b
| | | | | | | | | | | | | | | | | | | | | attribute familyback:h
| | | | | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | | | | attribute familyback:p
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familyback:g
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familyback:n
| | | | | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | attribute fath_prof:t
| | | | | | | | | | | | | | | | | | | | | attribute physics<=84
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute chem<=85
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute chem>85
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (5)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute physics>84
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute chem<=88
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute chem>88
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | | | | (5)
| | | | | | | | | | | | | | | | | | attribute fath_prof:n
| | | | | | | | | | | | | | | | | | | | | attribute branch:it
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | | | | (2)
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | attribute branch:ec
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute branch:cs
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute branch:ee
| | | | | | | | | | | | | | | | | | | | | | | | attribute physics<=85
| | | | | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | | | | attribute physics>85
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (6)
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute branch:bt
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute branch:ps
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | attribute branch:ce
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute branch:me
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute creative:4
| | | | | | | | | | | | | | | | | | attribute fav:p
| | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | attribute physics<=82
| | | | | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | | | | | | | attribute physics>82
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute physics<=91
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (10)
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute physics>91
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute work_pressure:1
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute work_pressure:2
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute work_pressure:3
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (4)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute work_pressure:4
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute work_pressure:5
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute fav:c
| | | | | | | | | | | | | | | | | | | | | (7)
```

`

```
| | | | | | | | | | | | | | | | | | attribute fav:m
| | | | | | | | | | | | | | | | | | | | | attribute familyback:h
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familyback:p
| | | | | | | | | | | | | | | | | | | | | | | | attribute fath_prof:b
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute fath_prof:t
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute fath_prof:n
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familyback:g
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:f
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familyback:n
| | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | attribute fav:b
| | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | attribute fav:cs
| | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute creative:5
| | | | | | | | | | | | | | | | | | attribute chem<=82
| | | | | | | | | | | | | | | | | | | | | attribute worker:e
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | attribute maths<=85
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familyback:h
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familyback:p
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familyback:g
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familyback:n
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute maths>85
| | | | | | | | | | | | | | | | | | | | | | | | | | | (2)
| | | | | | | | | | | | | | | | | | | | | attribute worker:h
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | attribute chem>82
| | | | | | | | | | | | | | | | | | | | | attribute work_pressure:1
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute work_pressure:2
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute work_pressure:3
| | | | | | | | | | | | | | | | | | | | | | | | attribute chem<=91
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute chem>91
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute work_pressure:4
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (5)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute physics<=86
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute physics>86
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute work_pressure:5
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute placement:5
| | | | | | | | | | | | | | | (8)
| | | | | | | | | attribute type_work:t
| | | | | | | | | | | | (8)
| | | attribute scope:5
| | | | | | (8)
Algorithm Takes 206406 milli seconds
```

Section 2. Decision Tree Generated on Running Algorithm 1 looks like

```
| | attribute fav:p
| | | | | | attribute type_work:h
| | | | | | | | | (9)
| | | | | | attribute type_work:p
| | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | attribute chem<=79
| | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | attribute chem>79
| | | | | | | | | | | | | | | attribute physics<=87
```

`

```
| | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | attribute physics>87
| | | | | | | | | | | | | | | | | | attribute physics<=94
| | | | | | | | | | | | | | | | | | | | | (4)
| | | | | | | | | | | | | | | | | | attribute physics>94
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | attribute familyback:h
| | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | attribute familyback:p
| | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | (4)
| | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute familyback:g
| | | | | | | | | | | | | | | attribute physics<=77
| | | | | | | | | | | | | | | | | | (5)
| | | | | | | | | | | | | | | attribute physics>77
| | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | attribute familyback:n
| | | | | | | | | | | | | | | (10)
| | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | attribute chem<=86
| | | | | | | | | | | | | | | (8)
```

`

```
| | | | | | | | | | | | attribute chem>86
| | | | | | | | | | | | | | | attribute chem<=89
| | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | attribute chem>89
| | | | | | | | | | | | | | | | | | attribute physics<=90
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute physics>90
| | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | (4)
| | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | attribute fath_prof:b
| | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | attribute fath_prof:t
| | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | attribute fath_prof:n
| | | | | | | | | | | | | | | attribute chem<=86
| | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute chem>86
| | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | (10)
```

`

```
| | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | attribute type_work:t
| | | | | | | | | (8)
| | | attribute fav:c
| | | | | | attribute physics<=86
| | | | | | | | | (7)
| | | | | | attribute physics>86
| | | | | | | | | attribute perspective:f
| | | | | | | | | | | | (8)
| | | | | | | | | attribute perspective:p
| | | | | | | | | | | | attribute physics<=91
| | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | attribute physics>91
| | | | | | | | | | | | | | | (7)
| | | attribute fav:m
| | | | | | attribute syalbus:u
| | | | | | | | | (8)
| | | | | | attribute syalbus:n
| | | | | | | | | attribute creative:1
| | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | attribute physics<=77
```

`

```
| | | | | | | | | | | | | | | | | | (2)
| | | | | | | | | | | | | | | attribute physics>77
| | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | attribute creative:2
| | | | | | | | | | | | (4)
| | | | | | | | | attribute creative:3
| | | | | | | | | | | | attribute fath_prof:b
| | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | attribute physics<=81
| | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | attribute physics>81
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute fath_prof:t
| | | | | | | | | | | | | | | attribute familyback:h
| | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | attribute familyback:p
| | | | | | | | | | | | | | | | | | attribute physics<=89
| | | | | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | | | | attribute physics>89
| | | | | | | | | | | | | | | | | | | | | attribute physics<=93
| | | | | | | | | | | | | | | | | | | | | | | | (5)
| | | | | | | | | | | | | | | | | | | | | attribute physics>93
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | attribute chem<=88
| | | | | | | | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | | | | | | | attribute chem>88
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute familyback:g
| | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | attribute familyback:n
| | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute fath_prof:n
| | | | | | | | | | | | | | | attribute chem<=75
| | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | attribute chem>75
| | | | | | | | | | | | | | | | | | attribute maths<=91
| | | | | | | | | | | | | | | | | | | | | (3)
| | | | | | | | | | | | | | | | | | attribute maths>91
| | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | attribute physics<=82
| | | | | | | | | | | | | | | | | | | | | | | | | | | (2)
| | | | | | | | | | | | | | | | | | | | | | | | attribute physics>82
| | | | | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | attribute creative:4
| | | | | | | | | | | | attribute physics<=77
```

`

