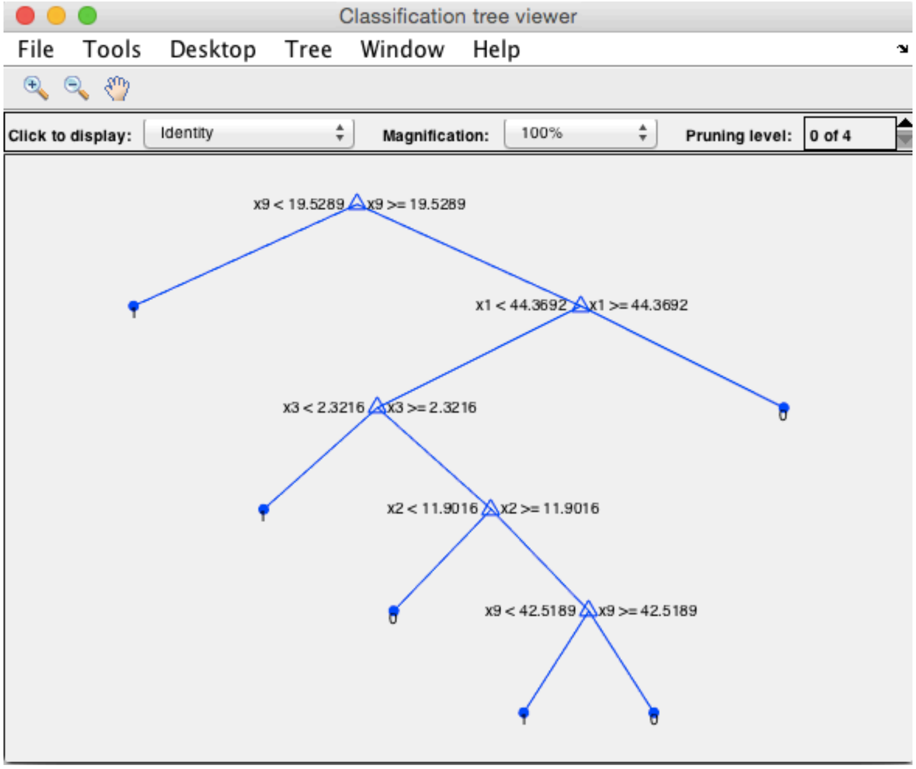
Assignment 2

5a. Graphical view of the decision tree



Click to display: Identity

Magnification: 100%

Pruning level: 0 of 4

5b).
Results of comparing **training data** actual and predicted values

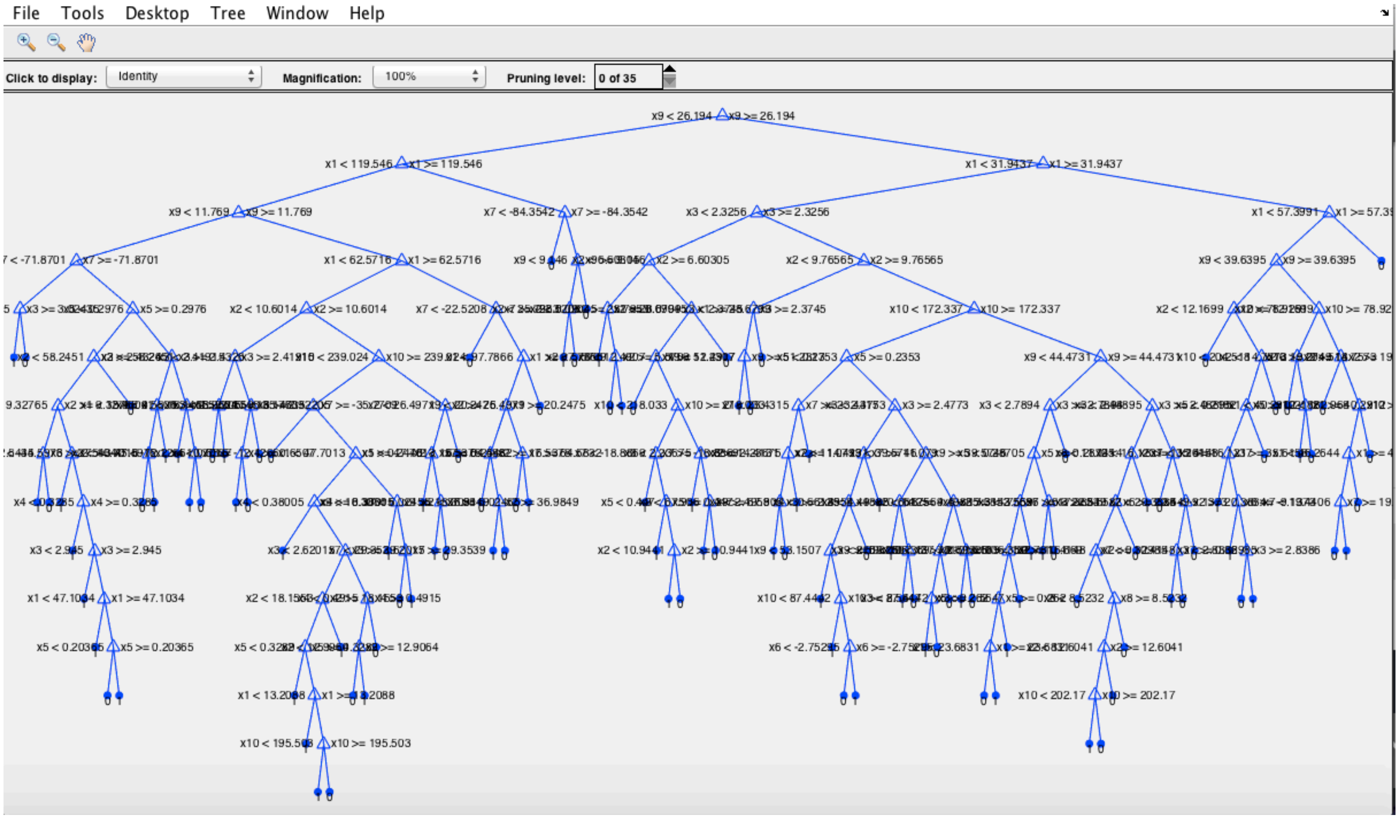
T =

TP	FP	FN	TN	Accuracy	Precision	Recall
8133	308	2519	2060	0.78287	0.96351	0.76352

5c).
Results of comparing **validation data** actual and predicted values
T =

TP	FP	FN	TN	Accuracy	Precision	Recall
