Question 1) **Take Data2 and split it into randomly selected 210 training instances and remaining 100 as test instance. Create decision trees using the training set and the “minimum records per leaf node” values of 3, 8, 12, 30, and 50.**
a. Show the trees for all the five cases of min record values. Comment on what you see in a comparative analysis of the five trees. Just reporting the numbers is not enough; you must try to give an explanation of the changes observed. Which of these five trees would you prefer to use and why?
b. For each of the five decision trees compute and report the accuracy, precision, and recall values. Comment on the comparison of these values and show these values on a plot. Give your reasons for the observed trends/differences.

Answer 1)a)
Code in MATLAB for Answer 1 :

```
% data2 is provided to us in the question. To import data2, I used "Import Data" in the Home tab.
% data2 is imported as a table in the workspace..
%to randomly select training and testing data. First, I shuffled the data
%using randperm function.

[m,n] = size(data2);
disp(m);
shuffle = randperm(m);

%out of the randomly shuffled data, 210 rows are chosen as training data and
%remaining 100 as test data
trainingData = data2(shuffle(1:210),:);
testDataSet = data2(shuffle(211:310),:);

% assigning the Predictor Data to X
```

```
X = trainingData(:,1:1:end-1);
% assigning the Class Labels to Y
Y = trainingData(:,end);


testData = testDataSet(:,1:1:end-1); %it contains the
Predictor data for predict function
actual = testDataSet(:,end);% it is the desired output
of the predict function

% building a Fit binary classification decision tree:
min records per leaf
% node is 3
tree1 = fitctree(X,Y,'MinLeafSize',3);
%predict function is used to predict the class of test
data
label1 = predict(tree1, testData);

%there are 2 ways to view a decision tree
view(tree1); %this is the text representation which
shows up in the Command Window
view(tree1,'Mode','graph'); %this is the graphical
representation which shows up in the
%classification tree viewer tab


% the group and grouphat should be of the same type.(
Referred the
% confusionmat(group,grouphat) in matlab documentation
)
%table is first converted to an array. Then the array
is converted to a
%categorical array
cActual = table2array(actual);
cActual = categorical(cActual);
% confusion matrix is created which is then further
used to calculate
% precision, recall and accuracy
c1 = confusionmat(cActual,label1);
[p1, r1, pn1, rn1 ,q1 ,pf1, rf1] = measure(c1);
```

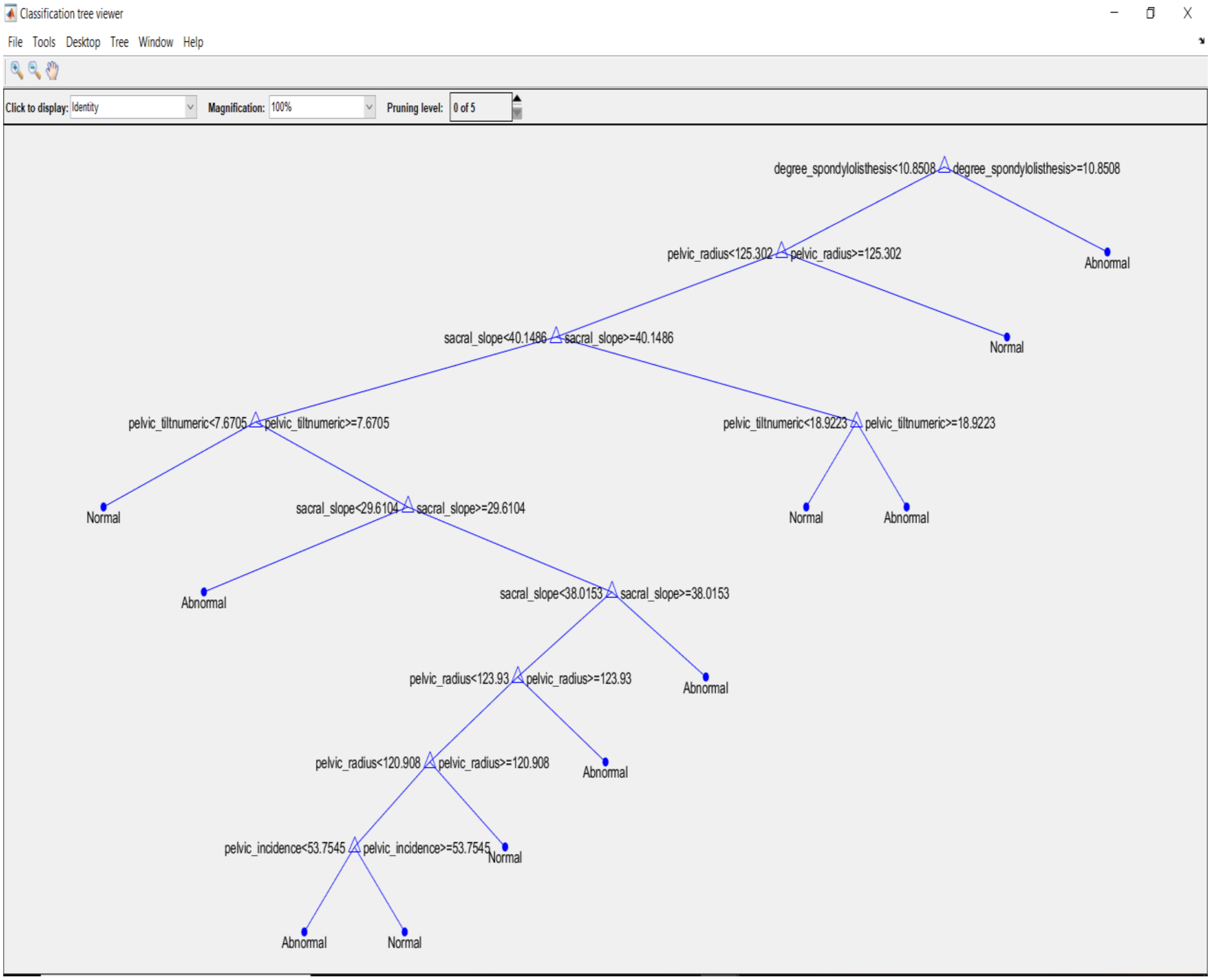
```
% building a Fit binary classification decision tree:
min records per leaf
% node is 8
tree2 = fitctree(X,Y,'MinLeafSize',8);
label2 = predict(tree2, testData);
view(tree2);
view(tree2,'Mode','graph');
c2 = confusionmat(cActual,label2);
[p2, r2, pn2, rn2, q2, pf2, rf2 ] = measure(c2);

% building a Fit binary classification decision tree:
min records per leaf
% node is 12
tree3 = fitctree(X,Y,'MinLeafSize',12);
label3 = predict(tree3, testData);
view(tree3);
view(tree3,'Mode','graph');
c3 = confusionmat(cActual,label3);
[p3, r3, pn3, rn3 ,q3 ,pf3, rf3] = measure(c3);

% building a Fit binary classification decision tree:
min records per leaf
% node is 30
tree4 = fitctree(X,Y,'MinLeafSize',30);
label4 = predict(tree4, testData);
view(tree4);
view(tree4,'Mode','graph');
c4 = confusionmat(cActual,label4);
[p4, r4, pn4, rn4 ,q4 ,pf4, rf4]  = measure(c4);

% building a Fit binary classification decision tree:
min records per leaf
% node is 50
tree5 = fitctree(X,Y,'MinLeafSize',50);
label5 = predict(tree5, testData);
view(tree5);
view(tree5,'Mode','graph');
c5 = confusionmat(cActual,label5);
[p5, r5, pn5, rn5 ,q5 ,pf5, rf5]  = measure(c5);
```

```
%precesion, recall and accuracy values of decision
trees with min records per leaf node
% as 3, 8, 12, 30, 50 respectively
p = [pf1,pf2,pf3,pf4,pf5];
r = [rf1,rf2,rf3,rf4,rf5];
q = [q1,q2,q3,q4,q5];
l = [3,8,12,30,50];

% a combined plot of precision, recall and accuracy
against min records per
% leaf node
plot(l,p, '--or');
hold on;
plot(l,r, '--*k');
plot(l,q, '--+b');
xlabel('min records per leaf node');
f2 = figure();
f2 = plot(l,p, '--or');% precision against min records
per
% leaf node
xlabel('min records per leaf node');
ylabel('precision ');
f3 = figure();
f3 = plot(l,r, '--*k');% recall against min records per
% leaf node
xlabel('min records per leaf node');
ylabel('recall');
f4 = figure();
f4 = plot(l,q, '--+b'); % accuracy against min records
per
% leaf node
xlabel('min records per leaf node');
ylabel('accuracy');

disp("decision tree min records per leaf node is 3");
disp("precision: " +pf1 );
disp("recall: " +rf1);
disp("accuracy: " +q1);
disp("decision tree min records per leaf node is 8");
disp("precision: " +pf2);
disp("recall: " +rf2);
disp("accuracy: " +q2);
disp("decision tree min records per leaf node is 12");
disp("precision: " +pf3 );
disp("recall: " +rf3);
disp("accuracy: " +q3);
disp("decision tree min records per leaf node is 30");
disp("precision: " +pf4 );
disp("recall: " +rf4);
disp("accuracy: " +q4);
disp("decision tree min records per leaf node is 50");
disp("precision: " +pf5 );
disp("recall: " +rf5);
disp("accuracy: " +q5);

%%to calculate precision, recall and accuracy. I have
created a
%%function that takes the confusion matrix as input and
gives the
%%precision, recall and accuracy as output. x
corresponds to the
%%confuion matrix sent as an argument to the function.


function [p, r, pn, rn, a, pf, rf] = measure(x)
tp = x(1,1); %true positive
fn = x(1,2); %false negative
fp = x(2,1); %false positive
tn = x(2,2); %true negative
p = tp/(tp + fp); %positive precision value

r = tp/(tp + fn); %positive recall value
pn = tn/(tn + fn); %negative precision value
rn = tn/(tn + fp); %negative recall value
a = (tp + tn)/ (tp + tn + fp + fn);%accuracy
pf = (p + pn)/2; %mean precision
rf = (r + rn)/2; %mean recall
end
```

Output : Decision Tree when

Answer 1)A

1) Minimum records per leaf node is 3



Click to display: Identity

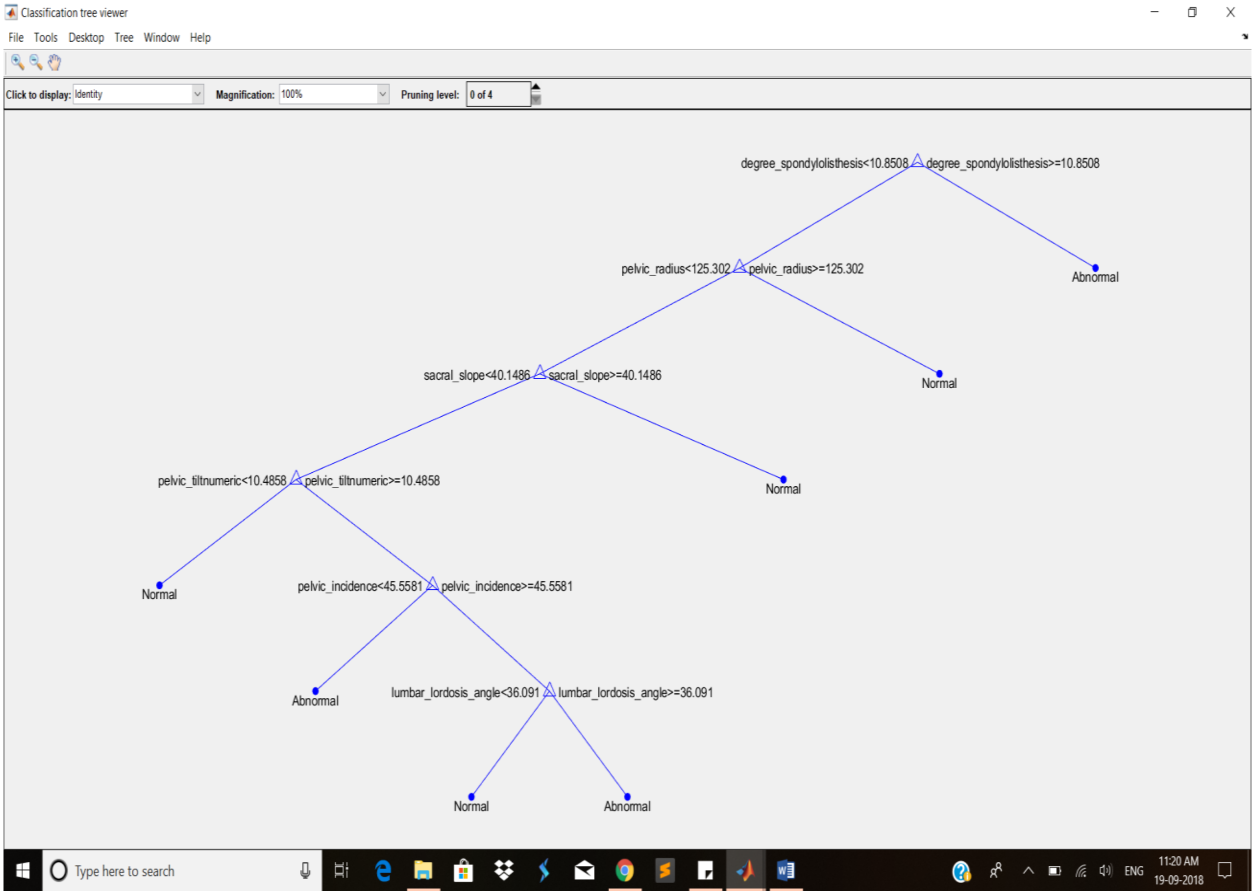
Magnification: 100%

Pruning level: 0 of 5

- ❖ When the minimum records per leaf node is 3, the decision tree has the training data which is **overfitted. Here there is over specific.**
- ❖ Thus, the accuracy of the model is lower than the accuracy with the test data.