```
| | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | attribute physics>77
| | | | | | | | | | | | | | | attribute familyback:h
| | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute familyback:p
| | | | | | | | | | | | | | | | | | attribute maths<=87
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute maths>87
| | | | | | | | | | | | | | | | | | | | | attribute maths<=91
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute maths>91
| | | | | | | | | | | | | | | | | | | | | | | | attribute physics<=84
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute physics>84
| | | | | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | attribute familyback:g
| | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | attribute chem<=78
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | attribute chem>78
| | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | attribute familyback:n
```

`

```
| | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | attribute creative:5
| | | | | | | | | | | | attribute fath_prof:b
| | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | attribute fath_prof:t
| | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | attribute fath_prof:n
| | | | | | | | | | | | | | | attribute maths<=87
| | | | | | | | | | | | | | | | | | (5)
| | | | | | | | | | | | | | | attribute maths>87
| | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | attribute physics<=81
| | | | | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | | | | attribute physics>81
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | attribute syalbus:o
| | | | | | | | | attribute familyback:h
| | | | | | | | | | | | (6)
| | | | | | | | | attribute familyback:p
| | | | | | | | | | | | attribute chem<=79
| | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute chem>79
```

`

```
| | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | attribute physics<=82
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute physics>82
| | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | attribute familyback:g
| | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | (3)
| | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | (5)
| | | | | | | | | attribute familyback:n
| | | | | | | | | | | | (9)
| | | | | | attribute syalbus:a
```

`

```
| | | | | | | | | (8)
| | | attribute fav:b
| | | | | | attribute physics<=70
| | | | | | | | | (1)
| | | | | | attribute physics>70
| | | | | | | | | attribute chem<=81
| | | | | | | | | | | | (6)
| | | | | | | | | attribute chem>81
| | | | | | | | | | | | attribute physics<=83
| | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute physics>83
| | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | attribute maths<=90
| | | | | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | | | | attribute maths>90
| | | | | | | | | | | | | | | | | | | | | attribute physics<=87
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | attribute physics>87
| | | | | | | | | | | | | | | | | | | | | | | | (5)
| | | attribute fav:cs
| | | | | | attribute maths<=86
| | | | | | | | | (7)
```

`

```
| | | | | | attribute maths>86
| | | | | | | | | attribute physics<=82
| | | | | | | | | | | | (9)
| | | | | | | | | attribute physics>82
| | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | attribute maths<=94
| | | | | | | | | | | | | | | | | | (2)
| | | | | | | | | | | | | | | attribute maths>94
| | | | | | | | | | | | | | | | | | attribute physics<=95
| | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | attribute physics>95
| | | | | | | | | | | | | | | | | | | | | (8)
Algorithm Takes 66094 milli seconds
```

Section 3. Decision Tree Generated on Running Algorithm 2 looks like

`

```
| | | attribute chem<=74
| | | | | | attribute physics<=67
| | | | | | | | | attribute chem<=61
| | | | | | | | | | | | attribute physics<=57
| | | | | | | | | | | | | | | (1)
| | | | | | | | | | | | attribute physics>57
| | | | | | | | | | | | | | | attribute physics<=62
| | | | | | | | | | | | | | | | | | (2)
| | | | | | | | | | | | | | | attribute physics>62
| | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | attribute chem>61
| | | | | | | | | | | | attribute physics<=62
| | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | attribute physics>62
| | | | | | | | | | | | | | | (7)
| | | | | | attribute physics>67
| | | | | | | | | attribute physics<=76
| | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | (5)
| | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | (7)
```

`

```
| | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | (10)
| | | | | | | | | attribute physics>76
| | | | | | | | | | | | attribute physics<=81
| | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | attribute physics>81
| | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | (4)
| | | attribute chem>74
| | | | | | attribute physics<=83
| | | | | | | | | attribute maths<=86
| | | | | | | | | | | | attribute maths<=79
| | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | attribute familypressure:3
```

`

```
| | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | attribute maths>79
| | | | | | | | | | | | | | | attribute chem<=83
| | | | | | | | | | | | | | | | | | attribute chem<=79
| | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | attribute chem>79
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | attribute chem<=81
| | | | | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | | | | | | | attribute chem>81
| | | | | | | | | | | | | | | | | | | | | | | | | | | (5)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | (7)
```

`

```
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | attribute chem>83
| | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | attribute maths>86
| | | | | | | | | | | | attribute maths<=92
| | | | | | | | | | | | | | | attribute physics<=76
| | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | attribute physics>76
| | | | | | | | | | | | | | | | | | attribute chem<=85
| | | | | | | | | | | | | | | | | | | | | attribute physics<=81
| | | | | | | | | | | | | | | | | | | | | | | | (4)
| | | | | | | | | | | | | | | | | | | | | attribute physics>81
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute chem>85
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | (8)
```

`

```
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | attribute maths>92
| | | | | | | | | | | | | | | attribute maths<=95
| | | | | | | | | | | | | | | | | | attribute physics<=79
| | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | attribute physics>79
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | (2)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute maths>95
| | | | | | | | | | | | | | | | | | attribute maths<=97
```

`

