

# DESIGN AND IMPLEMENTATION OF BUILDING DECISION TREE USING C4.5 ALGORITHM

KUSRINI

**Abstract.** Decision tree is one of data mining techniques that is applied in classification and prediction. Decision tree model consists of a set of rules for dividing a large heterogeneous population into smaller, more homogeneous groups with respect to a particular target variable. There are some algorithms to build decision tree, such as Classification and Regression Trees (CART) algorithm and C4.5 algorithm. In this research, we designed and implemented a tool to build decision tree using C4.5 algorithm. To simplify the understanding of our tool, we put sample data in our database. The sample database can help describe the process to make a decision to play tennis or not, by considering outlook, temperature, humidity and windy.

*Key words and Phrases:* Data mining, Decision tree, C4.5 Algorithm

## 1. INTRODUCTION

By using Online Transactional Processing System (OLTP), business organization can produce data in great quantities. These data can be processed to become useful knowledge the for decision maker in organizations.

One of the steps in knowledge discovery is data mining, instead of some other steps such as data cleaning, data integration, data selection and data transformation. Data mining techniques is a specific implementation of algorithm that is used in data mining operations. The five most common data mining techniques are association discovery, sequential pattern discovery, classification, clustering and forecasting [2].

One of techniques for classification is using decision tree. Decision trees represent a supervised approach to classification. A decision tree is a simple structure where non-terminal nodes represent tests on one or more attributes and terminal nodes reflect decision outcomes[4].

There are some algorithm to generate decision tree such as CART and C4.5[4]. The C4.5 algorithm is a Quinlan's extension of his own ID3 algorithm for generating decision trees. Just like CART, the C4.5 algorithm recursively visits each decision node, selecting the optimal split, until no further splits are possible. But unlike CART, the C4.5 produces a tree of more variable shape.

In this research we designed and implemented a tool to build decision tree using C4.5 algorithm. This tool can be used to build decision tree for every problem that has categorical and discrete value of variable. We have provided interfaces to input variables and their data. As a programming tool and database

management system tool we used Borland Delphi and Interbase.

The steps in generating a decision tree using C4.5 algorithm are[1]:

1. Choose attribute for root node
2. Create branch for each value of that attribute
3. Split cases according to branches
4. Repeat process for each branch until all cases in the branch have the same class

To choose root node we use formula 1 [1]:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (1)$$

with:

$\{S_1, \dots, S_i, \dots, S_n\}$  : partitions of S according to value of attribute A

n : number of attributes A

$|S_i|$  : number of cases in the partitions  $S_i$

$|S|$  : total number of cases in S

while Entropy( $S_i$ ) is calculate by formula 2 [1]:

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i \quad (1)$$

with:

S : case Set

n : number of cases in the partition S

$p_i$  : proportion of  $S_i$  to S

Kusrini and Hartati, S (2007) have implemented C4.5 to build decision tree that used to classify cancellation possibility of new student at STMIK AMIKOM Yogyakarta[3].

## 2. MAIN RESULTS

The data flow diagram for our application is shown in figure 1:

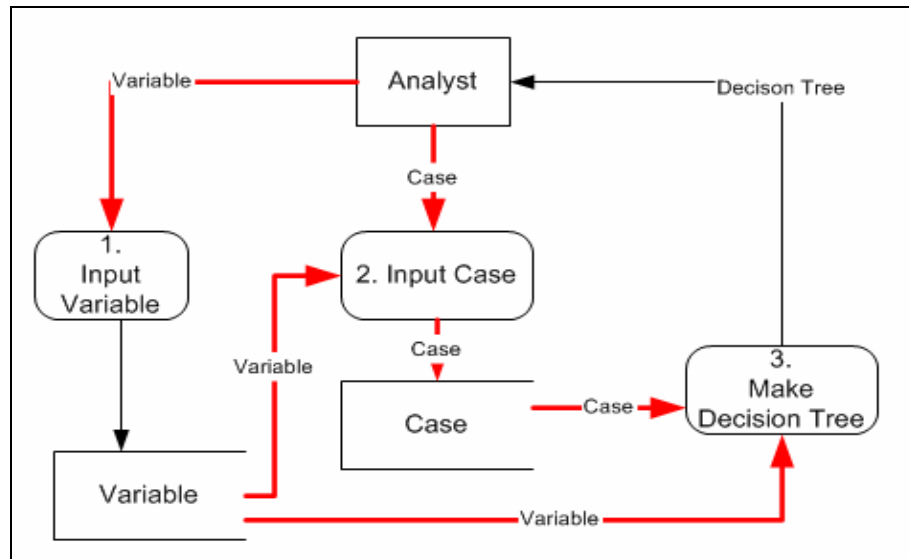


Figure 1. Data Flow Diagram

From the Figure 1, we see that processes in our system are:

1. Analyst input variable data and it will be stored in *variable* table.
2. Analyst input case data and it will be store in *case* table. The variables of cases in this process are depended on data variable in *variable* table.
3. System will make decision tree considering variables and cases data in *variable* table and *case* table and it will present to the analyst.

This application uses 2 kinds of tables; they are initial table and running table. Initial table is tables that create when the application is developed, while running table is tables that generate by system when application is running.

The initial tables in the application are:

1. Table *Variable\_List*. This table has 3 attributes, they are *variable name*, *is\_result* and *is\_active*. Table *Variable\_List* is used to store list of variable that used to make decision tree. *Is\_Result* is told about whether the variable is a result variable or not, while *is\_active* is flagged whether the variable is used or not.
2. Table *Case*. This table has n attributes, they are *variable\_name[1]*, *variable\_name[2]*, ..., *variable\_name[n]*. n is representing count of record in the *variable\_list* table that has value of *is\_active* is True, while *variable\_name[1]*, *variable\_name[2]*, ..., *variable\_name[n]* is value of *variable\_name* that has value of *is\_active* is True. For example, value of

variable list table is shown in table 1.

Table 1. Table Variable\_List

Variable_name	Is_result	Is_active
Windy	False	True
Temperature	False	True
Outlook	False	False
Play	True	True

Based on value of table *variable\_list* cases table will have 3 attribute, they are Windy, Temperature and Play.

3. Table *Tree*. This table has 5 attributes; they are *id\_node*, *node*, *prime*, *value* and *is\_variable*. This table is used for store result of decision tree making process.

There are two kinds of running table in our application, they are:

1. Table *Work[0] .. Work[n-1]*. Attributes of each work table are *Variable\_name* and *gain*. Each work table is created for every calculation cycle to store variable and its gain value.
2. Table *Sub\_work[0]...Sub\_work[n-1]*. Attributes of each sub\_work table are *Variable\_Name*, *Value*, *Result[1]*, ..., *Result[n]*, *count*, and *entropy*. Each *sub\_work* table is created for every calculation cycle to store variable and value. *Result* attribute are value in *result\_variable* of *case* table. Each *result*, *count* and *entropy* attribute is calculate for every combination of *variable\_name* and *value* attribute

Table 2. Cases

<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play</i>
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Cloudy	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Cloudy	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Cloudy	Mild	High	True	Yes
Cloudy	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

To simplify understanding of our tool, we gave cases below of playing tennis decision by considering outlook, temperature, humidity and windy. Cases are shown in table 2.

The first step of using our tool is to input variable that is used in decision making process. This process is shown in table figure 2. After that, analyst can input cases from cases input interface that shown in figure 3.