- ❖ With the test data, this decision tree will give highest precision, accuracy and recall as it is overfitting the errors too.

2) Minimum records per leaf node is 8

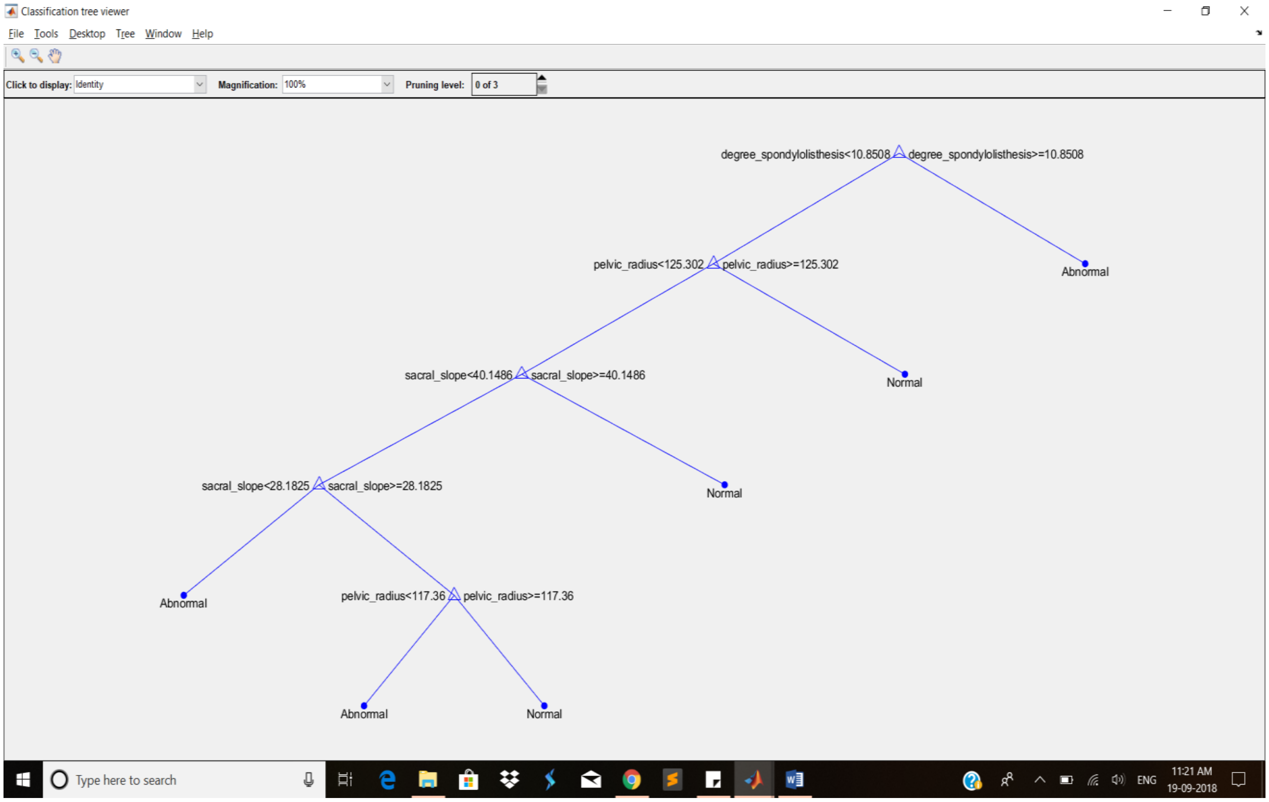


Click to display: Identity Magnification: 100% Pruning level: 0 of 4

Classification tree viewer
File Tools Desktop Tree Window Help
degree_spondylolisthesis<10.8508
degree_spondylolisthesis>=10.8508
pelvic_radius<125.302
pelvic_radius>=125.302
Abnormal
sacral_slope<40.1486
sacral_slope>=40.1486
Normal
pelvic_tiltnumeric<10.4858
pelvic_tiltnumeric>=10.4858
Normal
Normal
pelvic_incidence<45.5581
pelvic_incidence>=45.5581
Abnormal
lumbar_lordosis_angle<36.091
lumbar_lordosis_angle>=36.091
Normal
Abnormal
Type here to search
ENG
11:20 AM
19-09-2018

- ❖ Again, when the minimum records per leaf node is 5, the decision tree has the training data which is **overfitted. Here there is also there is over specific**
- ❖ This decision tree will perform better than the tree with minimum number of records per leaf node as 3

3) Minimum records per leaf node is 12



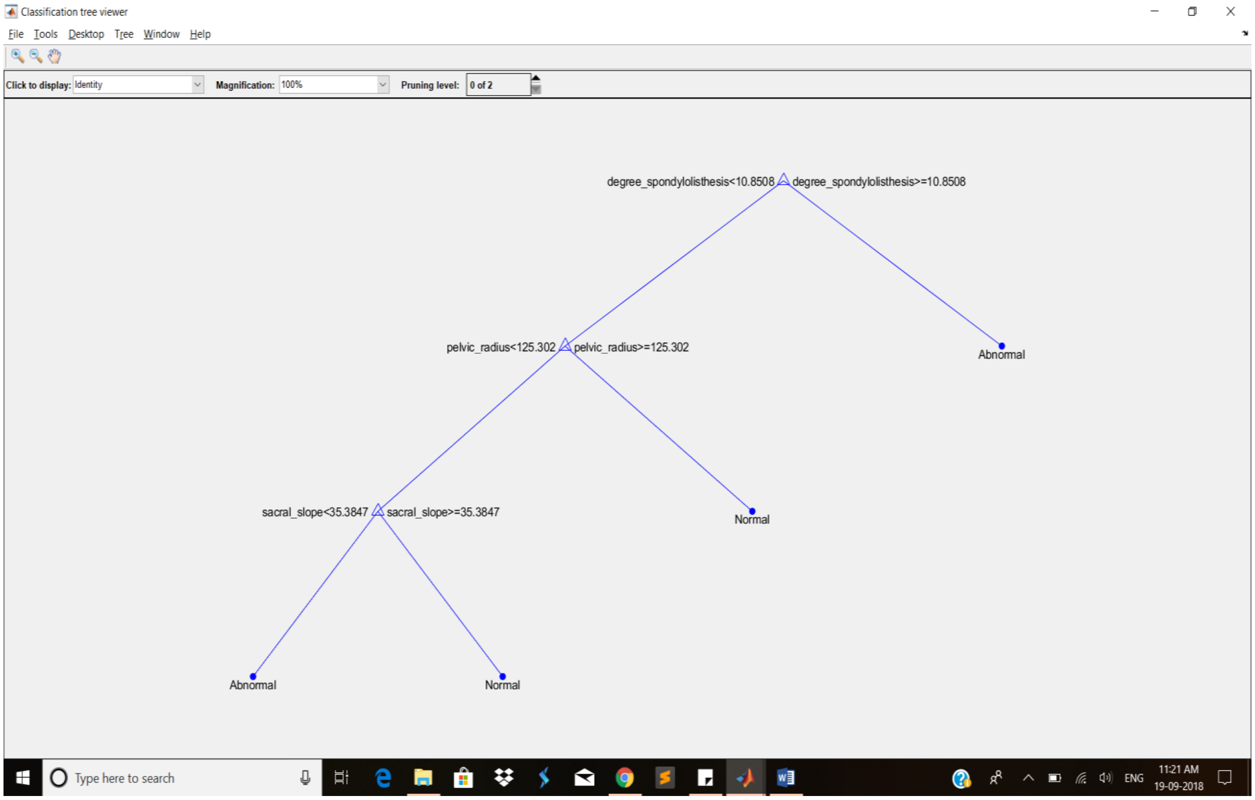
Click to display: Identity

Magnification: 100%

Pruning level: 0 of 3

- ❖ As the minimum records per leaf node increases, **the length or the number of splits to reach the conclusion decreases.**
- ❖ This is a good sign as it **prevents overfitting and over specification of data.**
- ❖ The **performance also increases,** as the accuracy, precision and recall
- ❖ We are moving from general to specific classification.

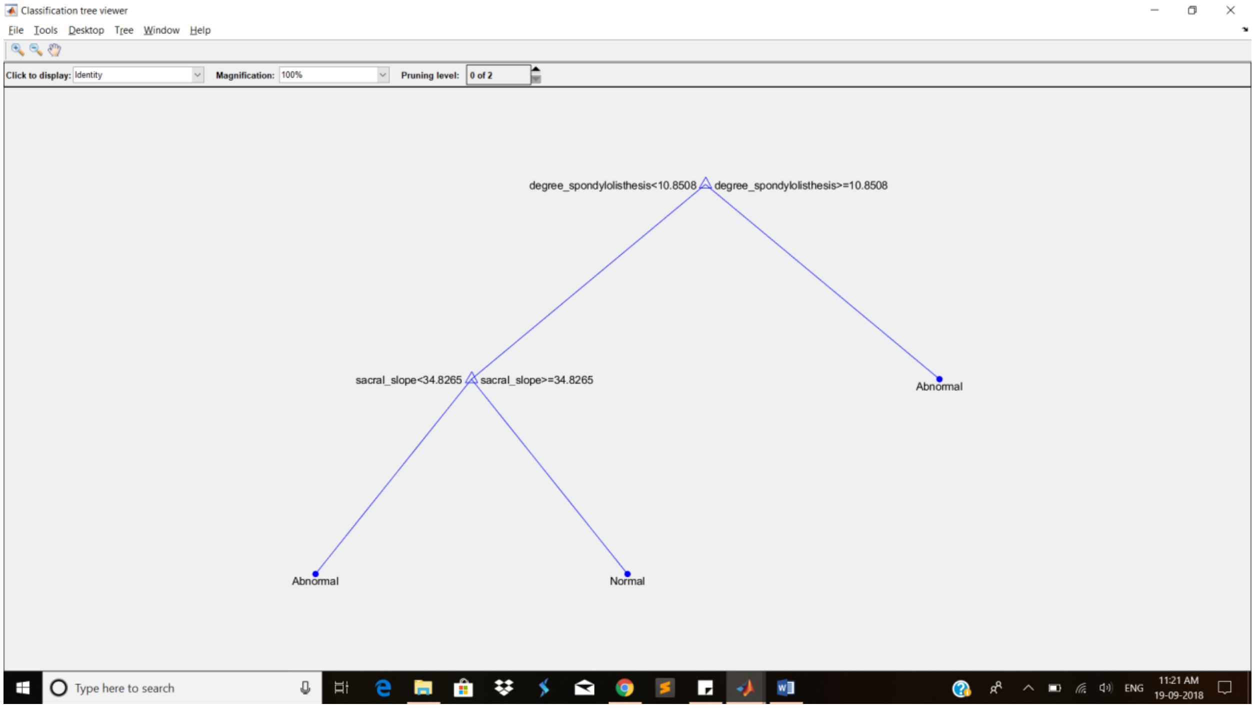
4) Minimum records per leaf node is 30



Click to display: Identity
Magnification: 100%
Pruning level: 0 of 2

- ❖ As the minimum records per leaf node increases. The decision tree keeps on **shrinking, since overfitting is avoided. Here, there are chances of underfitting. The data is overly generalized**
- ❖ The pruning increases, which may increase the performance to some extent. But, the overall performance is reduced as compared to the tree with min records per leaf node as 30.