1855	70	614	461	0.772	0.96364	0.75132

6a). Graphical view of the decision tree with minleafnodes 20



Click to display: Identity

Magnification: 100%

Pruning level: 0 of 35

Results of comparing **training data** actual and predicted values
T =

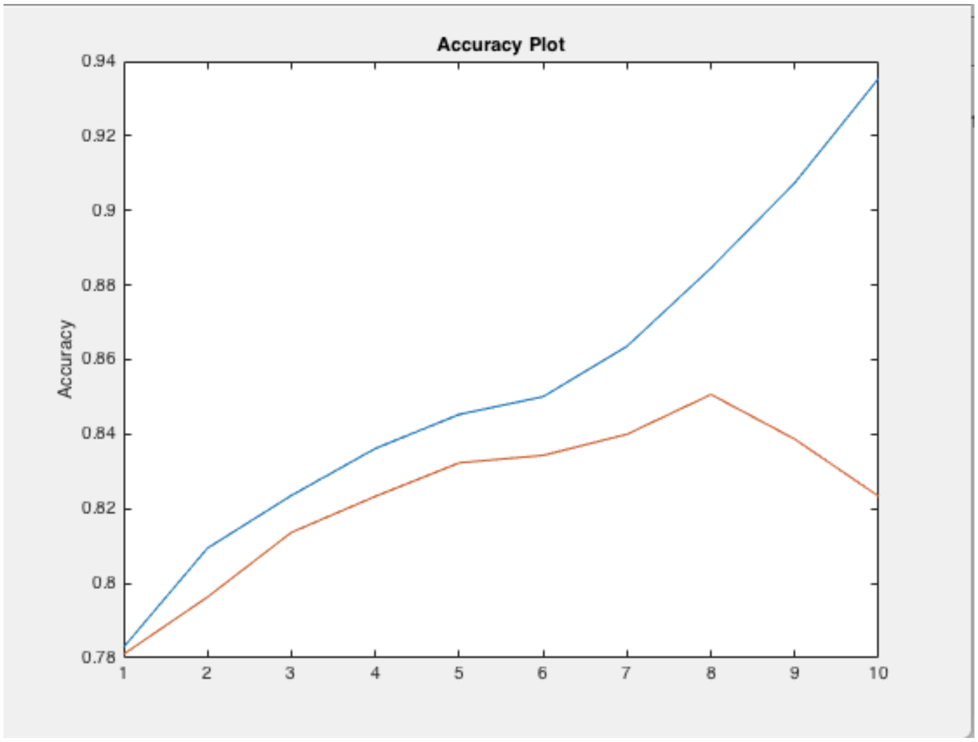
TP	FP	FN	TN	Accuracy	Precision	Recall
7816	587	977	3640	0.87988	0.93014	0.88889

Results of comparing **validation data** actual and predicted values
T =

TP	FP	FN	TN	Accuracy	Precision	Recall
1811	184	257	748	0.853	0.90777	0.87573

7.

