RESEARCH ARTICLE OPEN ACCESS

Design and Applications of Decision Trees

Amit Kumar
Department of Computer Science and Engineering,
Guru Nanak Dev Engineering College
Punjab Technical University
Ludhiana, 141006
Punjab –India

ABSTRACT
This paper discusses a basic design procedure for constructing decision trees from examples. We then discuss about some applications of the decision trees. Decision trees are the most important tool which is used in order to reach the final decision regarding any problem. The most important advantage of the decision tree can be seen in the multilevel-Decision making. Decision Trees have been applied in the field of software project request approval and also in predicting policy risk. In this paper we will see some examples from each of these applications to understand how decision trees are important tool for decision oriented fields.

Keywords:- Multilevel decision, Dataset, Policy insurance, Insurance surrenders value.

I. INTRODUCTION

As the name indicates, A Decision Tree is decision inclinable ([1], [2]) tool that uses a graphical [3] tree shaped like structure or multiple level [4] of decision along with their potential outcomes. Objective of the decision tree is prediction by dividing observation into mutually exclusive and exhaustive sub groups. Every branch of the decision tree shows a potential occurrence or decision. Structure of the decision tress shows how one option leads to the next. Main aim of the decision trees is to extract the answer from the complex or uncertain environment. Tree structure allows us to model a complex situation with its potential outcomes in terms of solutions and shaped it in a simply understandable format that describes the relationship among different decisions.

There are three types of elements of the decision tress in terms of nodes-

- Decision node (Root node)
- Chance node/Inner node (Option oriented node)
- Leaf node (Result oriented node)

Root node represents the main problem of any uncertain situation. Final result of the problem will be extracted on the basis of the root node. This node in the decision tree is represented by the rectangle having rounded corners. Chance node or Inner node are option oriented nodes, these nodes consist of some criteria or condition specific requirement and branches from these nodes also consist of possible outcomes. Chance nodes are represented by the rectangle in decision tree. Leaf nodes are said to be result oriented node, consist of decision regarding the problem or situation. These are generally represented by the triangle. We have now discussed about the decision trees and its components, now the next thing is how to design decision trees from the given complex problem or situation.

II. DECISION TREE DESIGN PROCEDURE

Designing of a decision tree 'T' from a Dataset 'D' is 4 steps procedure which is based on Divide and Conquer rule [5]. A dataset is represented with the help of the pair <A, R>, where 'A' is set of attribute in the dataset and 'R' is record of the appropriate attribute.

Suppose we have a dataset
D= {$<A_1, R_1>$, $<A_2, R_2>$ to $<A_i, R_i>$}

Design procedure of the decision tree depends on following steps:

A. ***Step (1):*** If all the records in the dataset are labelled with same attribute then return a leaf node having labelled with that attribute/test node name.

B. ***Step (2):*** We can select some test criterion ‘t’ that consist of two or more test outcomes like ‘t_1’ to ‘t_i’ for i^th records.

C. ***Step (3):*** Now the whole dataset is divided among a set of sub datasets <D_1, D_2, D_3, D_4 to D_i> where ‘D_i’ captures test outcomes ‘t_i’ for that test criteria‘t’.

D. ***Step (4):*** As we know Divide and Conquer [5] consist of relation with recursive [5] nature, and then call this design procedure recursively from ‘D_1’ to ‘D_i’ having results outcomes ‘t_1’ to ‘t_i’. These result outcomes will be sub tree for that constructed tree ‘T’.

This is 4-Step design procedure of decision tree. Explanation of this procedure will prove how decision tree works as decision inclinable ([1], [2]) tool and provide result of a particular problem.