```
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute maths>97
| | | | | | | | | | | | | | | | | | | | | (3)
| | | | | | attribute physics>83
| | | | | | | | | attribute chem<=88
| | | | | | | | | | | | attribute maths<=91
| | | | | | | | | | | | | | | attribute maths<=86
| | | | | | | | | | | | | | | | | | attribute branch:it
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute branch:ec
| | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | | | | attribute branch:cs
```

`

```
| | | | | | | | | | | | | | | | | | | | | (5)
| | | | | | | | | | | | | | | | | | attribute branch:ee
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute branch:bt
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute branch:ps
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute branch:ce
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute branch:me
| | | | | | | | | | | | | | | | | | | | | (3)
| | | | | | | | | | | | | | | attribute maths>86
| | | | | | | | | | | | | | | | | | attribute chem<=83
| | | | | | | | | | | | | | | | | | | | | attribute familyback:h
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | attribute familyback:p
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familyback:g
| | | | | | | | | | | | | | | | | | | | | | | | (2)
| | | | | | | | | | | | | | | | | | | | | attribute familyback:n
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute chem>83
| | | | | | | | | | | | | | | | | | | | | attribute maths<=88
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:f
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | | | | attribute maths>88
| | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | attribute maths>91
| | | | | | | | | | | | | | | attribute physics<=91
| | | | | | | | | | | | | | | | | | attribute chem<=83
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | (10)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute chem>83
| | | | | | | | | | | | | | | | | | | | | attribute physics<=87
| | | | | | | | | | | | | | | | | | | | | | | | attribute physics<=85
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute physics<=84
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute physics>84
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | | | | attribute physics>85
| | | | | | | | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | | | | attribute physics>87
| | | | | | | | | | | | | | | | | | | | | | | | (5)
| | | | | | | | | | | | | | | attribute physics>91
| | | | | | | | | | | | | | | | | | attribute chem<=84
| | | | | | | | | | | | | | | | | | | | | attribute chem<=81
| | | | | | | | | | | | | | | | | | | | | | | | (5)
| | | | | | | | | | | | | | | | | | | | | attribute chem>81
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | attribute chem>84
```

`

```
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | (4)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | attribute chem>88
| | | | | | | | | | | | attribute chem<=94
| | | | | | | | | | | | | | | attribute maths<=93
| | | | | | | | | | | | | | | | | | attribute chem<=91
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute chem>91
| | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | attribute maths>93
| | | | | | | | | | | | | | | | | | attribute branch:it
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute branch:ec
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute branch:cs
| | | | | | | | | | | | | | | | | | | | | attribute perspective:f
```

```
`

| | | | | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute branch:ee
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | | | | (9)
| | | | | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute branch:bt
| | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | attribute branch:ps
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute branch:ce
| | | | | | | | | | | | | | | | | | | | | (6)
| | | | | | | | | | | | | | | | | | attribute branch:me
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | attribute chem>94
| | | | | | | | | | | | | | | attribute familyback:h
```

`

```
| | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | attribute familyback:p
| | | | | | | | | | | | | | | | | | attribute familypressure:1
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute familypressure:2
| | | | | | | | | | | | | | | | | | | | | (4)
| | | | | | | | | | | | | | | | | | attribute familypressure:3
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute familypressure:4
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | | | | attribute familypressure:5
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute familyback:g
| | | | | | | | | | | | | | | | | | attribute perspective:f
| | | | | | | | | | | | | | | | | | | | | (7)
| | | | | | | | | | | | | | | | | | attribute perspective:p
| | | | | | | | | | | | | | | | | | | | | (8)
| | | | | | | | | | | | | | | attribute familyback:n
| | | | | | | | | | | | | | | | | | (9)

Algorithm Takes 362282 milli seconds
```

`

APPENDIX B

Decision Trees Generated on Application of Algorithms on the CASE STUDY- Predicting the level of students and monitoring their performance using Decision Tree Algorithm (Grading System)

Section 1. Decision Tree Generated on Running C4.5 Algorithm looks like

```
| | | attribute midterm_marks<=20
| | | | | | attribute participation:1
| | | | | | | | | attribute tutorials:1
| | | | | | | | | | | | attribute midterm_marks<=16
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute midterm_marks>16
| | | | | | | | | | | | | | | (c)
| | | | | | | | | attribute tutorials:2
| | | | | | | | | | | | (d)
| | | | | | | | | attribute tutorials:3
| | | | | | | | | | | | (c)
| | | | | | | | | attribute tutorials:4
| | | | | | | | | | | | (d)
```

`

```
| | | | | | | | | attribute tutorials:5
| | | | | | | | | | | | (d)
| | | | | | attribute participation:2
| | | | | | | | | attribute scope_placement:1
| | | | | | | | | | | | (d)
| | | | | | | | | attribute scope_placement:2
| | | | | | | | | | | | (d)
| | | | | | | | | attribute scope_placement:3
| | | | | | | | | | | | (d)
| | | | | | | | | attribute scope_placement:4
| | | | | | | | | | | | attribute interaction:1
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute interaction:2
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute interaction:3
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute interaction:4
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute interaction:5
| | | | | | | | | | | | | | | (d)
| | | | | | | | | attribute scope_placement:5
| | | | | | | | | | | | attribute interaction:1
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute interaction:2
```

`

```
| | | | | | | | | | | | | | | attribute labs:1
| | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | attribute labs:2
| | | | | | | | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | attribute labs:3
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=19
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>19
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute labs:4
| | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | attribute labs:5
| | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute interaction:3
| | | | | | | | | | | | | | | attribute bname:it
```

`