5) Minimum records per leaf node is 50



Click to display: Identity

Magnification: 100%

Pruning level: 0 of 2

Classification tree viewer
File Tools Desktop Tree Window Help
degree_spondylolisthesis<10.8508
degree_spondylolisthesis>=10.8508
sacral_slope<34.8265
sacral_slope>=34.8265
Abnormal
Abnormal
Normal
Type here to search
ENG
11:21 AM
19-09-2018

- ❖ The performance gets lowest here, the data is underfitted.
- ❖ There is over generalization of data. The results derived from it have more chances of being misfit into a particular class.

Due to all of the above deductions we can conclude that the decision tree with minimum number of records per leaf node as 12 performs the best. It prevents overfitting and underfitting. Also, has the best performance.

Output in the Command Window of MATLAB

Decision tree for classification

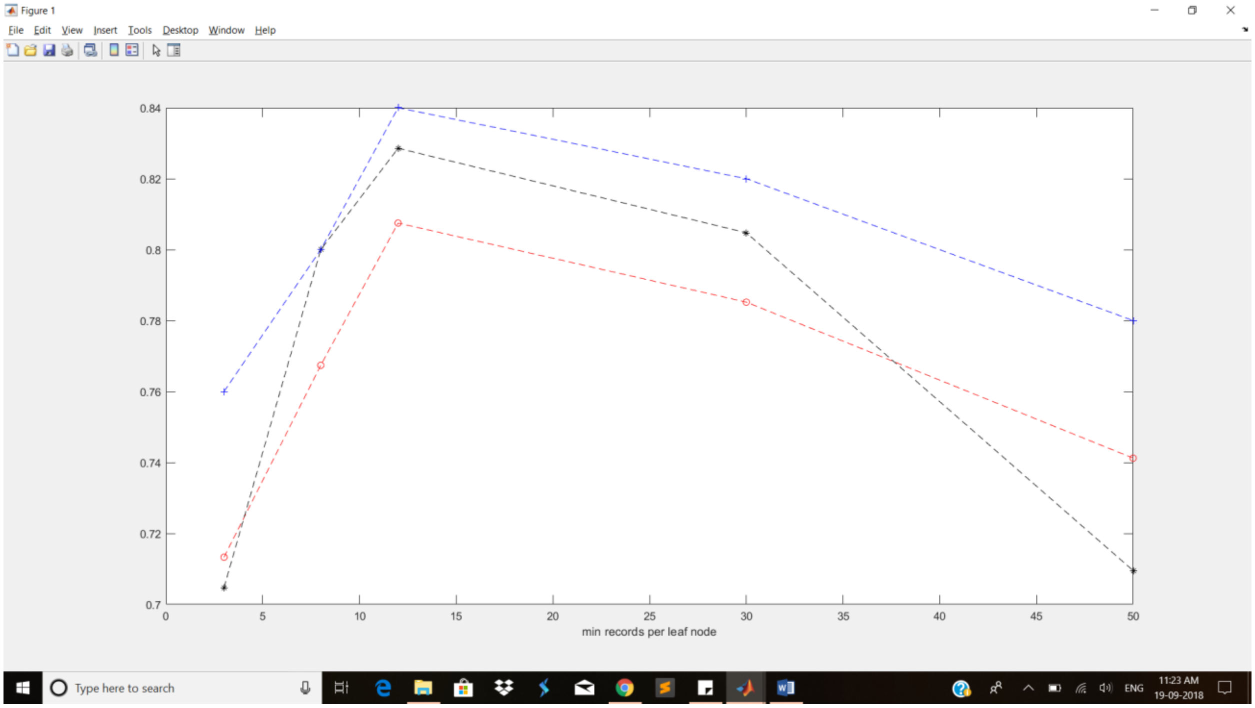
```
 1  if degree_spondylolisthesis<10.8508 then node 2
elseif degree_spondylolisthesis>=10.8508 then node
3 else Abnormal
 2  if pelvic_radius<125.302 then node 4 elseif
pelvic_radius>=125.302 then node 5 else Normal
 3  class = Abnormal
 4  if sacral_slope<40.1486 then node 6 elseif
sacral_slope>=40.1486 then node 7 else Normal
 5  class = Normal
 6  if pelvic_tiltnumeric<7.6705 then node 8 elseif
pelvic_tiltnumeric>=7.6705 then node 9 else
Abnormal
 7  if pelvic_tiltnumeric<18.9223 then node 10
elseif pelvic_tiltnumeric>=18.9223 then node 11
else Normal
 8  class = Normal
 9  if sacral_slope<29.6104 then node 12 elseif
sacral_slope>=29.6104 then node 13 else Abnormal
10  class = Normal
11  class = Abnormal
12  class = Abnormal
13  if sacral_slope<38.0153 then node 14 elseif
sacral_slope>=38.0153 then node 15 else Abnormal
14  if pelvic_radius<123.93 then node 16 elseif
pelvic_radius>=123.93 then node 17 else Normal
15  class = Abnormal
16  if pelvic_radius<120.908 then node 18 elseif
pelvic_radius>=120.908 then node 19 else Normal
17  class = Abnormal
18  if pelvic_incidence<53.7545 then node 20 elseif
pelvic_incidence>=53.7545 then node 21 else
Abnormal
19  class = Normal
20  class = Abnormal
21  class = Normal
```

```
Decision tree for classification
 1  if degree_spondylolisthesis<10.8508 then node 2
elseif degree_spondylolisthesis>=10.8508 then node
3 else Abnormal
 2  if pelvic_radius<125.302 then node 4 elseif
pelvic_radius>=125.302 then node 5 else Normal
 3  class = Abnormal
 4  if sacral_slope<40.1486 then node 6 elseif
sacral_slope>=40.1486 then node 7 else Normal
 5  class = Normal
 6  if pelvic_tiltnumeric<10.4858 then node 8
elseif pelvic_tiltnumeric>=10.4858 then node 9 else
Abnormal
 7  class = Normal
 8  class = Normal
 9  if pelvic_incidence<45.5581 then node 10 elseif
pelvic_incidence>=45.5581 then node 11 else
Abnormal
10  class = Abnormal
11  if lumbar_lordosis_angle<36.091 then node 12
elseif lumbar_lordosis_angle>=36.091 then node 13
else Abnormal
12  class = Normal
13  class = Abnormal
```

```
Decision tree for classification
 1  if degree_spondylolisthesis<10.8508 then node 2
elseif degree_spondylolisthesis>=10.8508 then node
3 else Abnormal
 2  if pelvic_radius<125.302 then node 4 elseif
pelvic_radius>=125.302 then node 5 else Normal
 3  class = Abnormal
 4  if sacral_slope<40.1486 then node 6 elseif
sacral_slope>=40.1486 then node 7 else Normal
 5  class = Normal
 6  if sacral_slope<28.1825 then node 8 elseif
sacral_slope>=28.1825 then node 9 else Abnormal
 7  class = Normal
 8  class = Abnormal
 9  if pelvic_radius<117.36 then node 10 elseif
pelvic_radius>=117.36 then node 11 else Abnormal
10  class = Abnormal
11  class = Normal


Decision tree for classification
1  if degree_spondylolisthesis<10.8508 then node 2
elseif degree_spondylolisthesis>=10.8508 then node
3 else Abnormal
2  if pelvic_radius<125.302 then node 4 elseif
pelvic_radius>=125.302 then node 5 else Normal
3  class = Abnormal
4  if sacral_slope<35.3847 then node 6 elseif
sacral_slope>=35.3847 then node 7 else Normal
5  class = Normal
6  class = Abnormal
7  class = Normal


Decision tree for classification
1  if degree_spondylolisthesis<10.8508 then node 2
elseif degree_spondylolisthesis>=10.8508 then node
3 else Abnormal
2  if sacral_slope<34.8265 then node 4 elseif
sacral_slope>=34.8265 then node 5 else Normal
3  class = Abnormal
4  class = Abnormal
5  class = Normal

decision tree min records per leaf node is 3
precision: 0.71329
recall: 0.70476
```

```
accuracy: 0.76
decision tree min records per leaf node is 8
precision: 0.7674
recall: 0.8
accuracy: 0.8
decision tree min records per leaf node is 12
precision: 0.80749
recall: 0.82857
accuracy: 0.84
decision tree min records per leaf node is 30
precision: 0.7852
recall: 0.80476
accuracy: 0.82
decision tree min records per leaf node is 50
precision: 0.74123
recall: 0.70952
accuracy: 0.78
>>
```

Answer 1) b)



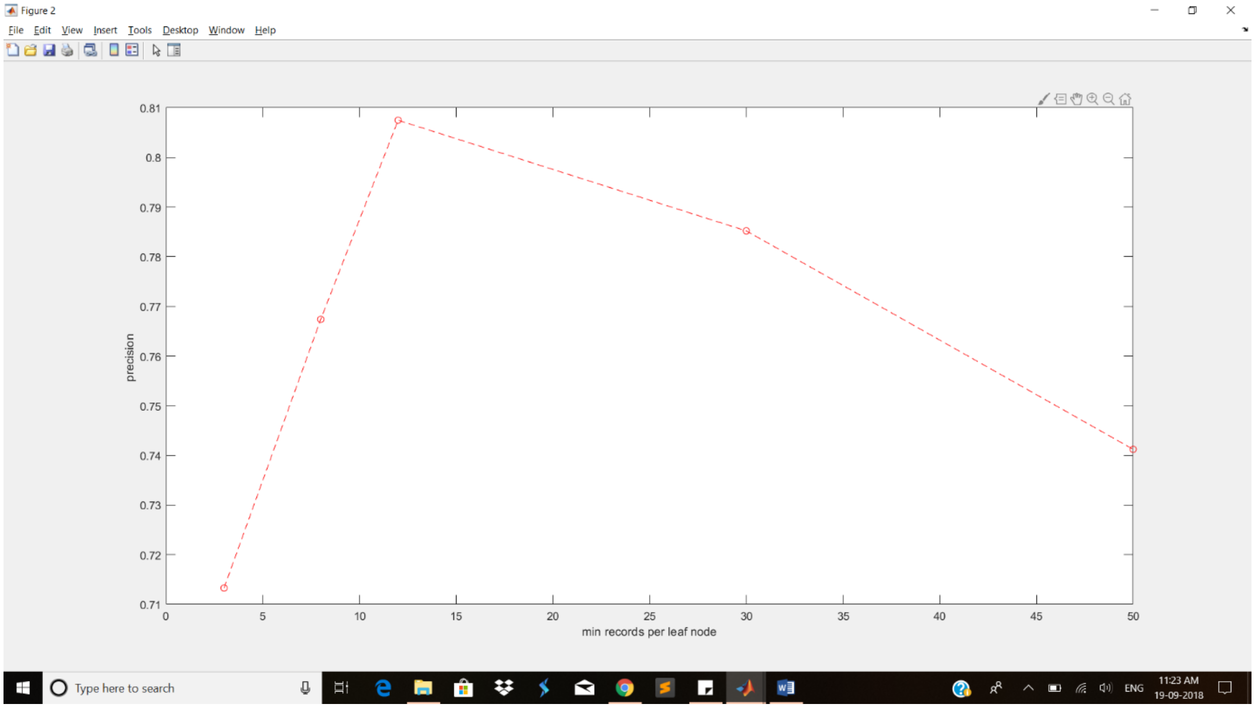
- **The precision, accuracy and recall first increase with the increase in the minimum records per leaf node and then decreases.**
- This is because we are moving from the zone of overfitting the records to the **optimal solution** to underfitting.
- After executing the program multiple times, I observed that the **performance measures are highest when the minimum records per leaf node is 12. This is the optimal solution by far.**