Algorithm:

```
Design_Tree (Dataset D, Tree_Node t, Divide_Selection_Criterion C)
{
Operate C on D to find possible outcomes ($t_1$ to $t_i$).
If (t is not a leaf)
    Create chance node of t and partition D into sub dataset.
    Recursively call each partition.
EndIf
}
```

Explanation: Assume a dataset of a Software organization named ‘XYZ’. Recruitment team of firm ‘XYZ’ want to recruit those candidates (fresher or experience) who fulfil their requirements [8] or conditions, for this the registrations for the applicants are going on. When an applicant registers itself in it then an ID ‘AID’ will be generated automatically for applicant, this ID will be used for further communication between applicant and team. Recruitment team requires some details from the applicants like Name, Father’s name, Applicant’s status (Fresher or Experience), CGPA and Total experience (for professionals). Dataset ‘D’ of ‘XYZ’ consist of following records:

TABLE I
AVAILABLE RECORDS IN DATASET OF ‘XYZ’

AID	Name	Father name	Applicant’s Status	Required Experience	CGPA
A01	Pam	Peter	Fresher	0 year	7.0
A02	Jin	Paul	Fresher	0 year	7.5
A03	Mick	Lee	Experience	3 years	6.0
A04	Nina	Pat	Experience	3 years	7.5
A05	Sam	Duke	Experience	2 years	7.5
A06	Leo	Mike	Experience	2 years	6.0

Recruitment team of ‘XYZ’ consist of these records. Now the main problem of the team is that they have very few positions as comparison to records in Table I. So they have decided to prepare a criteria having some terms or conditions in order to reduce these records to select the applicant for interview. So,

Main problem/situation: how to model this criterion in a structure which provides a solution of this problem in terms of ‘Accept’ or ‘Reject’ the applicant for the interview.

Here decision tree works as decision inclinable ([1], [2]) tool that provide the solution of this problem by model this criterion in multilevel [4] decision fashion. In order to design a decision tree from this available dataset ‘D’ of ‘XYZ’, we can use 4-step design procedure on this example such as:

Step (1): Dataset D= {AI01, AI02, - - - -, AI06} have six mix set of the records then we can move to step 2.

Step (2): Assume we use attribute applicant’s status as test criterion then this criteria can connect with two test outcomes<t_1, t_2> as <Fresher, Experience>. See Fig. 1.

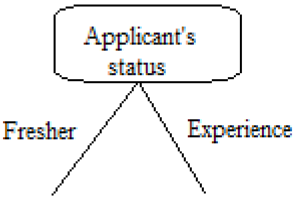

Fig. 1 Decision tree having root node 'Applicant's status' and two outcomes 'Fresher' or 'Experience'.

Step (3): Now whole dataset 'D' is divided into two sub dataset 'd_1' and 'd_2' where as

d1= <AI01, AI02>.

d2= <AI03, AI04, AI05, AI06>.

See Fig. 2.

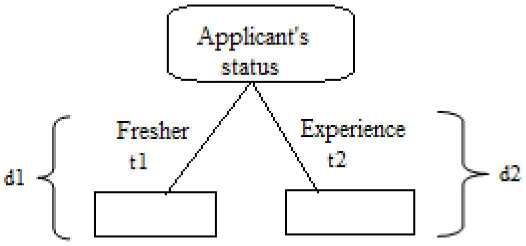

Fig. 2 Decision tree having two sub dataset 'd1' and 'd2' and two outcomes 't1' and 't2'.

Step (4): Now recursively apply this design procedure on 'd_1' and 'd_2' and furthermore until a decision tree 'T' constructed. See Fig. 3

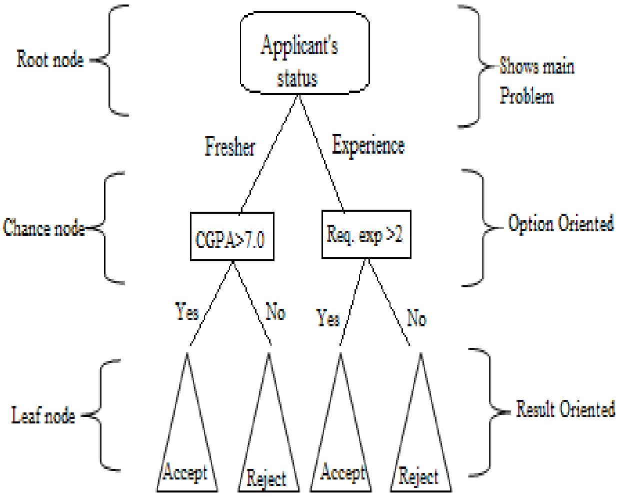

Fig. 3 Final Decision tree for given example.

Now final decision can be obtained for this Dataset 'D' as: See Table II.