```
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=16
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>16
| | | | | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute interaction:4
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute interaction:5
| | | | | | | | | | | | | | | (d)
| | | | | | attribute participation:3
| | | | | | | | | (b)
| | | | | | attribute participation:4
| | | | | | | | | attribute midterm_marks<=18
| | | | | | | | | | | | (b)
| | | | | | | | | attribute midterm_marks>18
| | | | | | | | | | | | (a)
| | | | | | attribute participation:5
| | | | | | | | | (c)
| | | attribute midterm_marks>20
| | | | | | attribute participation:1
| | | | | | | | | attribute midterm_marks<=25
| | | | | | | | | | | | attribute creativity_and_effectiveness:1
| | | | | | | | | | | | | | | (c)
```

`

```
| | | | | | | | | | | | attribute creativity_and_effectiveness:2
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute creativity_and_effectiveness:3
| | | | | | | | | | | | | | | attribute understanding_theory:1
| | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | attribute understanding_theory:2
| | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | attribute understanding_theory:3
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=22
| | | | | | | | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | | | | | | | attribute midterm_marks<=21
| | | | | | | | | | | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | | | | | | | | | | attribute midterm_marks>21
| | | | | | | | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>22
| | | | | | | | | | | | | | | | | | | | | attribute labs:1
| | | | | | | | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | | | | | | | attribute labs:2
| | | | | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | | | | attribute labs:3
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | attribute labs:4
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | | | | attribute labs:5
| | | | | | | | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | attribute understanding_theory:4
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute understanding_theory:5
| | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute creativity_and_effectiveness:4
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute creativity_and_effectiveness:5
| | | | | | | | | | | | | | | (d)
| | | | | | | | | attribute midterm_marks>25
| | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=29
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>29
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | (d)
```

```
`

| | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | (d)
| | | | | | attribute participation:2
| | | | | | | | | attribute understanding_theory:1
| | | | | | | | | | | | (d)
| | | | | | | | | attribute understanding_theory:2
| | | | | | | | | | | | (c)
| | | | | | | | | attribute understanding_theory:3
| | | | | | | | | | | | attribute labs:1
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute labs:2
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute labs:3
| | | | | | | | | | | | | | | attribute interaction:1
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute interaction:2
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute interaction:3
| | | | | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | | | | (c)
```

`

```
| | | | | | | | | | | | | | | attribute interaction:4
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute interaction:5
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute labs:4
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute labs:5
| | | | | | | | | | | | | | | (c)
| | | | | | | | | attribute understanding_theory:4
| | | | | | | | | | | | attribute work_type:t
| | | | | | | | | | | | | | | attribute interaction:1
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute interaction:2
| | | | | | | | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute interaction:3
```

`

```
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute interaction:4
| | | | | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | | | | | | | | | | attribute midterm_marks<=27
| | | | | | | | | | | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | | | | | | | | | | attribute midterm_marks>27
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute interaction:5
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute work_type:p
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute work_type:m
| | | | | | | | | | | | | | | (c)
| | | | | | | | | attribute understanding_theory:5
| | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | (c)
| | | | | | attribute participation:3
| | | | | | | | | attribute scope_placement:1
| | | | | | | | | | | | (c)
| | | | | | | | | attribute scope_placement:2
| | | | | | | | | | | | (c)
| | | | | | | | | attribute scope_placement:3
| | | | | | | | | | | | (c)
| | | | | | | | | attribute scope_placement:4
| | | | | | | | | | | | (a)
| | | | | | | | | attribute scope_placement:5
```

`

```
| | | | | | | | | | | | attribute interaction:1
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute interaction:2
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute interaction:3
| | | | | | | | | | | | | | | attribute work_type:t
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute work_type:p
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | attribute work_type:m
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute interaction:4
| | | | | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | | | | attribute labs:1
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | attribute labs:2
| | | | | | | | | | | | | | | | | | | | | | | | (b)
```

```
`

| | | | | | | | | | | | | | | | | | | | | attribute labs:3
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | attribute labs:4
| | | | | | | | | | | | | | | | | | | | | | | | attribute midterm_marks<=23
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute midterm_marks<=22
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute midterm_marks>22
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | | | | | | | | | | attribute midterm_marks>23
| | | | | | | | | | | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | | | | | | | attribute labs:5
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | attribute interaction:5
| | | | | | | | | | | | | | | (b)
| | | | | | attribute participation:4
| | | | | | | | | attribute bname:it
| | | | | | | | | | | | (b)
| | | | | | | | | attribute bname:cs
| | | | | | | | | | | | attribute lib_visit:1
| | | | | | | | | | | | | | | (b)
```

`

```
| | | | | | | | | | | | attribute lib_visit:2
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute lib_visit:3
| | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | attribute lib_visit:4
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute lib_visit:5
| | | | | | | | | | | | | | | (a)
| | | | | | attribute participation:5
| | | | | | | | | (a)
Algorithm Takes 66172 milli seconds
```

Section 2. Decision Tree Generated on Running Algorithm 1 looks like

```
| | | attribute lectures:1
| | | | | | attribute bname:it
| | | | | | | | | (d)
| | | | | | attribute bname:cs
| | | | | | | | | attribute midterm_marks<=18
| | | | | | | | | | | | (d)
| | | | | | | | | attribute midterm_marks>18
| | | | | | | | | | | | (c)
| | | attribute lectures:2
| | | | | | attribute bname:it
| | | | | | | | | (d)
```

`

```
| | | | | | attribute bname:cs
| | | | | | | | | attribute midterm_marks<=19
| | | | | | | | | | | | (d)
| | | | | | | | | attribute midterm_marks>19
| | | | | | | | | | | | attribute lib_visit:1
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute lib_visit:2
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute lib_visit:3
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute lib_visit:4
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute lib_visit:5
| | | | | | | | | | | | | | | (a)
| | | attribute lectures:3
| | | | | | attribute tutorials:1
| | | | | | | | | attribute bname:it
| | | | | | | | | | | | (c)
| | | | | | | | | attribute bname:cs
| | | | | | | | | | | | attribute midterm_marks<=19
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute midterm_marks>19
| | | | | | | | | | | | | | | (c)
| | | | | | attribute tutorials:2
| | | | | | | | | attribute sname:d
```

`