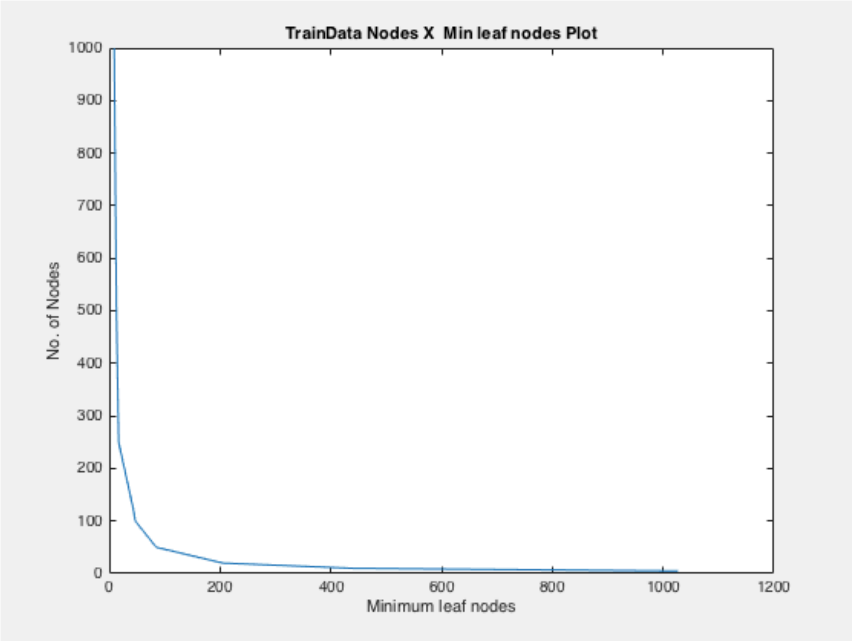
The accuracy of decision trees for the vector of number of leaf nodes 1000, 750, 500, 250, 125, 100, 50, 20, 10, 5 calculated for both training and validation data. Those accuracy values are plotted on the single graph as shown in the following



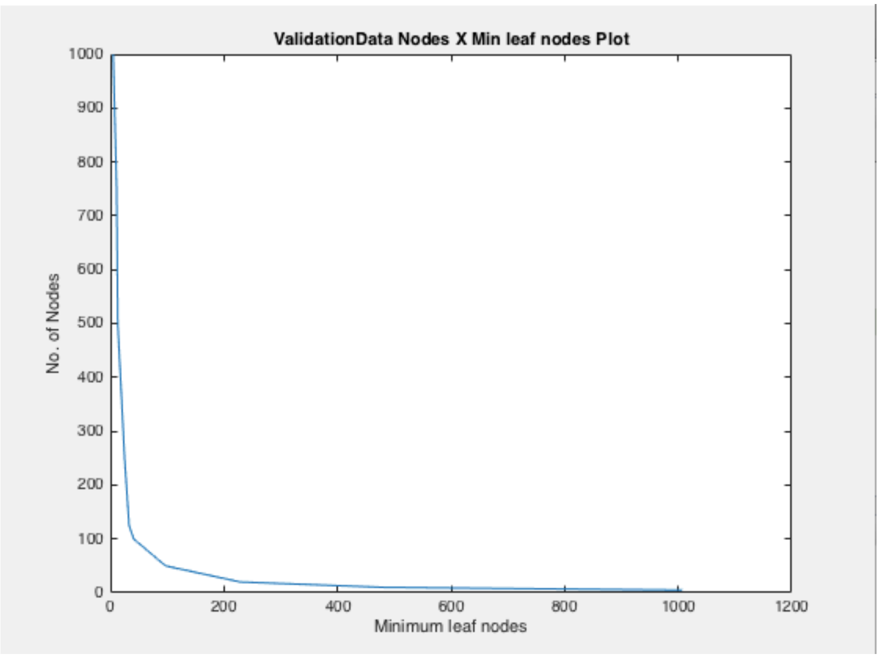
The test data indicated by green plot, the accuracy ranges from 0.78 to 0.93 when the model is tested against training data itself.