TABLE II
AVAILABLE RECORDS IN DATASET OF 'XYZ'

AID	Decision
A01	Reject
A02	Accept
A03	Reject
A04	Accept
A05	Reject
A06	Reject

Thus with the help of this example we can understand how decision tree act as an important decision tool to find out solution of a problem/situation.

III. APPLICATIONS OF DECISION TREES

This section includes the use of the decision trees in practical applications and describes how decision tree learning is used in order to solve the problems.

A. ***Project Request Approval:*** In the software industry when a project comes to an organization then HR and Business analyst of this organization plays an important role in project approval ([6], [7]); with the help of the Fig. 4 and Fig. 5 we can understand how decision trees works in this application.

Assume a customer needs software that can fulfil his/her requirements [8] so customer meets to HR of a software firm and provides all requirements [8]. Assume all the requirements [8] are unambiguous, complete and consistent then HR forwards these requirements [8] to Business analyst, See Fig. 4.

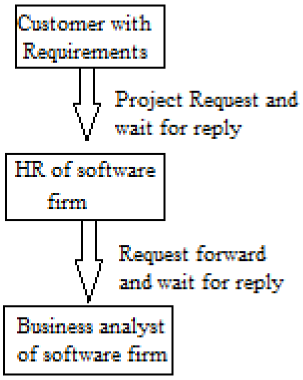
Customer with Requirements
Project Request and wait for reply
HR of software firm
Request forward and wait for reply
Business analyst of software firm

Fig. 4 Initiation of software project request

Now Business analyst prepares decision trees from these requirements [8], here requirements [8] becomes dataset, with the help of the 4-step design procedure of decision trees the result of the root problem (feasible status) can be obtain as 'Accept' or 'Reject' the software project request. See Fig. 5. This status will be forwarded by the business analyst to HR, now HR will take decision on the collected result as whether to accept or reject software request. If project request is approved then corresponding project manager is called to predict several metrics ([9], [10]) and after that other development stages [11] starts.

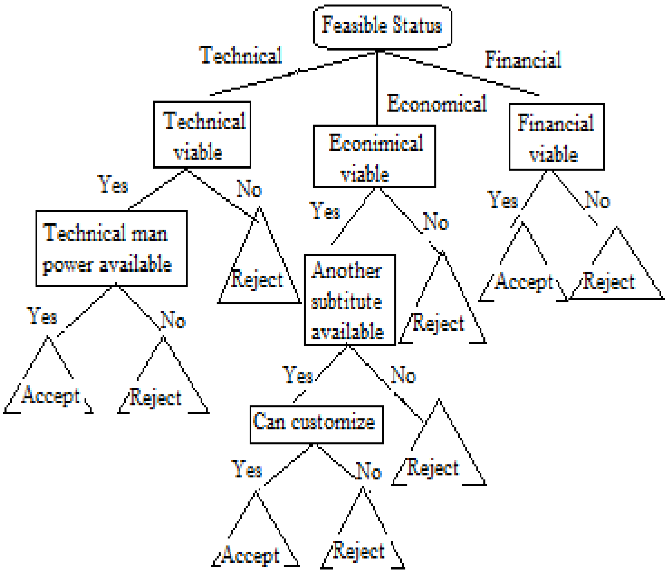

Fig. 5 Final decision tree for the software project request approval application.

Thus Fig. 5 shows how decision tree works in this application.

B. Insurance Policy [12] Suggestion: Decision trees as we know are decision inclinable ([1], [2]) tool, can be used in predicting risk in register a policy [12] for a candidate, here risk in terms of whether to register the policy [12] for the candidate or drop the idea for policy [12] to candidate will be obtained as a result from the leaf node. We can illustrate or understand this application of the decision tree with the help of the example:

Assume an insurance company has just started a policy [12] named as 'PQR'. If any candidate wants to start this policy [12] then he/she must follow some terms and conditions. If any condition does not follow any of these conditions then there can be a risk to register this policy [12] for this candidate. Risk in terms of, if policy [12] starts then the investment in policy [12] from this candidate will consist less growth or also may be not. So all the requirements [8] are available in this decision tree, from these two decisions regarding this policy [12] can be obtain as 'Register' or 'Reject' (see Fig. 6) where register shows that to start this policy [12] by this candidate does not have any risk from required requirements [8] and reject indicates that there can be a risk. Other work of the decision tree can also be in predicting risk in the policy [12] surrender value or insurance surrender value [13].