```
| | | | | | | | | | | | attribute midterm_marks<=22
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute midterm_marks>22
| | | | | | | | | | | | | | | (b)
| | | | | | | | | attribute sname:s
| | | | | | | | | | | | attribute midterm_marks<=19
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute midterm_marks>19
| | | | | | | | | | | | | | | (c)
| | | | | | | | | attribute sname:ds
| | | | | | | | | | | | attribute labs:1
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute labs:2
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute labs:3
| | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=19
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>19
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute labs:4
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute labs:5
```

`

```
| | | | | | | | | | | | | | | (d)
| | | | | | | | | attribute sname:os
| | | | | | | | | | | | attribute midterm_marks<=20
| | | | | | | | | | | | | | | attribute midterm_marks<=17
| | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | attribute midterm_marks>17
| | | | | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute midterm_marks>20
| | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=23
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>23
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | attribute sname:cn
| | | | | | | | | | | | attribute midterm_marks<=20
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute midterm_marks>20
| | | | | | | | | | | | | | | (b)
| | | | | | attribute tutorials:3
| | | | | | | | | attribute midterm_marks<=18
```

`

```
| | | | | | | | | | | | (b)
| | | | | | | | | attribute midterm_marks>18
| | | | | | | | | | | | attribute midterm_marks<=21
| | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=20
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>20
| | | | | | | | | | | | | | | | | | | | | attribute work_type:t
| | | | | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | | | | attribute work_type:p
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | attribute work_type:m
| | | | | | | | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute midterm_marks>21
| | | | | | | | | | | | | | | (b)
| | | | | | attribute tutorials:4
| | | | | | | | | attribute midterm_marks<=18
| | | | | | | | | | | | (b)
| | | | | | | | | attribute midterm_marks>18
| | | | | | | | | | | | attribute midterm_marks<=23
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute midterm_marks>23
| | | | | | | | | | | | | | | (a)
```

`

```
| | | | | | attribute tutorials:5
| | | | | | | | | (b)
| | | attribute lectures:4
| | | | | | attribute understanding_practical:1
| | | | | | | | | (c)
| | | | | | attribute understanding_practical:2
| | | | | | | | | attribute work_type:t
| | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | attribute midterm_marks<=23
| | | | | | | | | | | | | | | | | | attribute interaction:1
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute interaction:2
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute interaction:3
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute interaction:4
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute interaction:5
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute midterm_marks>23
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | attribute lib_visit:1
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute lib_visit:2
```

`

```
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute lib_visit:3
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=21
| | | | | | | | | | | | | | | | | | | | | attribute midterm_marks<=20
| | | | | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | | | | attribute midterm_marks>20
| | | | | | | | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>21
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute lib_visit:4
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=25
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>25
| | | | | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | attribute lib_visit:5
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | attribute work_type:p
| | | | | | | | | | | | (b)
| | | | | | | | | attribute work_type:m
| | | | | | | | | | | | attribute midterm_marks<=23
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute midterm_marks>23
| | | | | | | | | | | | | | | attribute midterm_marks<=25
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute midterm_marks>25
```

`

```
| | | | | | | | | | | | | | | | | | (b)
| | | | | | attribute understanding_practical:3
| | | | | | | | | attribute midterm_marks<=23
| | | | | | | | | | | | attribute midterm_marks<=19
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute midterm_marks>19
| | | | | | | | | | | | | | | attribute work_type:t
| | | | | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | | | | | | | | | | attribute participation:1
| | | | | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | | | | attribute participation:2
| | | | | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | | | | | | | attribute participation:3
| | | | | | | | | | | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | | | | | | | | | | attribute participation:4
| | | | | | | | | | | | | | | | | | | | | | | | | | | (b)
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | attribute participation:5
| | | | | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute work_type:p
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute work_type:m
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | attribute midterm_marks>23
| | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | attribute midterm_marks<=25
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute midterm_marks>25
| | | | | | | | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | | | | | | | (a)
```

`

```
| | | | | | attribute understanding_practical:4
| | | | | | | | | attribute sname:d
| | | | | | | | | | | | attribute midterm_marks<=18
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute midterm_marks>18
| | | | | | | | | | | | | | | (a)
| | | | | | | | | attribute sname:s
| | | | | | | | | | | | (b)
| | | | | | | | | attribute sname:ds
| | | | | | | | | | | | attribute midterm_marks<=18
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute midterm_marks>18
| | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=22
| | | | | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>22
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | attribute sname:os
| | | | | | | | | | | | (b)
| | | | | | | | | attribute sname:cn
| | | | | | | | | | | | attribute midterm_marks<=24
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute midterm_marks>24
```

`

```
| | | | | | | | | | | | | | | (a)
| | | | | | attribute understanding_practical:5
| | | | | | | | | attribute midterm_marks<=21
| | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | attribute midterm_marks<=17
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute midterm_marks>17
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | (b)
| | | | | | | | | attribute midterm_marks>21
| | | | | | | | | | | | (a)
| | | attribute lectures:5
| | | | | | attribute sname:d
| | | | | | | | | attribute bname:it
| | | | | | | | | | | | (c)
| | | | | | | | | attribute bname:cs
| | | | | | | | | | | | (b)
| | | | | | attribute sname:s
| | | | | | | | | (b)
| | | | | | attribute sname:ds
| | | | | | | | | attribute bname:it
| | | | | | | | | | | | (b)
| | | | | | | | | attribute bname:cs
| | | | | | | | | | | | (a)
```

`