Thus, highest values of performance measures area
precision: 0.80749

recall: 0.82857

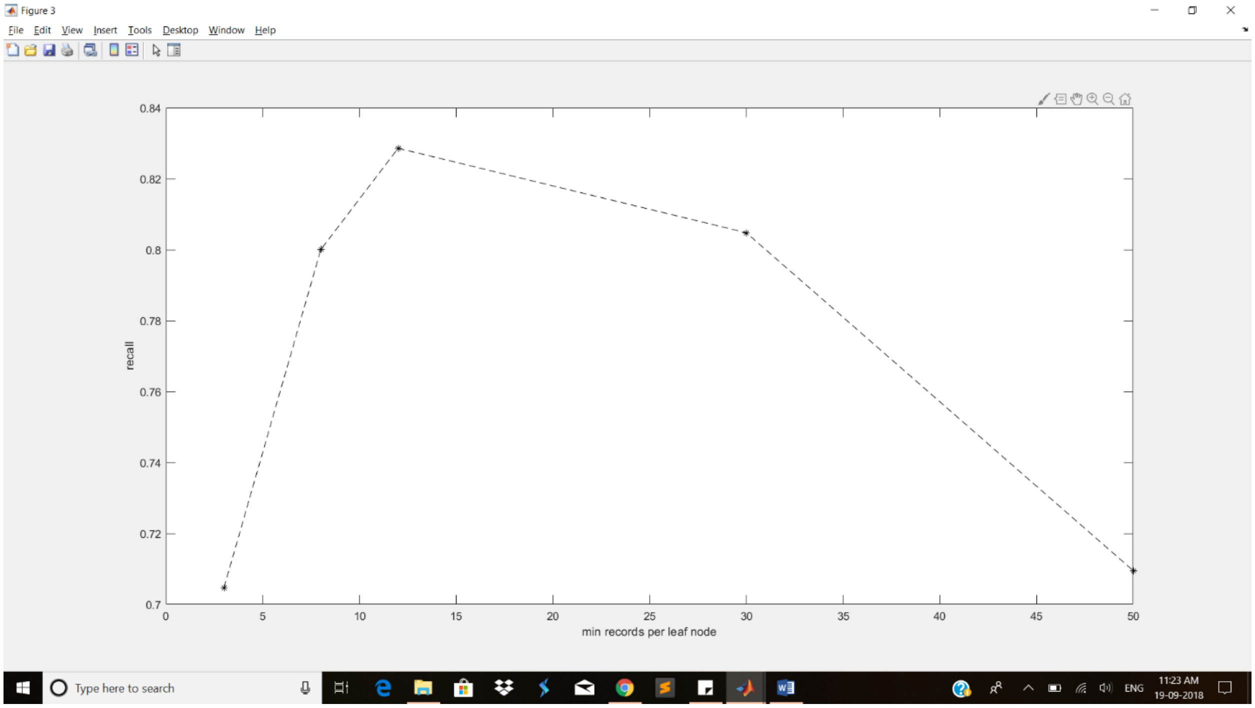
accuracy: 0.84

Figure: precision vs min records per leaf node



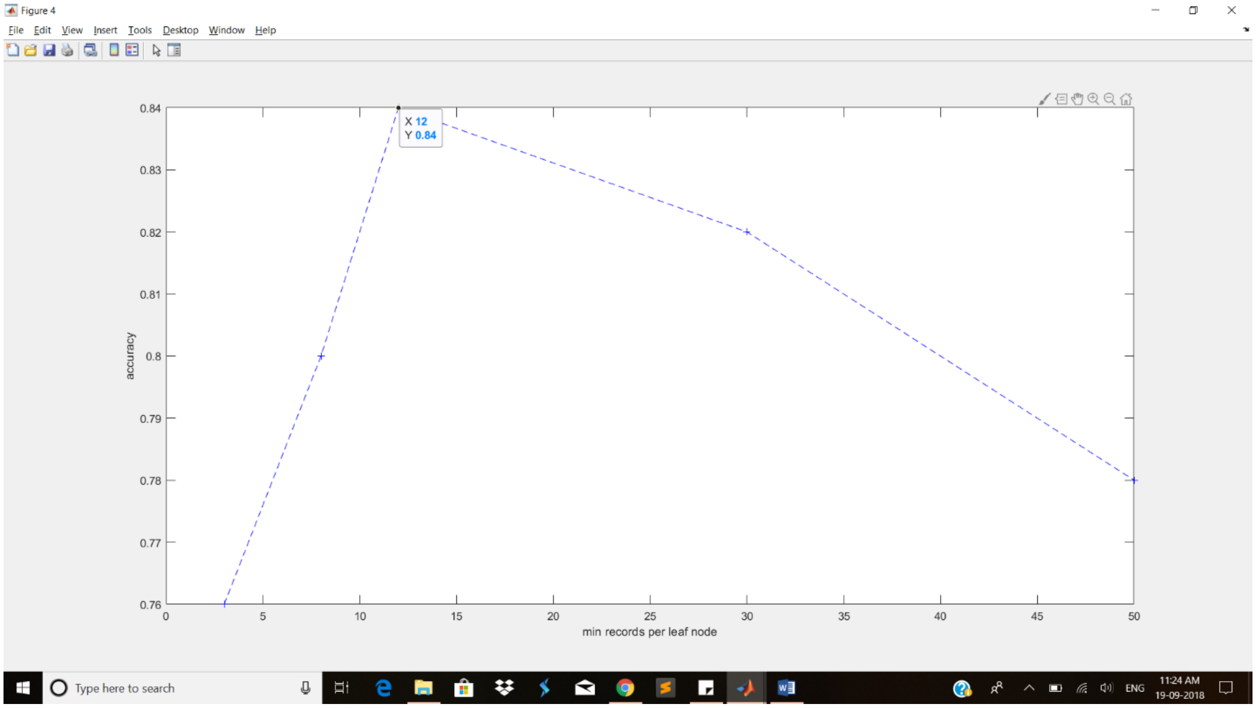
Here, the overfitting occurs at min number of records per leaf node is 3. Overfitting actually captures the noise of the data. As we move further, it is lessened.

Figure: recall vs min records per leaf node



Hence, moving further in the graphs, the common pattern of underfitting is followed.

Figure: accuracy vs min records per leaf node



Accuracy is one of the most good measures for performance. But, it alone cannot be used to deduce conclusions.

Question 2) Repeat the same tasks as done in Question-1 above for Data3 (Now the decision tree has three classes to work with). In addition to reporting results for parts (a) and (b) comment on the comparison of results obtained for (1a) and (2a) and also for (1b) and (2b). Give your analysis for the differences in results. Label this answer as 2c in your submission.

<u>Answer 2</u>

Code in MATLAB for Answer 2 :

```
% data3 is provided to us in the question. To import
data3, I used "Import Data" in the Home tab.
% data3 is imported as a table in the workspace..
%to randomly select training and testing data. First, I
shuffled the data
%using randperm function.

[m,n] = size(data3);
disp(m);
shuffle = randperm(m);

%out of the randomly shuffled data, 210 rows are chosen
as training data and
%remaining 100 as test data
trainingData = data3(shuffle(1:210),:);
testDataSet = data3(shuffle(211:310),:);

% assigning the Predictor Data to X
X = trainingData(:,1:1:end-1);
% assigning the Class Labels to Y
Y = trainingData(:,end);


testData = testDataSet(:,1:1:end-1); %it contains the
Predictor data for predict function
actual = testDataSet(:,end);% it is the desired output
of the predict function

% building a Fit binary classification decision tree:
min records per leaf
% node is 3
tree1 = fitctree(X,Y,'MinLeafSize',3);
%predict function is used to predict the class of test
data
label1 = predict(tree1, testData);

%there are 2 ways to view a decision tree
view(tree1); %this is the text representation which
shows up in the Command Window
```

```
view(tree1,'Mode','graph'); %this is the graphical
representation which shows up in the
%classification tree viewer tab


% the group and grouphat should be of the same type.(
Referred the
% confusionmat(group,grouphat) in matlab documentation
)
%table is first converted to an array. Then the array
is converted to a
%categorical array
cActual = table2array(actual);
cActual = categorical(cActual);
% confusion matrix is created which is then further
used to calculate
% precision, recall and accuracy
c1 = confusionmat(cActual,label1);
[p1, r1,q1] = measure(c1);

% building a Fit binary classification decision tree:
min records per leaf
% node is 8
tree2 = fitctree(X,Y,'MinLeafSize',8);
label2 = predict(tree2, testData);
view(tree2);
view(tree2,'Mode','graph');
c2 = confusionmat(cActual,label2);
[p2, r2, q2 ] = measure(c2);

% building a Fit binary classification decision tree:
min records per leaf
% node is 12
tree3 = fitctree(X,Y,'MinLeafSize',12);
label3 = predict(tree3, testData);
view(tree3);
view(tree3,'Mode','graph');
c3 = confusionmat(cActual,label3);
[p3, r3, q3] = measure(c3);
```

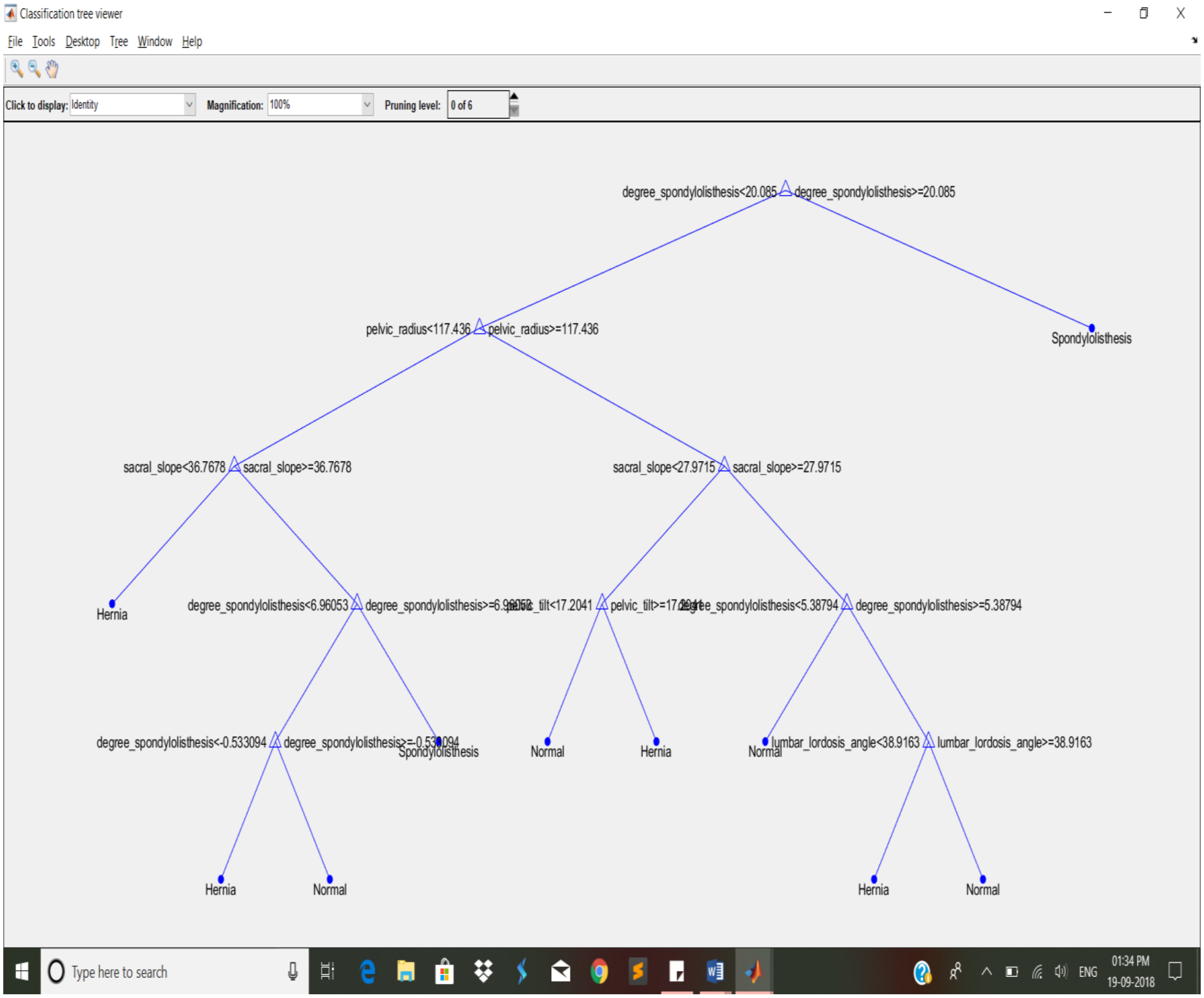
```
% building a Fit binary classification decision tree:
min records per leaf
% node is 30
tree4 = fitctree(X,Y,'MinLeafSize',30);
label4 = predict(tree4, testData);
view(tree4);
view(tree4,'Mode','graph');
c4 = confusionmat(cActual,label4);
[p4, r4,q4]  = measure(c4);

% building a Fit binary classification decision tree:
min records per leaf
% node is 50
tree5 = fitctree(X,Y,'MinLeafSize',50);
label5 = predict(tree5, testData);
view(tree5);
view(tree5,'Mode','graph');
c5 = confusionmat(cActual,label5);
[p5, r5,q5]  = measure(c5);


%precesion, recall and accuracy values of decision
trees with min records per leaf node
% as 3, 8, 12, 30, 50 respectively
pA = [p1(1,1), p2(1,1), p3(1,1), p4(1,1), p5(1,1)];
%precision for class 1 for all  decision trees
pB =[p1(1,2), p2(1,2), p3(1,2),
p4(1,2),p5(1,2)];%precision for class 2 for all
decision trees
pC = [p1(1,3), p2(1,3), p3(1,3), p4(1,3),
p5(1,3)];%precision for class 3 for all  decision trees

rA = [r1(1,1), r2(1,1),r3(1,1), r4(1,1),
r5(1,1)];%recall for class 1 for all  decision trees
rB = [r1(1,2), r2(1,2),r3(1,2), r4(1,2),
r5(1,2)];%recall for class 2 for all  decision trees
rC = [r1(1,3), r2(1,3),r3(1,3), r4(1,3),
r5(1,3)];%recall for class 3 for all  decision trees

q = [q1,q2,q3,q4,q5];
```

```
l = [3,8,12,30,50];

% a combined plot of precision, recall and accuracy
against min records per
% leaf node
f1 = figure();
f1 = plot(l,q, '--+b'); % accuracy against min records
per
% leaf node
xlabel('min records per leaf node');
ylabel('accuracy');


f2 = figure();
f2 = plot(l,pA, '--or');% precision against min records
per
% leaf node
xlabel('min records per leaf node');
ylabel('precision for class 1');

f3 = figure();
f3 = plot(l,pB, '--or');% precision against min records
per
% leaf node
xlabel('min records per leaf node');
ylabel('precision for class 2');


f4 = figure();
f4 = plot(l,pC, '--or');% precision against min records
per
% leaf node
xlabel('min records per leaf node');
ylabel('precision for class 3');


f5 = figure();
f5 = plot(l,rA, '--*k');% recall against min records
per
% leaf node
```

```
xlabel('min records per leaf node');
ylabel('recall for class 1');


f6 = figure();
f6 = plot(l,rB, '--*k');% recall against min records
per
% leaf node
xlabel('min records per leaf node');
ylabel('recall for class 2');

f7 = figure();
f7 = plot(l,rC, '--*k');% recall against min records
per
% leaf node
xlabel('min records per leaf node');
ylabel('recall for class 3');



disp("decision tree min records per leaf node is 3");
disp("accuracy: " +q1);
disp("precision for classes is respectively ");
disp(p1);
disp("recall for all classes is respectively" );
disp(r1);
disp("decision tree min records per leaf node is 8");
disp("accuracy: " +q2);
disp("precision for classes is respectively ");
disp(p2);
disp("recall for all classes is respectively" );
disp(r2);
disp("decision tree min records per leaf node is 12");
disp("accuracy: " +q3);
disp("precision for classes is respectively ");
disp(p3);
disp("recall for all classes is respectively" );
disp(r3);
disp("decision tree min records per leaf node is 30");
disp("accuracy: " +q4);
disp("precision for classes is respectively ");
disp(p4);
disp("recall for all classes is respectively" );
disp(r4);
disp("decision tree min records per leaf node is 50");
disp("accuracy: " +q5);
disp("precision for classes is respectively ");
disp(p1);
disp("recall for all classes is respectively" );
disp(r5);

%%to calculate precision, recall and accuracy. I have
created a
%%function that takes the confusion matrix as input and
gives the
%%precision, recall and accuracy as output. x
corresponds to the
%%confuion matrix sent as an argument to the function.


function [pr,re,a] = measure(x)
pr = [0,0,0]; %array containing precsion of class 1,
class 2, class 3 respectively
re = [0,0,0]; %array containing recall of class 1,
class 2, class 3 respectively
sum =0;
diag = 0;

for i= 1:3
    c_sum = 0;
    r_sum =0;
    for j=1:3
        c_sum = x(j,i) + c_sum; %sum of elements in
column of a particular class represented by i
        r_sum = x(i,j) + r_sum;%sum of elements in row
of a particular class represented by i
    end
    pr(1,i) = x(i,i)/c_sum;
    re(1,i) = x(i,i)/r_sum;
end
```

```
for i = 1:3
    for j = 1:3
        sum = sum + x(i,j); %sum of all the elements in
the confusion matrix

    end
    diag = diag + x(i,i); % sum of all the true
positives in a confusion matrix. All the elements in
the diagonal represent
        % the true positives
end
a = diag/sum;
end
```