But when the model is tested against the validation data the curve raises from 0.78 to 0.85 at the index 8 that means accuracy of the decision tree with minimum number of leaf nodes 20 shows decline in the curve. So that the accuracy till minimum number of leaf nodes from 1000 to 20, the decision tree model is working well. Because that node contains highest accuracy, it is considered best model for the tree.

Train Data: Graph of Number of nodes vs Min no. of leaf Nodes



Validation Data: Graph of Number of nodes vs Min no. of leaf Nodes



8)

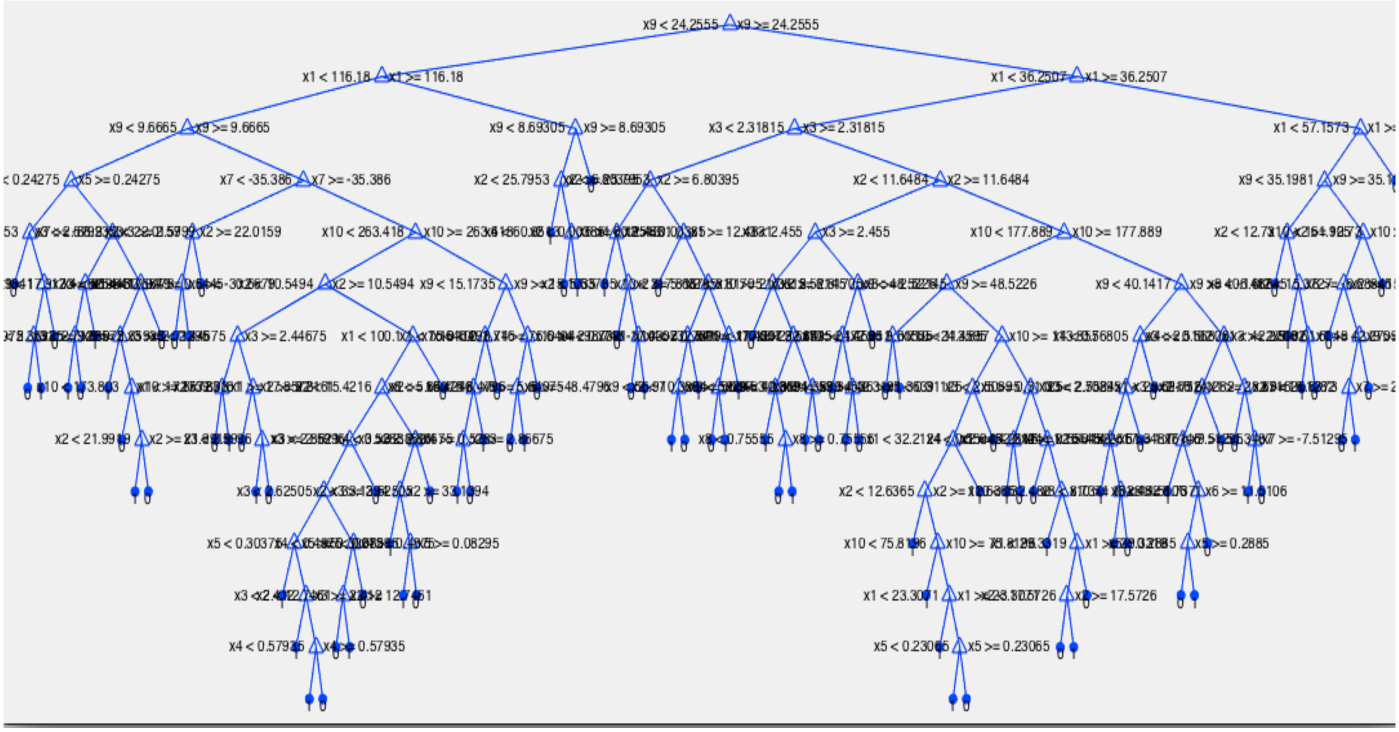
The best model is the decision tree with minimum no. of leaf nodes are 20.

```
Results of comparing training data actual and predicted values
T =

  TP      FP      FN      TN     Accuracy     Precision     Recall
_____    ____    ____    ____    __________    ___________    _________

1768     172     316     744     0.83733      0.91134       0.84837

No. of Nodes   205
```