```
| | | | | | attribute sname:os
| | | | | | | | | attribute bname:it
| | | | | | | | | | | | (a)
| | | | | | | | | attribute bname:cs
| | | | | | | | | | | | (c)
| | | | | | attribute sname:cn
| | | | | | | | | (b)
Algorithm Takes 43203 milli seconds
```

Section 3. Decision Tree Generated on Running Algorithm 2 looks like

```
| | | attribute midterm_marks<=20
| | | | | | attribute midterm_marks<=15
| | | | | | | | | attribute sname:d
| | | | | | | | | | | | (c)
| | | | | | | | | attribute sname:s
| | | | | | | | | | | | (d)
| | | | | | | | | attribute sname:ds
| | | | | | | | | | | | attribute interaction:1
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute interaction:2
| | | | | | | | | | | | | | | (d)
```

`

```
| | | | | | | | | | | | attribute interaction:3
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute interaction:4
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute interaction:5
| | | | | | | | | | | | | | | (d)
| | | | | | | | | attribute sname:os
| | | | | | | | | | | | (d)
| | | | | | | | | attribute sname:cn
| | | | | | | | | | | | (d)
| | | | | | attribute midterm_marks>15
| | | | | | | | | attribute understanding_practical:1
| | | | | | | | | | | | (d)
| | | | | | | | | attribute understanding_practical:2
| | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute bname:cs
```

`

```
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | (d)
| | | | | | | | | attribute understanding_practical:3
| | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | (b)
| | | | | | | | | attribute understanding_practical:4
| | | | | | | | | | | | attribute midterm_marks<=18
| | | | | | | | | | | | | | | attribute midterm_marks<=16
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute midterm_marks>16
| | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute midterm_marks>18
| | | | | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | attribute sname:s
```

`

```
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | attribute understanding_practical:5
| | | | | | | | | | | | attribute midterm_marks<=17
| | | | | | | | | | | | | | | attribute midterm_marks<=16
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute midterm_marks>16
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute midterm_marks>17
| | | | | | | | | | | | | | | (b)
| | | attribute midterm_marks>20
| | | | | | attribute midterm_marks<=25
| | | | | | | | | attribute understanding_theory:1
| | | | | | | | | | | | (c)
| | | | | | | | | attribute understanding_theory:2
| | | | | | | | | | | | (c)
| | | | | | | | | attribute understanding_theory:3
| | | | | | | | | | | | attribute understanding_practical:1
| | | | | | | | | | | | | | | (c)
```

```
`

| | | | | | | | | | | | attribute understanding_practical:2
| | | | | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=22
| | | | | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>22
| | | | | | | | | | | | | | | | | | | | | attribute midterm_marks<=23
| | | | | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | | | | attribute midterm_marks>23
| | | | | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | | | | (d)
| | | | | | | | | | | | attribute understanding_practical:3
| | | | | | | | | | | | | | | attribute midterm_marks<=23
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=22
| | | | | | | | | | | | | | | | | | | | | attribute midterm_marks<=21
| | | | | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | | | | attribute midterm_marks>21
| | | | | | | | | | | | | | | | | | | | | | | | (b)
```

`

```
| | | | | | | | | | | | | | | | | | attribute midterm_marks>22
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute midterm_marks>23
| | | | | | | | | | | | | | | | | | attribute lib_visit:1
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute lib_visit:2
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute lib_visit:3
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute lib_visit:4
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | attribute lib_visit:5
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute understanding_practical:4
| | | | | | | | | | | | | | | attribute midterm_marks<=23
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=22
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>22
| | | | | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | attribute midterm_marks>23
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | attribute understanding_practical:5
| | | | | | | | | | | | | | | (c)
| | | | | | | | | attribute understanding_theory:4
```

`

```
| | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | attribute tutorials:1
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute tutorials:2
| | | | | | | | | | | | | | | | | | attribute midterm_marks<=23
| | | | | | | | | | | | | | | | | | | | | attribute midterm_marks<=21
| | | | | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | | | | | | | attribute midterm_marks>21
| | | | | | | | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | | | | attribute midterm_marks>23
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute tutorials:3
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute tutorials:4
| | | | | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute tutorials:5
```

`

```
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | attribute bname:it
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | attribute bname:cs
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | attribute interaction:1
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute interaction:2
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute interaction:3
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute interaction:4
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | attribute interaction:5
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | attribute understanding_theory:5
| | | | | | | | | | | | (a)
| | | | | | attribute midterm_marks>25
| | | | | | | | | attribute interaction:1
| | | | | | | | | | | | attribute sname:d
```

`

```
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | (b)
| | | | | | | | | attribute interaction:2
| | | | | | | | | | | | (b)
| | | | | | | | | attribute interaction:3
| | | | | | | | | | | | (b)
| | | | | | | | | attribute interaction:4
| | | | | | | | | | | | attribute midterm_marks<=27
| | | | | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | attribute sname:cn
```

`

```
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | attribute midterm_marks>27
| | | | | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | | | | (b)
| | | | | | | | | attribute interaction:5
| | | | | | | | | | | | attribute sname:d
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute sname:s
| | | | | | | | | | | | | | | (b)
| | | | | | | | | | | | attribute sname:ds
| | | | | | | | | | | | | | | (a)
| | | | | | | | | | | | attribute sname:os
| | | | | | | | | | | | | | | (c)
| | | | | | | | | | | | attribute sname:cn
| | | | | | | | | | | | | | | (b)
Algorithm Takes 115297 milli seconds
```

`

APPENDIX C

Decision Trees Generated on Application of Algorithms on the CASE STUDY- Predicting the eligibility of students regarding AIEEE Counseling (Based on Real AIEEE data)

Section 1. Decision Tree Generated on Running C4.5 Algorithm looks like