Output : Decision Tree when
Answer 2)A
   1)Minimum records per leaf node is 3

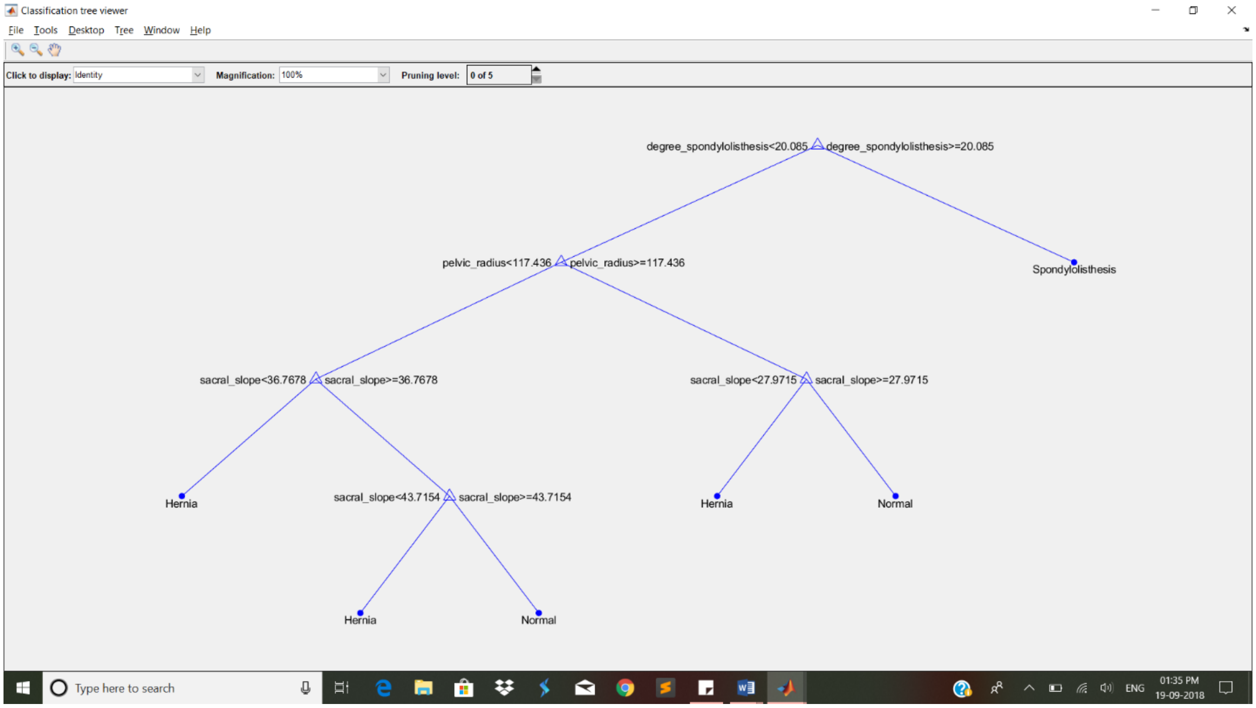

Classification tree viewer
File
Tools
Desktop
Tree
Window
Help
degree_spondylolisthesis<20.085
degree_spondylolisthesis>=20.085
pelvic_radius<117.436
pelvic_radius>=117.436
Spondylolisthesis
sacral_slope<36.7678
sacral_slope>=36.7678
sacral_slope<27.9715
sacral_slope>=27.9715
Hernia
degree_spondylolisthesis<6.96053
degree_spondylolisthesis>=6.96053
pelvic_tilt<17.2041
pelvic_tilt>=17.2041
degree_spondylolisthesis<5.38794
degree_spondylolisthesis>=5.38794
degree_spondylolisthesis<-0.533094
degree_spondylolisthesis>=-0.533094
Spondylolisthesis
Normal
Hernia
Normal
lumbar_lordosis_angle<38.9163
lumbar_lordosis_angle>=38.9163
Hernia
Normal
Hernia
Normal
Type here to search
ENG
01:34 PM
19-09-2018

Click to display: Identity
Magnification: 100%
Pruning level: 0 of 6

- ❖ When the minimum records per leaf node is 3, the decision tree has the training data which is **overfitted. Here there is over specific.**
- ❖ Thus, the accuracy of the model is lower than the accuracy with the test data.

- With the test data, this decision tree will give highest precision, accuracy and recall as it is overfitting the errors too

2)Minimum records per leaf node is 8



Classification tree viewer

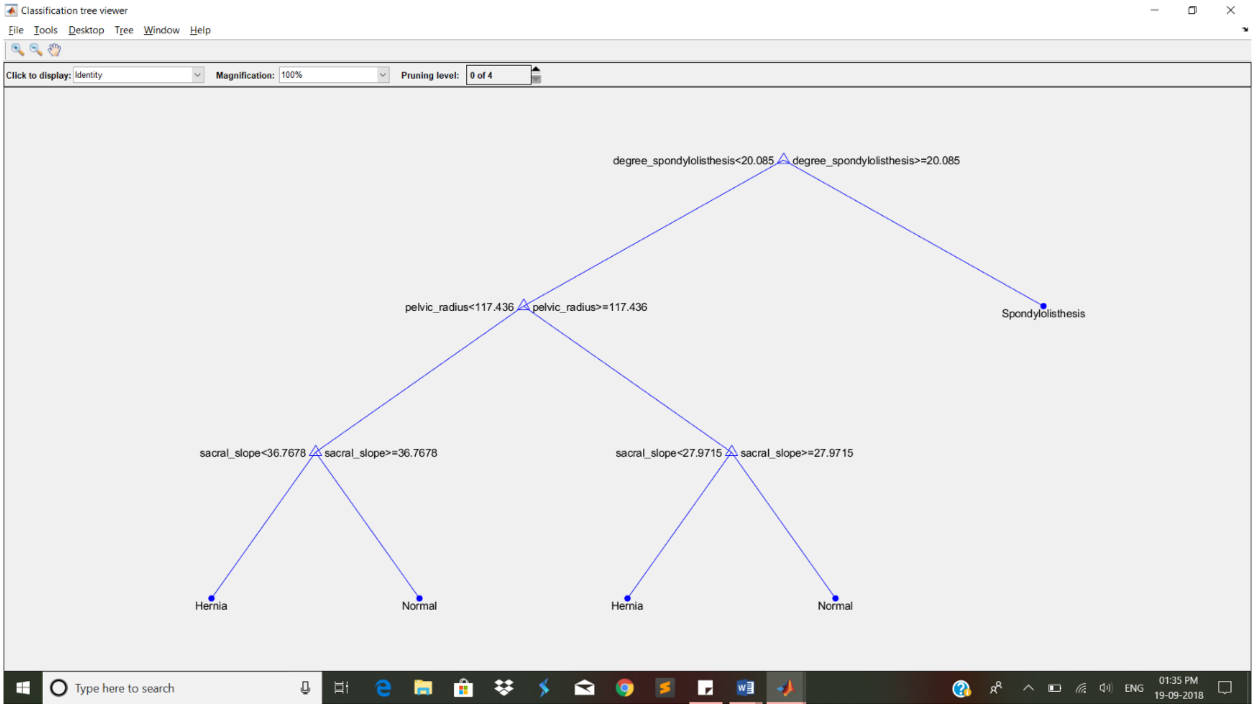
Click to display: Identity

Magnification: 100%

Pruning level: 0 of 5

- This decision tree gives us the best performance as it has the optimal number of test conditions through which a dataset needs to go to be classified as one of the three classes
- After increasing the min number of records per leaf node further the performance will decrease