Matlab Code:

```
% Program 1,
% Splitng the data into 3 parts
 MagicData=xlsread('Magic04.xlsx');
 T = array2table(MagicData,...
     'VariableNames',{'fLength' 'fWidth' 'fSize' 'fConc' 'fConc1'...
     'fAsym' 'fM3Long' 'fM3Trans' 'fAlpha' 'fDist' 'class'});
C=table2cell(T);
% Splitting the data into 3 parts training, validation and test data
[TrainData, ValidationData, TestData]=DatasetPartition(MagicData,C);

% Dataset partition function
function [TrainData, ValidationData, TestData]=DatasetPartition(MagicData,
Cell)
[rows,columns]=size(MagicData);
randIdx=randperm(rows);
trainIdx=randIdx(1,1:13020);
validationIdx=randIdx(1,13021:16020);
testIdx=randIdx(1,16021:19020);

TrainData=Cell(trainIdx,:);
ValidationData=Cell(validationIdx,:);
TestData=Cell(testIdx,:);
end

%Program2
%Split the training data into two tables: features and classlabels

 MagicData=xlsread('Magic04.xlsx');
 T = array2table(MagicData,...
     'VariableNames',{'fLength' 'fWidth' 'fSize' 'fConc' 'fConc1'...
     'fAsym' 'fM3Long' 'fM3Trans' 'fAlpha' 'fDist' 'class'});
C=table2cell(T);
% Splitting the data into 3 parts training, validation and test data
[TrainData, ValidationData, TestData]=DatasetPartition(MagicData,C);

%seperating features from the training data
Features = TrainData(:,1:10);
Features=cell2mat(Features);

%seperating classlables from the training data
ClassLabels = TrainData(:,11);
ClassLabels=cell2mat(ClassLabels);

%Program3
%Generating decision tree
 MagicData=xlsread('Magic04.xlsx');
 T = array2table(MagicData,...
     'VariableNames',{'fLength' 'fWidth' 'fSize' 'fConc' 'fConc1'...
     'fAsym' 'fM3Long' 'fM3Trans' 'fAlpha' 'fDist' 'class'});
C=table2cell(T);
% Splitting the data into 3 parts training, validation and test data
[TrainData, ValidationData, TestData]=DatasetPartition(MagicData,C);

%constructing decision tree by the conditin of leaf node should
% contain minimum of 600 records
 dtr=DesignDecisionTree(TrainData,600);
 view(dtr,'Mode','Graph');

 %Program 4
 %decision tree can be used to find the predicted class labels
  MagicData=xlsread('Magic04.xlsx');
 T = array2table(MagicData,...
     'VariableNames',{'fLength' 'fWidth' 'fSize' 'fConc' 'fConc1'...
     'fAsym' 'fM3Long' 'fM3Trans' 'fAlpha' 'fDist' 'class'});
C=table2cell(T);
% Splitting the data into 3 parts training, validation and test data
[TrainData, ValidationData, TestData]=DatasetPartition(MagicData,C);
dtr=DesignDecisionTree(ValidationData,600)
view(dtr,'Mode','Graph');
%calculating confusion matrix
[TP,FP,FN,TN]=confusionmatrix(ValidationData,dtr);
% calculating probabilities
[Accuracy,Precision,Recall]=InterpretProbability(TP,FP,FN,TN);


% Program 5
% decision tree from training data such that no leaf node has fewer than 1000
records.
 MagicData=xlsread('Magic04.xlsx');
 T = array2table(MagicData,...
     'VariableNames',{'fLength' 'fWidth' 'fSize' 'fConc' 'fConc1'...
     'fAsym' 'fM3Long' 'fM3Trans' 'fAlpha' 'fDist' 'class'});
C=table2cell(T);
% Splitting the data into 3 parts training, validation and test data
[TrainData, ValidationData, TestData]=DatasetPartition(MagicData,C);

%constructing decision tree by the conditin of leaf node should
% contain minimum of 1000 records
dtr=DesignDecisionTree(TrainData,1000);
view(dtr,'Mode','Graph');

%calculating confusion matrix for training data

[ConfMat]=confusionmatrix(TrainData,dtr);
fprintf('Results of comparing training data actual and predicted values');
 T = array2table(ConfMat,...
     'VariableNames',{'TP' 'FP' 'FN' 'TN' 'Accuracy'...
     'Precision' 'Recall'})