```
| | | attribute Marks<=93
| | | | | | attribute catagory:GEN
| | | | | | | | | attribute Gender:M
| | | | | | | | | | | | attribute AI_Rank<=310105
| | | | | | | | | | | | | | | attribute AI_Rank<=165516
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute AI_Rank>165516
| | | | | | | | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | | | | | | | attribute Birth_year<=89
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | attribute Birth_year>89
| | | | | | | | | | | | | | | | | | | | | | | | attribute Marks<=47
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute Marks>47
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks<=54
```

```
`

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks>54
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks<=57
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks>57
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=170607
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(category))
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>170607
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute course:BARCH
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute AI_Rank>310105
| | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | attribute Gender:F
| | | | | | | | | | | | (not eligible)
| | | | | | attribute catagory:OBC
| | | | | | | | | attribute AI_Rank<=309285
| | | | | | | | | | | | (not eligible)
| | | | | | | | | attribute AI_Rank>309285
| | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | attribute AI_Rank<=454325
| | | | | | | | | | | | | | | | | | attribute Marks<=21
| | | | | | | | | | | | | | | | | | | | | attribute Birth_year<=88
| | | | | | | | | | | | | | | | | | | | | | | | attribute Marks<=15
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute Marks>15
| | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(category))
| | | | | | | | | | | | | | | | | | | | | attribute Birth_year>88
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute Marks>21
| | | | | | | | | | | | | | | | | | | | | attribute Marks<=26
| | | | | | | | | | | | | | | | | | | | | | | | attribute Marks<=23
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=373466
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year<=88
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year>88
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>373466
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute Marks>23
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | attribute Marks>26
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute AI_Rank>454325
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute course:BARCH
| | | | | | | | | | | | | | | (not eligible)
| | | | | | attribute catagory:SC
```

`

```
| | | | | | | | | attribute Marks<=36
| | | | | | | | | | | | attribute Marks<=11
| | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute Marks>11
| | | | | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | | | | attribute Gender:M
| | | | | | | | | | | | | | | | | | | | | attribute Marks<=16
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | attribute Marks>16
| | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | attribute Gender:F
| | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | attribute course:BARCH
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | attribute Marks>36
| | | | | | | | | | | | attribute Marks<=65
| | | | | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | attribute course:BARCH
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute Marks>65
| | | | | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute course:BARCH
```

`

```
| | | | | | | | | | | | | | | | | | attribute Birth_year<=89
| | | | | | | | | | | | | | | | | | | | | attribute Marks<=73
| | | | | | | | | | | | | | | | | | | | | | | | attribute Marks<=69
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute Marks>69
| | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | attribute Marks>73
| | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | attribute Birth_year>89
| | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | attribute catagory:ST
| | | | | | | | | attribute ST_Rank<=4250
| | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | attribute Marks<=18
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute Marks>18
| | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | attribute course:BARCH
| | | | | | | | | | | | | | | attribute ST_Rank<=3120
| | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | attribute ST_Rank>3120
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | attribute ST_Rank>4250
| | | | | | | | | | | | (eligible(catagory))
```

`

```
| | | attribute Marks>93
| | | | | | attribute ST_Rank<=1181
| | | | | | | | | attribute AI_Rank<=43743
| | | | | | | | | | | | attribute AI_Rank<=12056
| | | | | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute course:BARCH
| | | | | | | | | | | | | | | | | | attribute AI_Rank<=5755
| | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | attribute AI_Rank>5755
| | | | | | | | | | | | | | | | | | | | | attribute Marks<=130
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | attribute Marks>130
| | | | | | | | | | | | | | | | | | | | | | | | attribute ST_Rank<=592
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year<=89
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year>89
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute ST_Rank>592
| | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | attribute AI_Rank>12056
| | | | | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | attribute course:BARCH
```

`

```
| | | | | | | | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | attribute AI_Rank>43743
| | | | | | | | | | | | (not eligible)
| | | | | | attribute ST_Rank>1181
| | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | (eligible)
| | | | | | | | | attribute course:BARCH
| | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | (eligible(catagory))
Algorithm Takes 30172 milli seconds
```

`

Section 2. Decision Tree Generated on Running Algorithm 1 looks like

```
| | | attribute Marks<=93
| | | | | | attribute catagory:GEN
| | | | | | | | | attribute Birth_year<=88
| | | | | | | | | | | | (not eligible)
| | | | | | | | | attribute Birth_year>88
| | | | | | | | | | | | attribute Birth_year<=89
| | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute Birth_year>89
| | | | | | | | | | | | | | | attribute Marks<=24
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute Marks>24
| | | | | | | | | | | | | | | | | | attribute AI_Rank<=188248
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=102850
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>102850
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=146791
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>146791
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=166710
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>166710
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks<=57
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks>57
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=170607
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(category))
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>170607
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute AI_Rank>188248
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | attribute catagory:OBC
| | | | | | | | | attribute AI_Rank<=309285
| | | | | | | | | | | | (not eligible)
| | | | | | | | | attribute AI_Rank>309285
| | | | | | | | | | | | attribute Marks<=-1
| | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute Marks>-1
| | | | | | | | | | | | | | | attribute Marks<=15
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute Marks>15
```

`

```
| | | | | | | | | | | | | | | | | | attribute AI_Rank<=366314
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute AI_Rank>366314
| | | | | | | | | | | | | | | | | | | | | attribute Gender:M
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=394701
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks<=20
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks>20
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks<=21
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks>21
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year<=88
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year>88
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=372248
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>372248
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>394701
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks<=17
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks>17
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year<=88
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(category))
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year>88
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | attribute Gender:F
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | attribute catagory:SC
| | | | | | | | | attribute Marks<=36
| | | | | | | | | | | | attribute Marks<=11
| | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute Marks>11
| | | | | | | | | | | | | | | attribute AI_Rank<=236692
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute AI_Rank>236692
| | | | | | | | | | | | | | | | | | attribute AI_Rank<=358052
| | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | attribute AI_Rank>358052
| | | | | | | | | | | | | | | | | | | | | attribute Marks<=16
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | attribute Marks>16
| | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | attribute Marks>36
| | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | attribute course:BARCH
| | | | | | | | | | | | | | | attribute Marks<=65
```

`