3)Minimum records per leaf node is 12

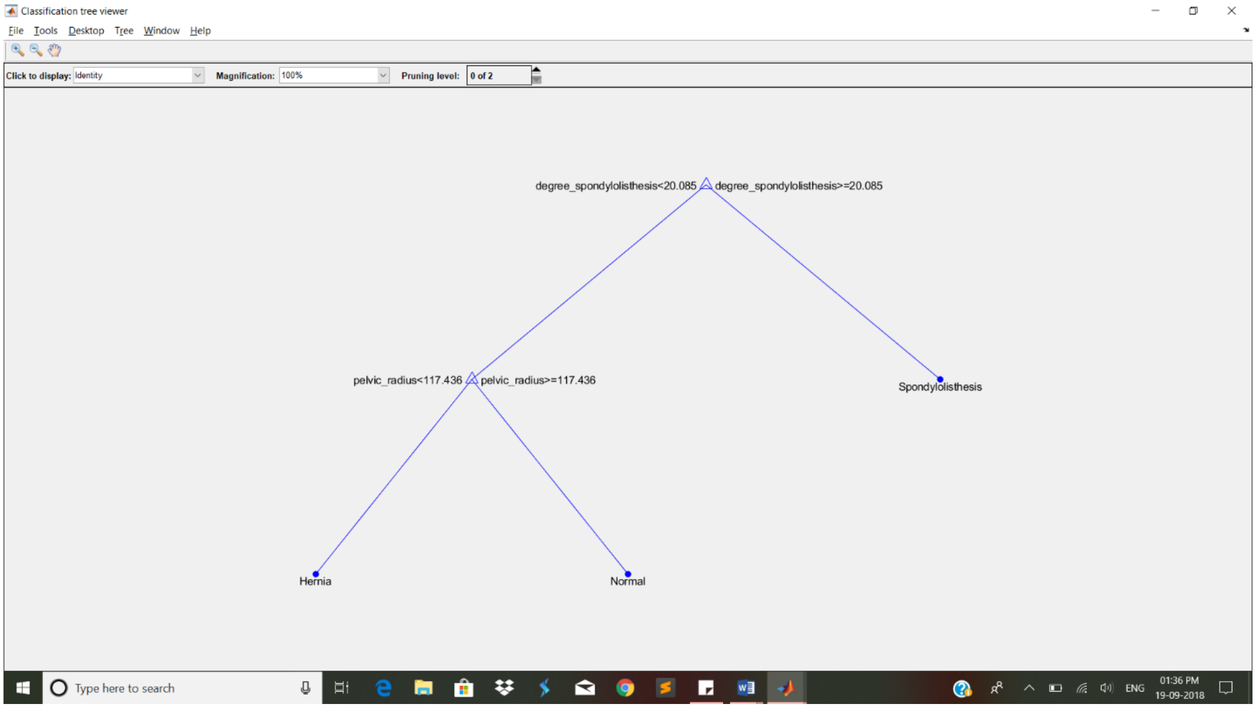


Click to display: Identity

Magnification: 100%

Pruning level: 0 of 4

4) Minimum records per leaf node is 30



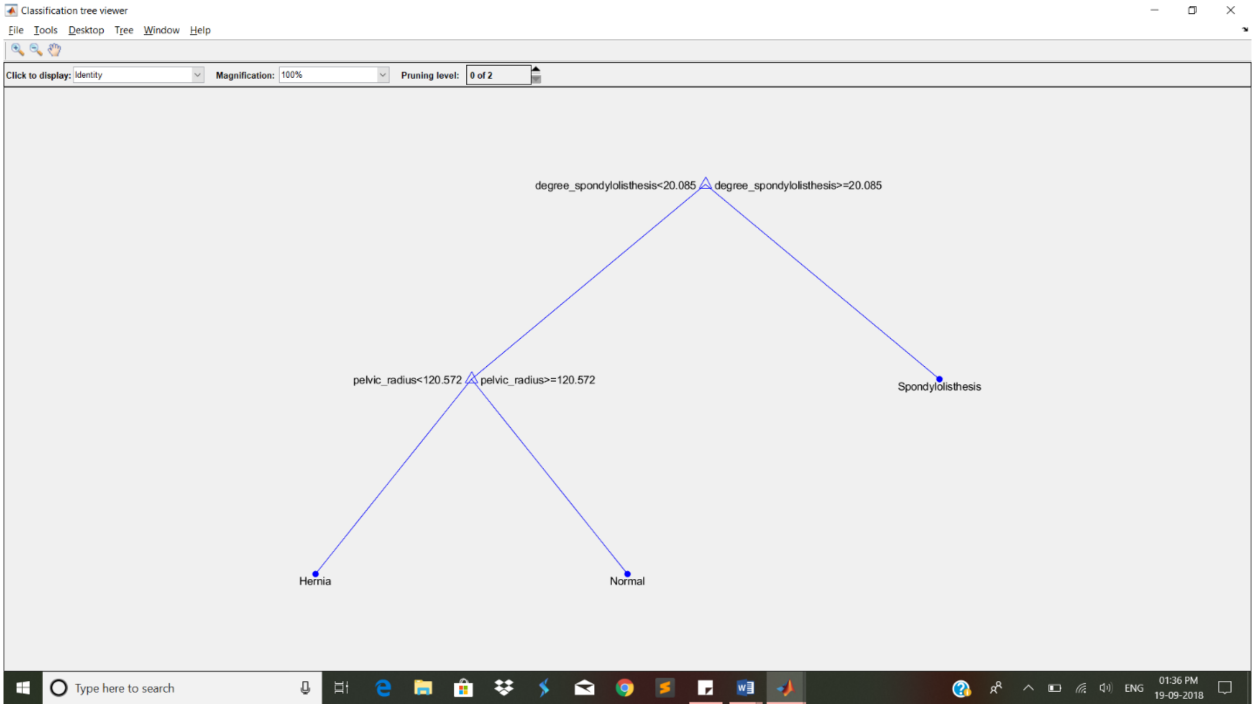
Click to display: Identity

Magnification: 100%

Pruning level: 0 of 2

Classification tree viewer
File Tools Desktop Tree Window Help
degree_spondylolisthesis<20.085
degree_spondylolisthesis>=20.085
pelvic_radius<117.436
pelvic_radius>=117.436
Spondylolisthesis
Hernia
Normal
Type here to search
ENG
01:36 PM
19-09-2018

5)Minimum records per leaf node is 50


Classification tree viewer
File Tools Desktop Tree Window Help
degree_spondylolisthesis<20.085
degree_spondylolisthesis>=20.085
pelvic_radius<120.572
pelvic_radius>=120.572
Spondylolisthesis
Hernia
Normal
Type here to search
ENG
01:36 PM
19-09-2018

Click to display: Identity

Magnification: 100%

Pruning level: 0 of 2

- As the minimum records per leaf node increases. The decision tree keeps on **shrinking, since overfitting is avoided. Here, there are chances of underfitting. The data is overly generalized**
- The pruning increases, which may increase the performance to some extent. But, the overall performance is reduced as compared to the tree with min records per leaf node as 30.

```
Decision tree for classification
 1  if degree_spondylolisthesis<20.085 then node 2
elseif degree_spondylolisthesis>=20.085 then node 3
else Spondylolisthesis
 2  if pelvic_radius<117.436 then node 4 elseif
pelvic_radius>=117.436 then node 5 else Normal
 3  class = Spondylolisthesis
 4  if sacral_slope<36.7678 then node 6 elseif
sacral_slope>=36.7678 then node 7 else Hernia
 5  if sacral_slope<27.9715 then node 8 elseif
sacral_slope>=27.9715 then node 9 else Normal
 6  class = Hernia
 7  if degree_spondylolisthesis<6.96053 then node
10 elseif degree_spondylolisthesis>=6.96053 then
node 11 else Normal
 8  if pelvic_tilt<17.2041 then node 12 elseif
pelvic_tilt>=17.2041 then node 13 else Hernia
 9  if degree_spondylolisthesis<5.38794 then node
14 elseif degree_spondylolisthesis>=5.38794 then
node 15 else Normal
10  if degree_spondylolisthesis<-0.533094 then node
16 elseif degree_spondylolisthesis>=-0.533094 then
node 17 else Normal
11  class = Spondylolisthesis
12  class = Normal
13  class = Hernia
14  class = Normal
15  if lumbar_lordosis_angle<38.9163 then node 18
elseif lumbar_lordosis_angle>=38.9163 then node 19
else Normal
16  class = Hernia
17  class = Normal
18  class = Hernia
19  class = Normal


Decision tree for classification
```

```
 1  if degree_spondylolisthesis<20.085 then node 2
elseif degree_spondylolisthesis>=20.085 then node 3
else Spondylolisthesis
 2  if pelvic_radius<117.436 then node 4 elseif
pelvic_radius>=117.436 then node 5 else Normal
 3  class = Spondylolisthesis
 4  if sacral_slope<36.7678 then node 6 elseif
sacral_slope>=36.7678 then node 7 else Hernia
 5  if sacral_slope<27.9715 then node 8 elseif
sacral_slope>=27.9715 then node 9 else Normal
 6  class = Hernia
 7  if sacral_slope<43.7154 then node 10 elseif
sacral_slope>=43.7154 then node 11 else Normal
 8  class = Hernia
 9  class = Normal
10  class = Hernia
11  class = Normal
```

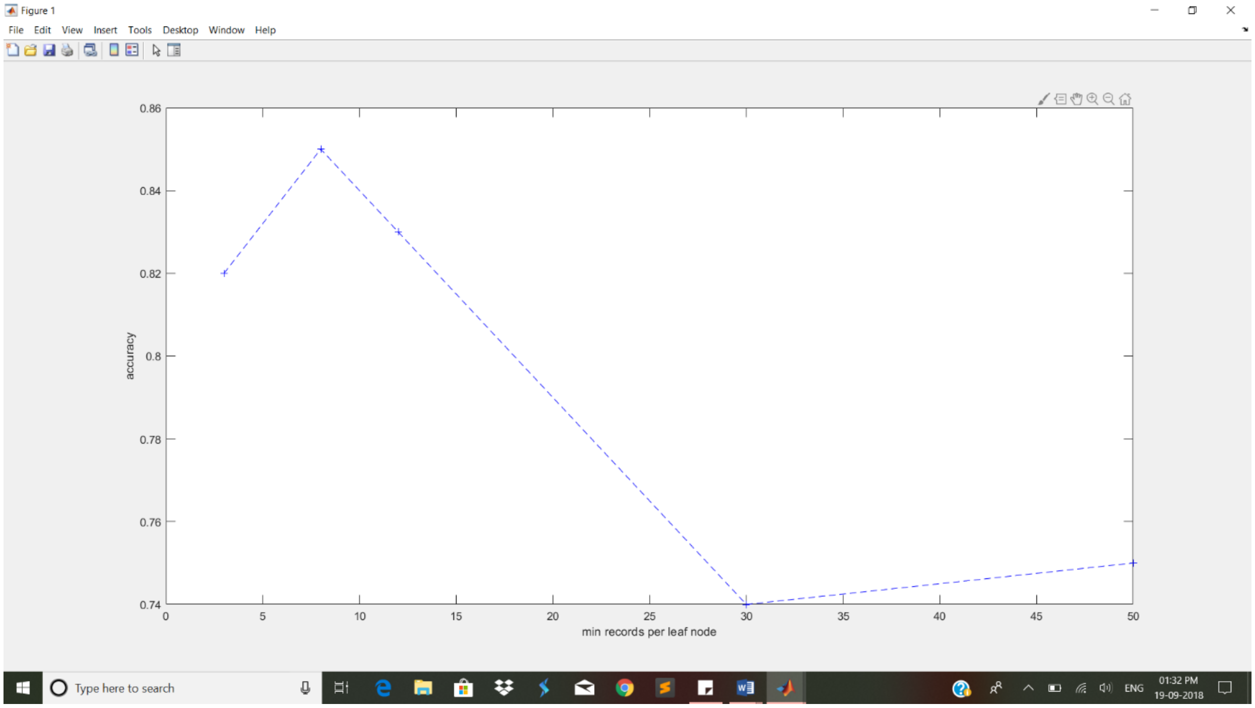
```
Decision tree for classification
1  if degree_spondylolisthesis<20.085 then node 2
elseif degree_spondylolisthesis>=20.085 then node 3
else Spondylolisthesis
2  if pelvic_radius<117.436 then node 4 elseif
pelvic_radius>=117.436 then node 5 else Normal
3  class = Spondylolisthesis
4  if sacral_slope<36.7678 then node 6 elseif
sacral_slope>=36.7678 then node 7 else Hernia
5  if sacral_slope<27.9715 then node 8 elseif
sacral_slope>=27.9715 then node 9 else Normal
6  class = Hernia
7  class = Normal
8  class = Hernia
9  class = Normal
```

```
Decision tree for classification
```

```
1  if degree_spondylolisthesis<20.085 then node 2
elseif degree_spondylolisthesis>=20.085 then node 3
else Spondylolisthesis
2  if pelvic_radius<117.436 then node 4 elseif
pelvic_radius>=117.436 then node 5 else Normal
3  class = Spondylolisthesis
4  class = Hernia
5  class = Normal


Decision tree for classification
1  if degree_spondylolisthesis<20.085 then node 2
elseif degree_spondylolisthesis>=20.085 then node 3
else Spondylolisthesis
2  if pelvic_radius<120.572 then node 4 elseif
pelvic_radius>=120.572 then node 5 else Normal
3  class = Spondylolisthesis
4  class = Hernia
5  class = Normal

decision tree min records per leaf node is 3
accuracy: 0.82
precision for classes is respectively
    0.7500    0.6923    0.9556

recall for all classes is respectively
    0.5000    0.8710    0.9556

decision tree min records per leaf node is 8
accuracy: 0.85
precision for classes is respectively
    0.7727    0.7222    1.0000

recall for all classes is respectively
    0.7083    0.8387    0.9333

decision tree min records per leaf node is 12
```

```
accuracy: 0.83
precision for classes is respectively
    0.7500    0.6842    1.0000

recall for all classes is respectively
    0.6250    0.8387    0.9333

decision tree min records per leaf node is 30
accuracy: 0.74
precision for classes is respectively
    0.5000    0.5789    1.0000

recall for all classes is respectively
    0.4167    0.7097    0.9333

decision tree min records per leaf node is 50
accuracy: 0.75
precision for classes is respectively
    0.7500    0.6923    0.9556

recall for all classes is respectively
    0.5833    0.6129    0.9333
```