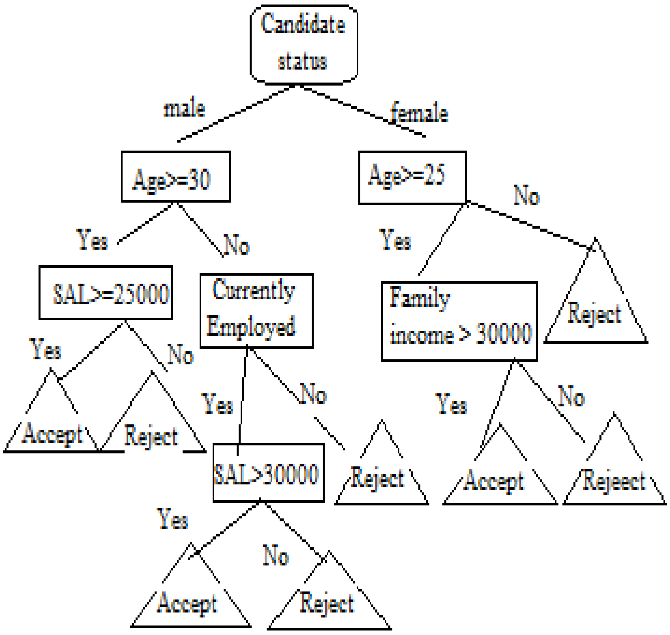

Fig. 6 Decision tree for Insurance Policy Suggestion application.

In this decision tree, two results are present. Here 'Accept' indicates that candidate is able to continue policy [12] after it starts and 'Reject' indicates that candidate may not be able to continue this policy [12] after it starts.

IV. CONCLUSIONS AND FUTURE WORK

This paper state, what the decision trees are and how to design them for a complex situation and also what are the applications of decision trees. Since for large number of conditions and action in the situation, decision trees become complex but they can handle large complex problem in more institutively hierarchical fashion.

Future work will be to find out the use of the decision trees in different fields for decision making.

REFERENCES

[1] Robert N. Britcher. The Limits of Software: People, Projects, and Perspectives. Addison-Wesley Pub Co; 1st edition (June 25, 1999).

[2] Safety brief, Triodyne Inc., safeguard evaluation protocol, vol 11, No 2, may 1995.

[3] Graphical Models in a Nutshell, Daphne Koller, Nir Friedman, Lise Getoor and Ben Taskar.

[4] Multilevel decision trees for static and dynamic pronounciation models, Eric Fosler-Lussier, University of California, Berkeley.

[5] Divide and Conquer (Tom Clancy's Op-Center, Book 7), Steve Pieczenik , Jeff Rovin.

[6] Watss S. Humphrey. "Why Can't We Manage Large Projects?", CrossTalk: The Journal of Defense Software Engineering, Jul/Aug 2010.

[7] Mark J. Christensen, Richard H. Thayer. The Project Manager's Guide to Software Engineering's Best Practices. Wiley-IEEE Press; 1st edition (©2002).

[8] Dorothy Graham. "Requirements and Testing: Seven Missing-Link Myths", IEEE Software Sept./Oct. 2002, pp.15-17.

[9] N.E. Fenton and S.L Pfleeger, (1997), “Software Metrics, A Rigorous & Practical Approach”, International Thomson Computer Press, London 1997, 638 pp.

[10] Gerard O'Regan. A Practical Approach to Software Quality. Springer Verlag; 1st edition (June 13, 2002).

[11] Steve McConnell. Code Complete: A Practical Handbook of Software Construction. Microsoft Press; 2nd edition (June 2004).1st Edition: 1993.

[12] Policy holder’s Perceptions on LIC Policies and Services, International Journal of Management and Social Sciences Research (IJMSSR) ISSN: 2319-4421, Volume 1, No. 1, October 2012.

[13] Surrender value, https://www.licindia.in/policy_guidelines.htm#16.