Assignment B11
Roll no. 4234

Title :

Implement Decision Trees on Digital Library Data

1 Problem Statement

Implement Decision Trees on Digital Library Data to mirror more titles(PDF) in the library application, compare it with Navie Bayes Algorithm.

2 Objectives

- To learn decision tree based algorithm for classification.
- To implement the Decision Tree algorithm.
- To show comparative study between Decision tree algorithm and Navie Bayes Algorithm.

3 Theory

3.1 Decision Tree :

- Decision tree learning, used in data mining and machine learning, uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value.
- In these tree structures, leaves represent classifications and branches represent conjunctions of features that lead to those classifications.
- In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data but not decisions; rather the resulting classification tree can be an input for decision making. This page deals with decision trees in data mining.
- Decision tree learning is a common method used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables.
- Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable.
- Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

- A tree can be ”learned” by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions.
- In data mining, trees can be described also as the combination of mathematical and computational techniques to aid the description, categorization and generalization of a given set of data.
- Data comes in records of the form:
  (x, y) = (x1, x2, x3..., xk, y)

  The dependent variable, Y, is the target variable that we are trying to understand, classify or generalize. The vector x is comprised of the input variables, x1, x2, x3 etc., that are used for that task.

3.2 Types of trees :

In data mining, trees have additional categories:

- Classification tree analysis is when the predicted outcome is the class to which the data belongs.
- Regression tree analysis is when the predicted outcome can be considered a real number (e.g. the price of a house, or a patients length of stay in a hospital).
- Classification And Regression Tree (CART) analysis is used to refer to both of the above procedures, first introduced by Breiman et al.
- CHi-squared Automatic Interaction Detector (CHAID). Performs multi-level splits when computing classification trees.
- A Random Forest classifier uses a number of decision trees, in order to improve the classification rate.
- Boosting Trees can be used for regression-type and classification-type problems

3.3 Algorithm :

Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition D.

Input:

- Data partition, D, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split point* or *splitting subset*.

Output: A decision tree.

Method:

(1) create a node N;
(2) **if** tuples in D are all of the same class, C **then**
(3)   return N as a leaf node labeled with the class C;
(4) **if** *attribute_list* is empty **then**
(5)   return N as a leaf node labeled with the majority class in D; // majority voting
(6) apply **Attribute_selection_method**(D, *attribute_list*) to **find** the "best" *splitting_criterion*;
(7) label node N with *splitting_criterion*;
(8) **if** *splitting_attribute* is discrete-valued **and**
    multiway splits allowed **then** // not restricted to binary trees
(9)   *attribute_list* $\leftarrow$ *attribute_list* $-$ *splitting_attribute*; // remove *splitting_attribute*
(10) **for each** outcome j of *splitting_criterion*
    // partition the tuples and grow subtrees for each partition

3.4 Input :

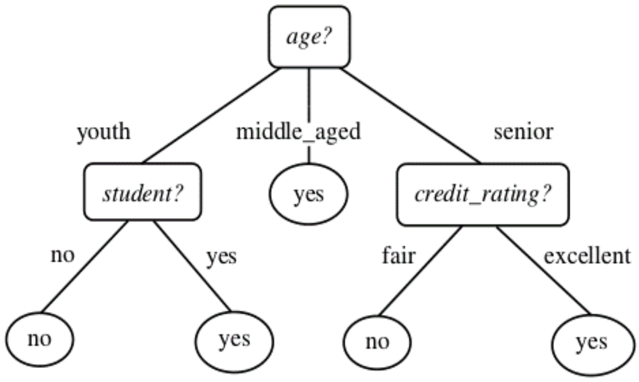
The Training Data Set D:

Table Class-labeled training tuples from the *AllElectronics* customer database.

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

3.5 Output:

Adecision tree :



3.6 ADVANTAGES:

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

- Simple to understand and interpret. People are able to understand decision tree models after a brief explanation.
- Requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed.
- Able to handle both numerical and categorical data. Other techniques are usually specialized in analyzing datasets that have only one type of variable. Ex: relation rules can be used only with nominal variables while neural networks can be used only with numerical variables.
- Use a white box model. If a given situation is observable in a model the explanation for the condition is easily explained by Boolean logic. An example of a black box model is an artificial neural network since the explanation for the results is difficult to understand.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
- Robust. Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

4 Mathematical Model :

Let S be the solution perspective of the class such that

S={s, e, i, o, f, DD, NDD, success, failure}

s={Initial state of the class}
e={End state or destructor of the class}
i={I1} where I1 is the set of inputs.
I1={x—x ϵ input file } where input file consist of the records.
o={decision tree, compare naive bayes}

where,
decision tree = display the decision tree according to the input file. compare naive bayes = display the result of comparison between Naive Bayes and decision tree.

F_{me}=set of functions.

F_{me}={f1,f2,f3,f4}
where,
f1= f1 represents the function to read the input from a file .

f2= f2 represents the function to display the decision tree .

f3= f3 represents the function to show comparative study between Naive Bayes and decision tree.

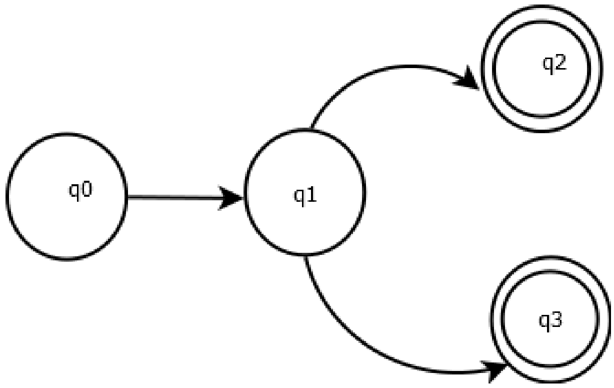
DD (Deterministic Data) = input file

NDD (Non Deterministic Data) = the decision tree and the comparative study

Sc = Success case.
= the decision tree is constructed.

Fc = Failure Case
= the decision tree may not be constructed.

5 State Diagram :



q0 → start state. Accept the input file

q1 → perform decision tree operations on data

q2 → to display decision tree

q3 → to display comparative study between decision tree and Naive Bayes Algorithm.

6 Conclusion

Thus we successfully implemented Decision tree algorithm and have done the comparative study with Naive Bayes Algorithm