Figure : accuracy plotted against min records per leaf node

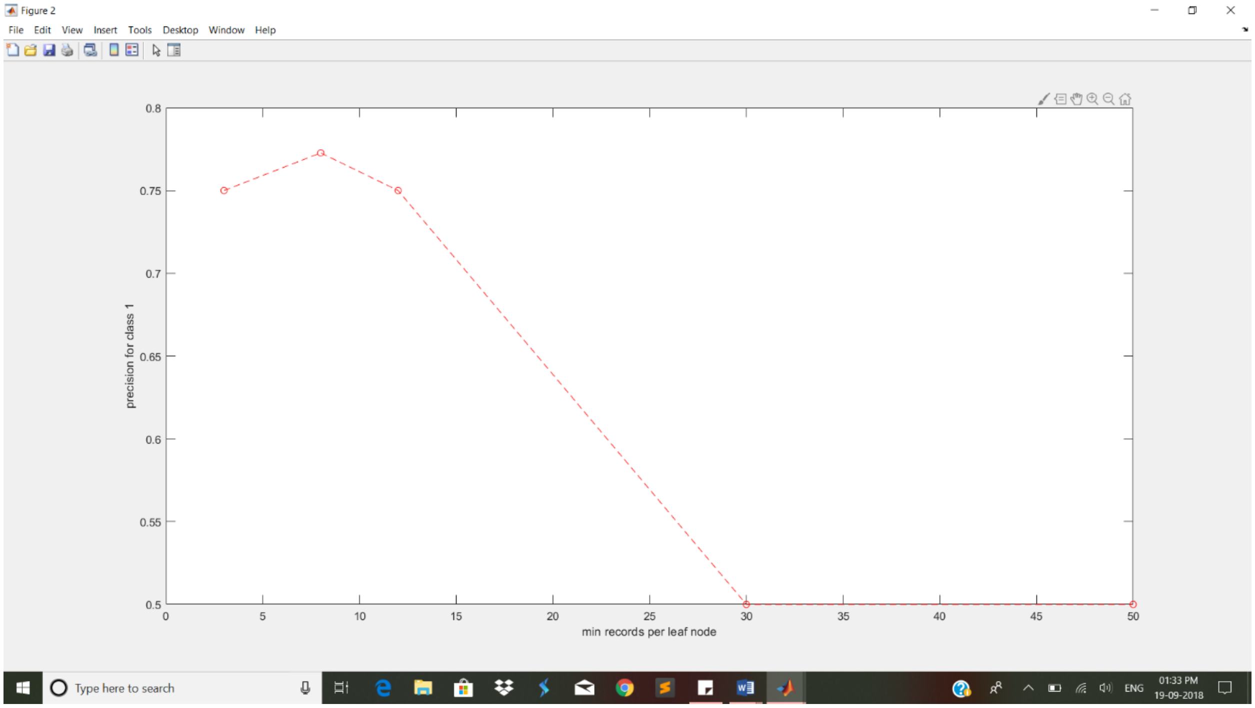


- Accuracy is defined for the whole classifier instead of the individual classes.
- From the above graph, we can conclude that the accuracy is highest when the minimum number o records per leaf node is 8. It’s value is 0.85

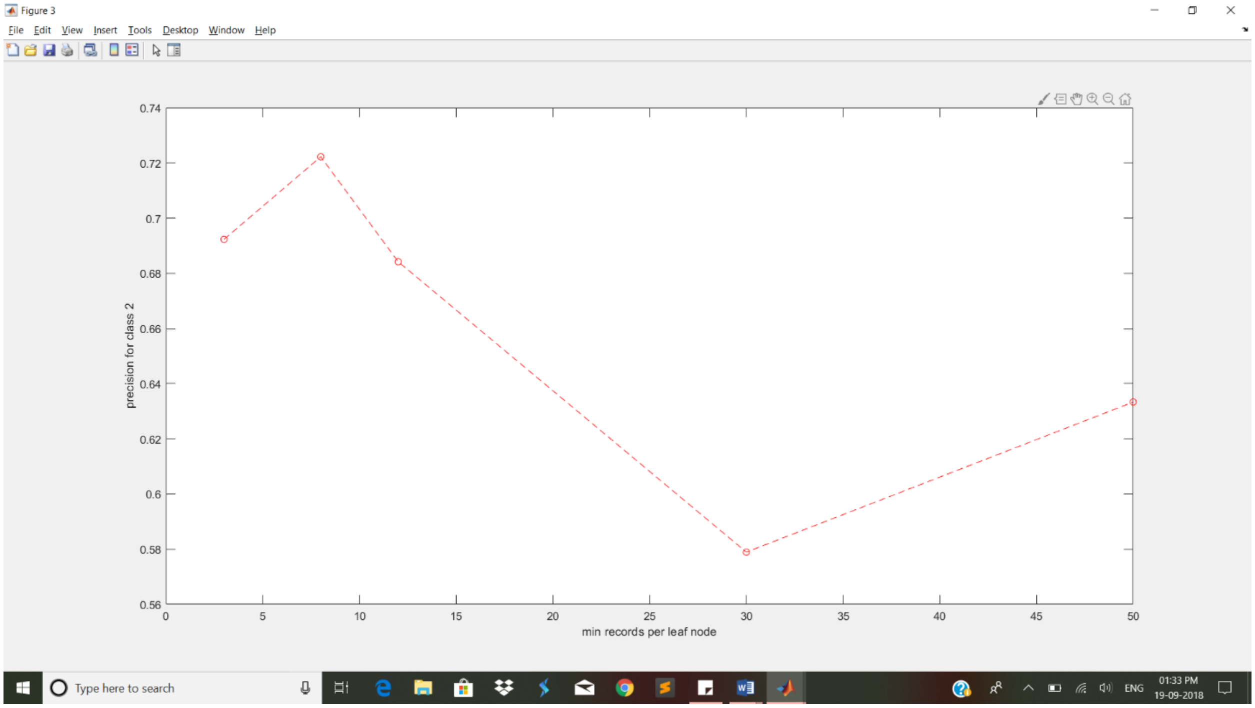
Figure:

a) Precsion for class 1

- The precision increases and then decreases. The reasons are same as we move generalization to optimal case to specification.
- It further remains constant when the number of records per leaf node increases



b) Precision for class 2



c)Precsion for class 3

In the precision for class 2 and 3

- The precision increases and then decreases, then further increases.
- This variation is due to a fewer instances of class B and C. thus, they are not properly bifurcated into divisions.

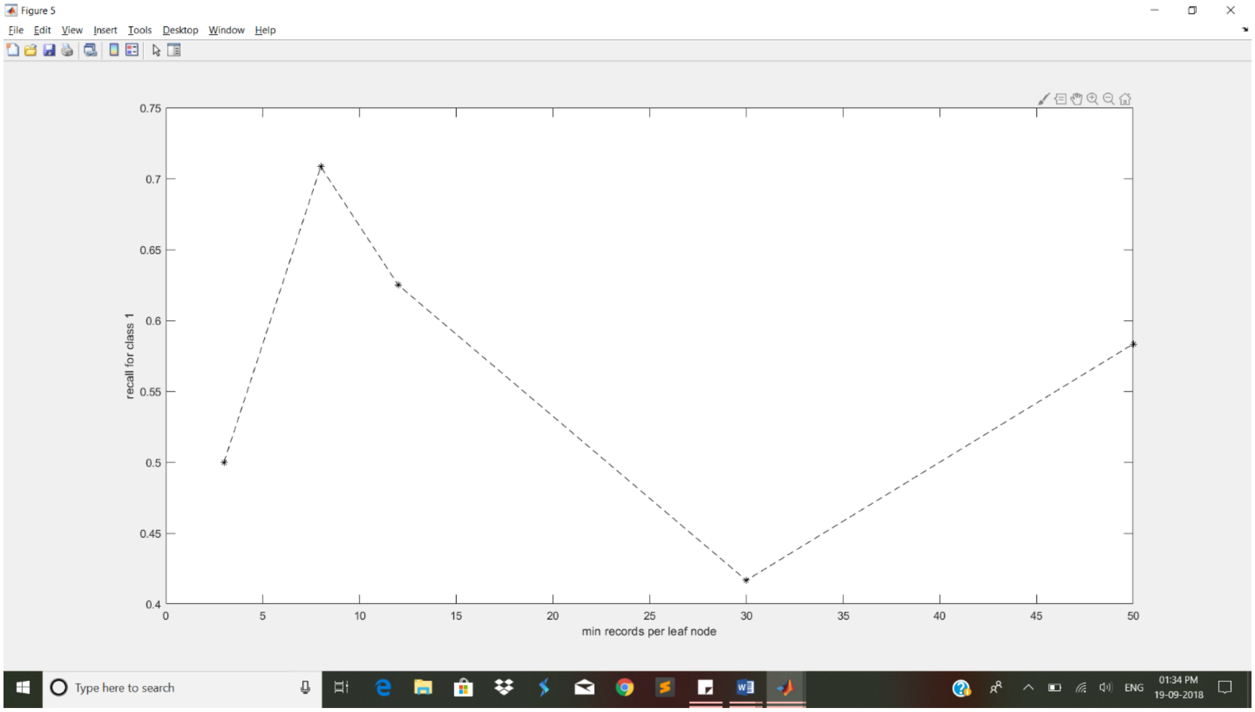
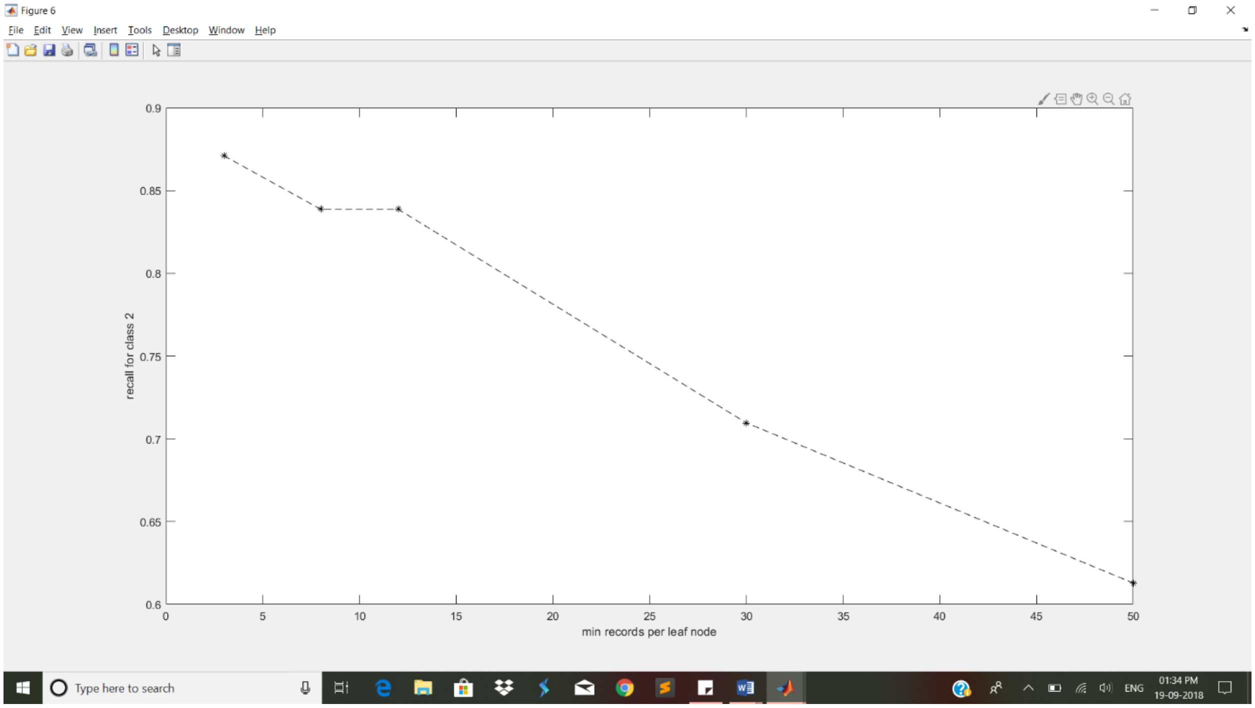
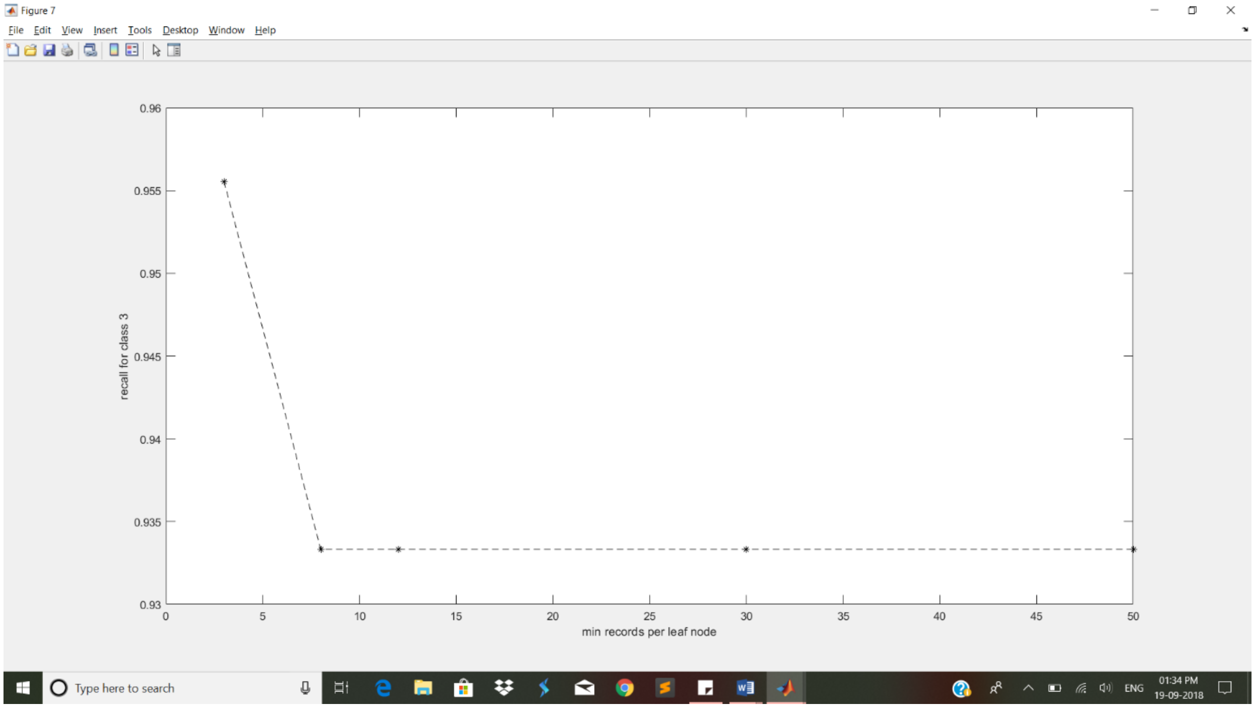