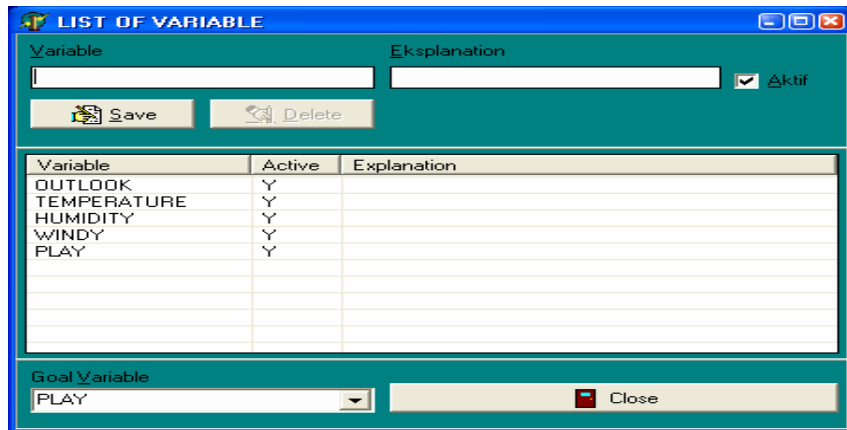


Figure 2. Input variable interface

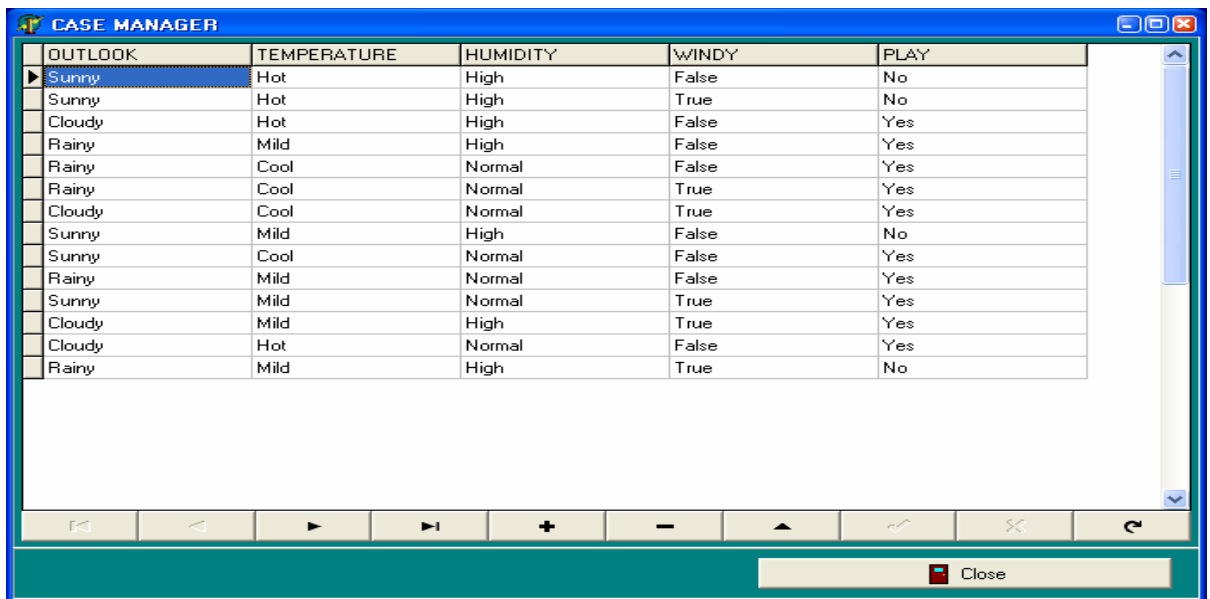


Figure 3. Case Manager Interface

The result of building tree process is shown in figure 4.

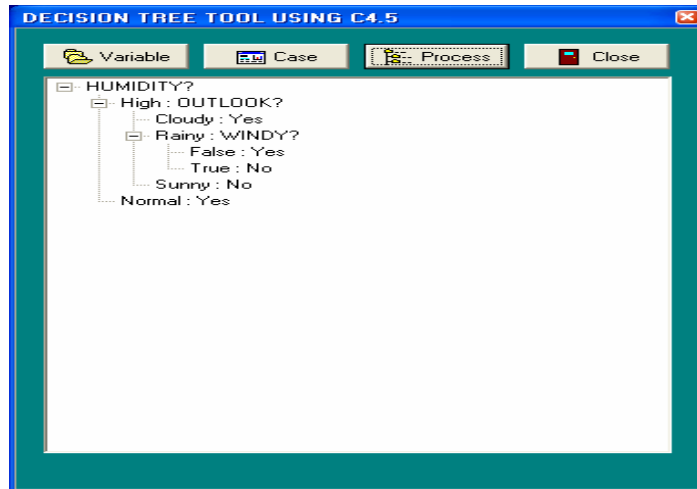


Figure 4. Decision tree building result

The process inside when process button is pressed to build decision tree is:

1. System will take attribute\_name from variable\_list and insert into work\_table that have attribute is\_active true and is\_result false. Example of work table is shown in figure 5

NAMA_ATRIBUT	GAIN
OUTLOOK	0.26
TEMPERATURE	0.18
HUMIDITY	0.37
WINDY	0.01

Figure 5. Example of work table

2. For each attribute\_name, system will find their value and then calculate total count of case, count of case that have result[n] and entropy using formula 2. attribute\_name, value, total count of case, count of case that have result[n] and entropy are inserted into table sub\_work. The example of table sub\_work is shown in figure 6.

- Entropy that is found in step 2, is used to calculate gain using formula 1. And then gain value will be updated into table work.

Properties for: SUB_KERJAO					
SUB_KERJAO					
Properties	Metadata	Permissions	Data	Dependencies	
NAMA_ATRIBUT	NILAI	ENTROPY	RESULT_1	RESULT_2	JML_KASUS
OUTLOOK	Cloudy	0 0	4		4
OUTLOOK	Rainy	0.72 1	4		5
OUTLOOK	Sunny	0.97 3	2		5
TEMPERATURE	Cool	0 0	4		4
TEMPERATURE	Hot	1 2	2		4
TEMPERATURE	Mild	0.92 2	4		6
HUMIDITY	High	0.99 4	3		7
HUMIDITY	Normal	0 0	7		7
WINDY	False	0.81 2	6		8
WINDY	True	0.92 2	4		6

Figure 6. Example of table sub\_work

- Choose the highest gain value from the table work and insert into table tree. The example of table tree is shown in figure 7

ID_NODE	NODE	NILAI	INDUK	IS_ATRIBUT
1	HUMIDITY	<null>	<null>	Y
2	OUTLOOK	High	1	Y
3	Yes	Cloudy	2	T
4	WINDY	Rainy	2	Y
5	Yes	False	4	T
6	No	True	4	T
7	No	Sunny	2	T
8	Yes	Normal	1	T

Figure 7. Example of table tree

- Repeat until all cases in a class.

### 3. CONCLUDING REMARKS

The application of building decision tree using C4.5 algorithm has been implemented and running well. The application is a tool for building decision tree of categorical data. However, it does not handle pre processing data. The assumption is that data has been preprocessed before until they ready to be inserted to case table.

### REFERENCES

1. Craw, S., Case Based Reasoning : Lecture 3: CBR Case-Base Indexing, [www.comp.rgu.ac.uk/staff/smc/teaching/cm3016/Lecture-3-cbr-indexing.ppt](http://www.comp.rgu.ac.uk/staff/smc/teaching/cm3016/Lecture-3-cbr-indexing.ppt)
2. Berry, Michael J.A., Linoff, Gordon S., *Data Mining Techniques For Marketing, Sales, and Customer Relationship Management Second Edition*, Wiley Publishing, Inc., Indianapolis, Indiana, 2004
3. Kusriani, Hartati, S., *Implementation of C4.5 algorithm to evaluate the cancellation possibility of new student applicants at STMIK AMIKOM Yogyakarta*. Proceeding of International Conference on Electrical Engineering and Informatics, 2007.
4. Larose, Daniel T., *Discovering Knowledge in Data: an Introduction to Data Mining*, John Wiley and Sons, USA, 2005

Kusriani: STMIK AMIKOM Yogyakarta, Indonesia. E-mail: kusriani@amikom.ac.id.