[ConfMat]=confusionmatrix(ValidationData,dtr);

fprintf('Results of comparing validation data actual and predicted values');
 T = array2table(ConfMat,...
     'VariableNames',{'TP' 'FP' 'FN' 'TN' 'Accuracy'...
     'Precision' 'Recall'})


%Program6
%no leaf node has fewer than 20 records.
 MagicData=xlsread('Magic04.xlsx');
 T = array2table(MagicData,...
     'VariableNames',{'fLength' 'fWidth' 'fSize' 'fConc' 'fConc1'...
     'fAsym' 'fM3Long' 'fM3Trans' 'fAlpha' 'fDist' 'class'});
C=table2cell(T);
% Splitting the data into 3 parts training, validation and test data
[TrainData, ValidationData, TestData]=DatasetPartition(MagicData,C);

%constructing decision tree by the conditin of leaf node should
% contain minimum of 1000 records
dtr=DesignDecisionTree(TrainData,20);
view(dtr,'Mode','Graph');

%calculating confusion matrix for training data

[ConfMat]=confusionmatrix(TrainData,dtr);
fprintf('Results of comparing training data actual and predicted values');
 T = array2table(ConfMat,...
     'VariableNames',{'TP' 'FP' 'FN' 'TN' 'Accuracy'...
     'Precision' 'Recall'})


[ConfMat]=confusionmatrix(ValidationData,dtr);

fprintf('Results of comparing validation data actual and predicted values');
 T = array2table(ConfMat,...
     'VariableNames',{'TP' 'FP' 'FN' 'TN' 'Accuracy'...
     'Precision' 'Recall'})

%  Program 7

 MagicData=xlsread('Magic04.xlsx');
 T = array2table(MagicData,...
     'VariableNames',{'fLength' 'fWidth' 'fSize' 'fConc' 'fConc1'...
     'fAsym' 'fM3Long' 'fM3Trans' 'fAlpha' 'fDist' 'class'});
C=table2cell(T);
% Splitting the data into 3 parts training, validation and test data
[TrainData, ValidationData, TestData]=DatasetPartition(MagicData,C);

 MinLeafNodes=[1000, 750, 500, 250, 125, 100, 50, 20, 10, 5];

%constructing decision tree by the conditin of leaf node should
% contain minimum of 1000 records
TestNumNodes=0;
ValidNumNodes=0;
TestAccuracy=0;
ValidaAccuracy=0;

for idx=1:length(MinLeafNodes)
dtr=DesignDecisionTree(TrainData,MinLeafNodes(1,idx));
%calculating confusion matrix for training data
[ConfMat]=confusionmatrix(TrainData,dtr);
TestAccuracy(idx,1)=ConfMat(1,5);
TestNumNodes(idx,1)=dtr.NumNodes;

[ConfMat]=confusionmatrix(ValidationData,dtr);
ValidaAccuracy(idx,1)=ConfMat(1,5);
ValidNumNodes(idx,1)=dtr.NumNodes;

end
```

```
figure();
plot(TestAccuracy);
title('Accuracy Plot');
ylabel('Accuracy');
hold on

plot(ValidaAccuracy);
ylabel('Accuracy');
hold off

figure();

plot(TestNumNodes,MinLeafNodes);
title('TrainData Nodes X  Min leaf nodes Plot');
xlabel('Minimum leaf nodes');
ylabel('No. of Nodes');



figure();

plot(ValidNumNodes,MinLeafNodes);
title('ValidationData Nodes X Min leaf nodes Plot');
xlabel('Minimum leaf nodes');
ylabel('No. of Nodes');

%Program 8
 MagicData=xlsread('Magic04.xlsx');
 T = array2table(MagicData,...
     'VariableNames',{'fLength' 'fWidth' 'fSize' 'fConc' 'fConc1'...
     'fAsym' 'fM3Long' 'fM3Trans' 'fAlpha' 'fDist' 'class'});
C=table2cell(T);
% Splitting the data into 3 parts training, validation and test data
[TrainData, ValidationData, TestData]=DatasetPartition(MagicData,C);

%constructing decision tree by the conditin of leaf node should
% contain minimum of 1000 records
dtr=DesignDecisionTree(TrainData,20);
view(dtr,'Mode','Graph');

%calculating confusion matrix for training data
[ConfMat]=confusionmatrix(TestData,dtr);
fprintf('Results of comparing training data actual and predicted values');
 T = array2table(ConfMat,...
     'VariableNames',{'TP' 'FP' 'FN' 'TN' 'Accuracy'...
     'Precision' 'Recall'})
```