```
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute Marks>65
| | | | | | | | | | | | | | | | | | attribute Gender:M
| | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | attribute Gender:F
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=25817
| | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>25817
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=27627
| | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>27627
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year<=89
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year>89
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | attribute catagory:ST
| | | | | | | | | attribute ST_Rank<=4250
| | | | | | | | | | | | attribute Marks<=26
| | | | | | | | | | | | | | | attribute Marks<=0
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute Marks>0
| | | | | | | | | | | | | | | | | | attribute Marks<=14
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute Marks>14
```

```
`

| | | | | | | | | | | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | attribute course:BARCH
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute Marks>26
| | | | | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | attribute course:BARCH
| | | | | | | | | | | | | | | | | | attribute Birth_year<=89
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute Birth_year>89
| | | | | | | | | | | | | | | | | | | | | attribute Marks<=64
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | attribute Marks>64
| | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | attribute ST_Rank>4250
| | | | | | | | | | | | (eligible(catagory))
| | | attribute Marks>93
| | | | | | attribute ST_Rank<=1181
| | | | | | | | | attribute AI_Rank<=43743
| | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | attribute course:BARCH
| | | | | | | | | | | | | | | attribute AI_Rank<=9852
```

`

```
| | | | | | | | | | | | | | | | | | attribute AI_Rank<=4811
| | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | attribute AI_Rank>4811
| | | | | | | | | | | | | | | | | | | | | attribute Marks<=135
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | attribute Marks>135
| | | | | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | attribute AI_Rank>9852
| | | | | | | | | | | | | | | | | | attribute Gender:M
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute Gender:F
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=16330
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>16330
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=18807
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>18807
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year<=89
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year>89
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | attribute AI_Rank>43743
| | | | | | | | | | | | (not eligible)
| | | | | | attribute ST_Rank>1181
```

`

```
| | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | (not eligible)
| | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | (not eligible)
| | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | (eligible(catagory))
Algorithm Takes 27578 milli seconds
```

Section 3. Decision Tree Generated on Running Algorithm 2 looks like

```
| | | attribute AI_Rank<=299408
| | | | | | attribute AI_Rank<=148992
| | | | | | | | | attribute AI_Rank<=74203
| | | | | | | | | | | | attribute AI_Rank<=35190
| | | | | | | | | | | | | | | attribute AI_Rank<=17555
| | | | | | | | | | | | | | | | | | attribute AI_Rank<=8767
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=4339
| | | | | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>4339
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=6509
| | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>6509
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=7588
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks<=136
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks<=135
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks>135
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Marks>136
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>7588
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | attribute AI_Rank>8767
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=13150
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=10910
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:GEN
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>10910
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=12152
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute course:BTECH
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute course:BARCH
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>12152
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>13150
| | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:GEN
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | attribute AI_Rank>17555
| | | | | | | | | | | | | | | | | | attribute ST_Rank<=1721
| | | | | | | | | | | | | | | | | | | | | attribute Marks<=103
| | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | attribute Marks>103
| | | | | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | attribute ST_Rank>1721
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=26325
| | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>26325
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=30705
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=28500
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>28500
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>30705
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute AI_Rank>35190
| | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | attribute AI_Rank>74203
```

```
`

| | | | | | | | | | | | (not eligible)
| | | | | | attribute AI_Rank>148992
| | | | | | | | | attribute AI_Rank<=225688
| | | | | | | | | | | | attribute AI_Rank<=189127
| | | | | | | | | | | | | | | attribute AI_Rank<=169742
| | | | | | | | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | attribute AI_Rank>169742
| | | | | | | | | | | | | | | | | | attribute AI_Rank<=178053
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=172628
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=170607
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year<=89
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year>89
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(category))
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>170607
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>172628
```

`

```
| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute AI_Rank>178053
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute AI_Rank>189127
| | | | | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | attribute AI_Rank>225688
| | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | (eligible(catagory))
| | | attribute AI_Rank>299408
| | | | | | attribute AI_Rank<=449212
| | | | | | | | | attribute AI_Rank<=372575
```

`

```
| | | | | | | | | | | | attribute AI_Rank<=336063
| | | | | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | attribute AI_Rank>336063
| | | | | | | | | | | | | | | attribute AI_Rank<=355983
| | | | | | | | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | attribute AI_Rank>355983
| | | | | | | | | | | | | | | | | | attribute AI_Rank<=364564
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | attribute AI_Rank>364564
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=369463
```

```
`

| | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>369463
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank<=371483
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year<=88
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | | | | | | | | | | | | | attribute Birth_year>88
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | | | | | | | | | | attribute AI_Rank>371483
| | | | | | | | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | attribute AI_Rank>372575
| | | | | | | | | | | | attribute AI_Rank<=409312
| | | | | | | | | | | | | | | attribute catagory:GEN
| | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute catagory:OBC
| | | | | | | | | | | | | | | | | | attribute Birth_year<=88
| | | | | | | | | | | | | | | | | | | | | (eligible(category))
| | | | | | | | | | | | | | | | | | attribute Birth_year>88
| | | | | | | | | | | | | | | | | | | | | (not eligible)
| | | | | | | | | | | | | | | attribute catagory:SC
| | | | | | | | | | | | | | | | | | (eligible(catagory))
| | | | | | | | | | | | | | | attribute catagory:ST
| | | | | | | | | | | | | | | | | | (eligible)
| | | | | | | | | | | | attribute AI_Rank>409312
| | | | | | | | | | | | | | | (not eligible)
```

```
`

| | | | | | attribute AI_Rank>449212

| | | | | | | | | (not eligible)


Algorithm Takes 108484 milli seconds
```