figure a) recall for class A

b) Recall for class 2



c) Recall for class 3



In the recall for classes there are more variations. The recall increases, reaches a maximum and either decreases or remains constant.

From all the above, I can conclude that the decision tree with minimum number of records per leaf node as 8 has the highest performance. It is the highest in accuracy and precision. Overall it is the best.

Answer 2)C)

Comparison for part 1)a) and 2)a)

- The size of the decision tree varies a lot.
- There are more splits to reach the end in answer)1)A than Answer 2)B.
- This is because there are 2 classes in Answer 1 and answer 3 has 3 classes. Three classes have more varied separations in the data set 3. That is the reason we obtain result in minimum number of steps
- This is helpful because each iteration of the dataset through the decision tree will require lesser time.

Comparison for part 1)b) and 2)b)

On comparing the b part, the conclusion is that the minimum number of records per leaf node as 8 has the highest performance in answer 2) and minimum number of records per leaf node as 12 has the highest performance in answer 1)

This is because of the **more the number of classes and the dataset belonging to it, It creates a balanced tree.** There will be no biasing towards the class having the majority datasets

Question 3) Take Data2 for this question. Partition each column into four sets of equal width of values. Assign these intervals as values 0, 1, 2, and 3 and replace each value in the original data by its corresponding interval number.
a. Show the boundaries for each interval for each attribute.
b. Learn a decision tree with this transformed data and compute performance parameters
in the same way as done for 1b and 2b.
c. Compare the performance metric as obtained in 1b with those obtained here in 3b.
Explain the differences in performance and give your intuitive reasons why these differences are observed.

<u>**Answer 3)A)**</u>

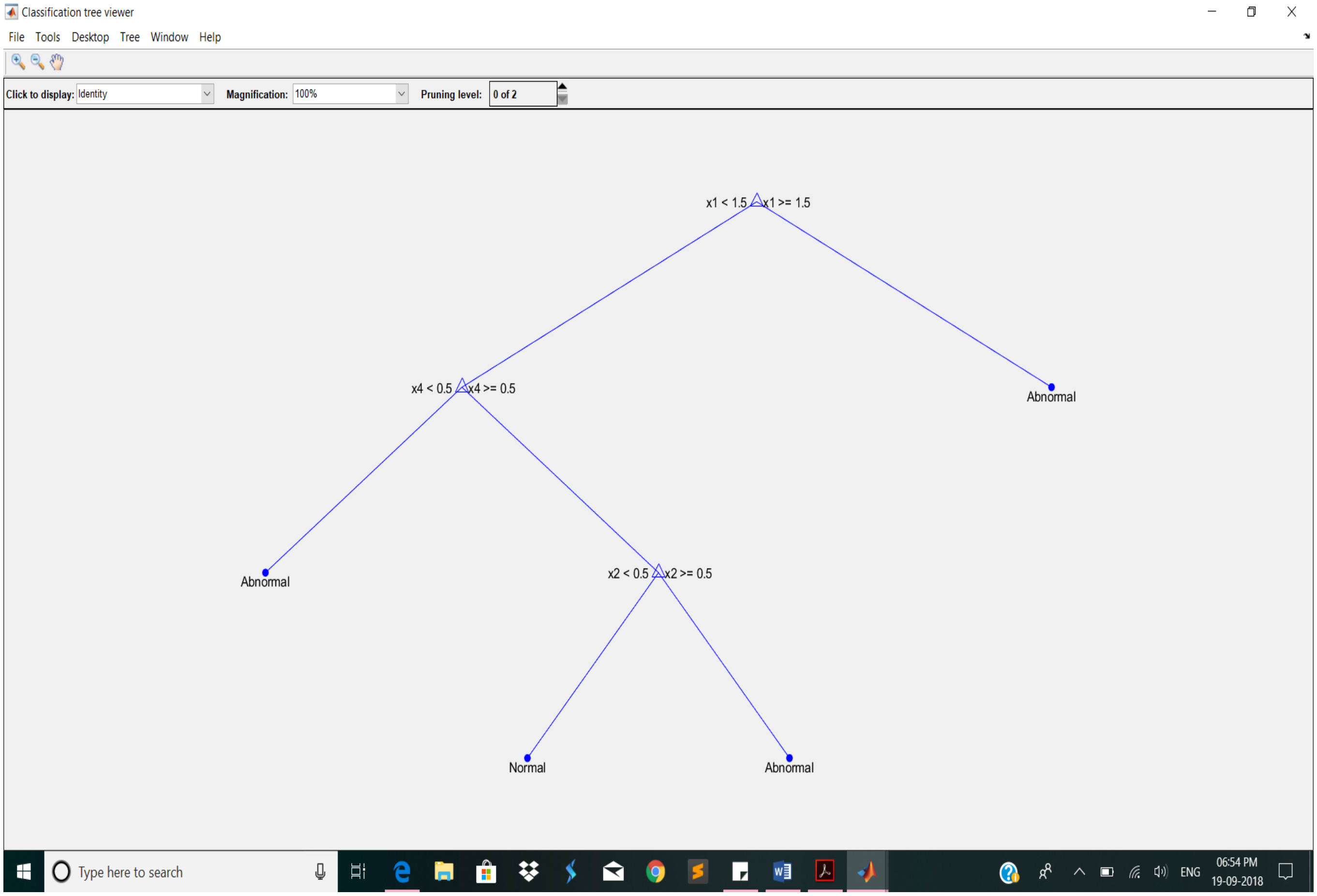
This output is generated in **the Commmand Window on running the program**

```
>> Answer3
we calculate minimum and maximum values to
calculate the range, hence, calculate boundaries
boundary for attribute 1
26.1479 to129.834
233.5202to337.2063
233.5202to337.2063
```

```
337.2063to440.8924
boundary for attribute 2
-6.5549 to49.4319
105.4187to161.4055
105.4187to161.4055
161.4055to217.3923
boundary for attribute 3
14 to125.7424
237.4848to349.2272
237.4848to349.2272
349.2272to460.9695
boundary for attribute 4
13.3669 to121.4296
229.4922to337.5548
229.4922to337.5548
337.5548to445.6175
boundary for attribute 5
70.0826 to163.071
256.0595to349.048
256.0595to349.048
349.048to442.0364
boundary for attribute 6
-11.0582 to418.5431
848.1443to1277.7456
848.1443to1277.7456
1277.7456to1707.3469
```

Answer 3)b)

Let us set the minimum records per leaf node as 11



Click to display: Identity

Magnification: 100%

Pruning level: 0 of 2

The performance measures are as follows

Output in Command Window :

```
Accuracy: 0.6422

Precision: 0.7885

Recall: 0.6845
```

Code in MATLAB for answer 3

```
[data,labels] = xlsread('s2.csv');
[m,n] = size(data);
inp = zeros(m,n);
 disp("we calculate minimum and maximum values to calculate
the range, hence, calculate boundaries");
for c = 1:n-1
    maximum = max(data(:,c));
    minimum = min(data(:,c));
    range = (maximum - minimum);
    partition = range/4;

    disp("boundary for attribute " + c);
    disp(minimum + " to" + (minimum + range));
    disp((minimum + (2*range)) +"to"+ (minimum +
(3*range)));
    disp((minimum + (2*range)) +"to"+ (minimum +
(3*range)));
    disp((minimum + (3*range)) +"to"+ (minimum +
(4*range)));

    for r = 1:m
        if(data(r,c) <= partition)
            inp(r,c) = 0;
        elseif(((data(r,c)) > (partition)) && (data(r,c) <=
(2*partition)) )
               inp(r,c) = 1;
        elseif(((data(r,c)) > (2*partition)) && (data(r,c)
<= (3*partition)) )
               inp(r,c) = 2;
         elseif(((data(r,c)) > (3*partition)) && (data(r,c)
<= (4*partition)) )
```

```
               inp(r,c) = 3;
        end
    end

end

leafSize = 11;
randVar = randperm(310);
Ind = randVar > 210;
Xtest = inp(Ind,:);
Xtrain = inp(~Ind,:);
ytest = labels(Ind,:);
ytrain = labels(~Ind,:);
Tree1 = fitctree(Xtrain,ytrain,'MinLeafSize',leafSize);
view(Tree1,'mode','graph');
pr = predict(Tree1,Xtest);
x = confusionmat(ytest,pr);
tp = x(1,1); %true positive
fn = x(1,2); %false negative
fp = x(2,1); %false positive
tn = x(2,2); %true negative
p = tp/(tp + fp); %positive precision value

r = tp/(tp + fn); %positive recall value
pn = tn/(tn + fn); %negative precision value
rn = tn/(tn + fp); %negative recall value
a = (tp + tn)/ (tp + tn + fp + fn);%accuracy
pf = (p + pn)/2; %mean precision
rf = (r + rn)/2; %mean recall

disp(“Accuracy: ” +a);
disp(“precision: ” +pf);
disp(“recall: ” +rf);
```

Answer 3) C)

	Ans 3)b)	**Ans 1)b)**
Accuracy	0.6422	0.80749
Precision	0.7885	0.82857
Recall	0.6845	0.84

I have compared the values in answer 1)b) and Answer 3)b). I have compared the values with the best case chosen in answer 1) (min number od records per leaf node as 12)

We can see here that the performance measures in the Answer 3) are quite lesser than the ones in answer 1). This is because the error rate increases as we have made assumptions in the answer 3. We have set the intervals which approximates certain values, thus, reduces the efficiency and performance