

## Practical 7

Decision trees are predictive models generated using sample data of a system. These models can be seen as trees. Each node of such a tree contains a simple classification question, which divides up the data without losing any of the data. The simplest way of generating a decision tree is to pick a random variable to start with and to build successively the tree.

### Aims:

1. Being able to build a simple decision tree from a given data set (training set)
2. Being able to reduce the number of nodes in that tree if possible
3. Being able to estimate the error using a separate data set (validation set)
4. Being able to use PROLOG to implement a decision tree and dynamically change the knowledge base using *asserta* and *dynamic* predicates.

### Tasks:

1. A classification system for football clubs need to be built. This system aims to identify football teams that are more likely to be qualified to UEFA Champions league. Hence, a predictive model based on the data of previously qualified teams has to be developed. It is your task to build a decision tree model from the given data set and to estimate the error of the model produced using additional validation data. Generate a decision tree from the following (artificial) data set (hint: you might want to use Market Value as the variable to begin with):

Team No.	Market value (Millions)	Players average age	Shot accuracy	Pass accuracy	Qualified
1	>=400	<23	>=40	>=60	Yes
2	>=400	<23	>=40	<60	Yes
3	>=400	<23	<40	>=60	Yes
4	>=400	<23	<40	<60	No
5	>=400	>=23	>=40	>=60	Yes
6	>=400	>=23	>=40	<60	No
7	>=400	>=23	<40	>=60	No
8	>=400	>=23	<40	<60	No
9	<400	<23	>=40	>=60	Yes
10	<400	<23	>=40	<60	No
11	<400	<23	<40	>=60	No
12	<400	<23	<40	<60	No
13	<400	>=23	>=40	>=60	Yes
14	<400	>=23	>=40	<60	No
15	<400	>=23	<40	>=60	Yes
16	<400	>=23	<40	<60	Yes

2. Can you simplify your model by removing nodes that always produce the same output? (hint: start looking at the leaves of the tree).
3. Use the original data and the following validation data set to calculate the model error in percentage.

Team No.	Market value	Players average age	Shooting accuracy	Passing accuracy	Qualified
17	500	22	32	55	No
18	350	28	45	45	Yes
19	420	23	41	65	No

4. Implement your decision tree predictor using Prolog. You can build on the Prolog code given below to implement your predictor.

```

:-dynamic (marketValue/2). % to informs the interpreter that the definition
:-dynamic (averageAge/2). % of the predicates
:-dynamic (shotAccuracy/2). % marketValue, averageAge , shotAccuracy and passAccuracy
:-dynamic (passAccuracy/2). % may change during execution

marketValue(team1,600). % feed the knowledge base with facts about team1
averageAge(team1,22).
shotAccuracy(team1,78).
passAccuracy(team1,30).

main(Qualified):- % main predicate to execute the predication model
    var(Team), % make sure that "Team" is unbounded variable
    write('What is the name of the team?'),
    read(Team), % read the name of the team
    predict(Team, Qualified). % predict if the team will be qualified or not.

predict(Team, Qualified):-
    % the first rule for predication based on the decision tree model
    ask_marketValue(Team,Value),
    Value>=400,
    ask_averageAge(Team,Avg),
    Avg<23,
    ask_shotAccuracy(Team,ShotAc),
    ShotAc>=60,
    Qualified=yes,!.
%add the rest of rules of your decision tree model here.
%
%
```

```

predict(_,Qualified):-
    Qualified=no.
    % if none of the rules of the decision tree
    % can be applied the predictor should return no

ask_marketValue(Team,Value):-
    marketValue(Team,Value). % to get the value market of the team
ask_marketValue(Team,Value):-
    not(marketValue(Team,Value)), % if market value of a team is not defined as fact
    write('what is the market value of the team?'),
    read(Value),
    % the user can add this fact to the knowledge base
    % using asserta predicate
    asserta(marketValue(Team,Value)),!.

ask_averageAge(Team,Avg):-
    averageAge(Team,Avg).
ask_averageAge(Team,Avg):-
    not(averageAge(Team,Avg)),
    write('what is the average age of the team?'),
    read(Avg),
    asserta(averageAge(Team,Avg)).

ask_shotAccuracy(Team,ShotAc):-
    shotAccuracy(Team,ShotAc).
ask_shotAccuracy(Team,ShotAc):-
    not(shotAccuracy(Team,ShotAc)),
    write('what is the shot accuracy of the team?'),
    read(ShotAc),
    asserta(shotAccuracy(Team,ShotAc)).

% you need to add the definition of the ask_passAccuracy predicate here
%
```