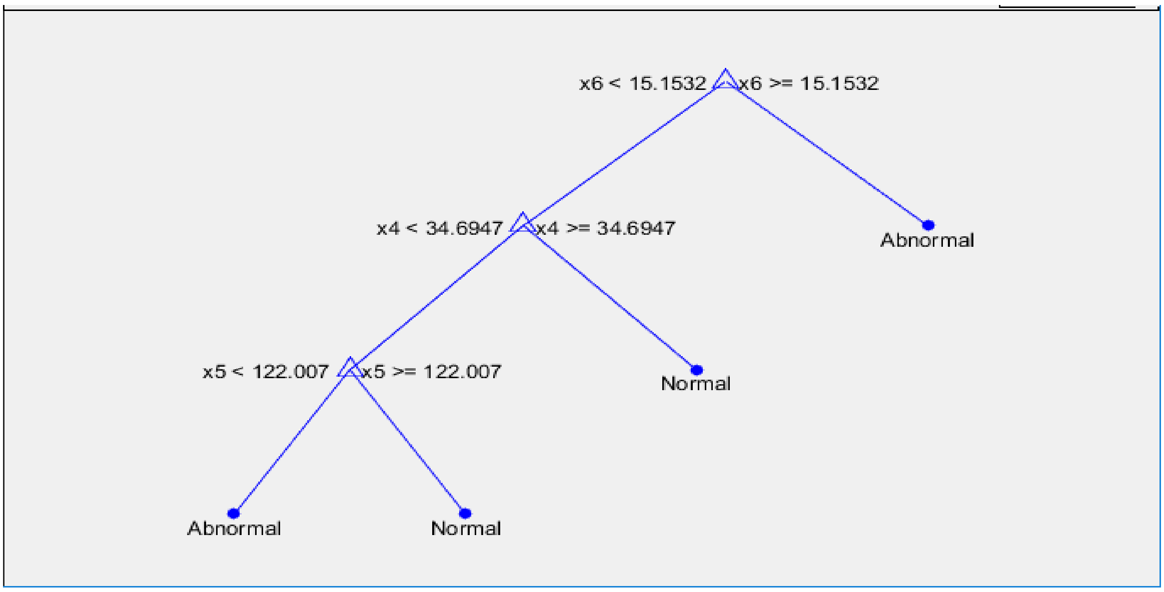
1) **Take Data2 and split it into randomly selected 210 training instances and remaining 100 as test instance. Create decision trees using the training set and the "minimum records per leaf node" values of 5, 10, 15, 20, and 25**.

```
clear ; close all; clc
filename = 'Biomechanical_Data_column_2C_weka.csv';
formatSpec = '%f%f%f%f%f%f%C';
Data2 = readtable(filename,'Delimiter',',', ...
    'Format',formatSpec);
fprintf('Decision tree For Data2\n\n');
%Take Data2 and split it into randomly selected 210 training instances and
remaining 100 as test instance
[trainInd,valInd,testInd] = dividerand(310,210,0,100);
train_Data2 = Data2(trainInd,:);
test_Data2  = Data2(testInd,:);
pred = train_Data2(:,1:6);
pred = table2array(pred);
label = train_Data2(:,7);
label = table2array(label);

%create decision trees using the training set and the "minimum records per
leaf node" values of 5, 10, 15, 20, and 25.
tree_Node5  = fitctree(pred,label,'MinLeafSize',5);
tree_Node10= fitctree(pred,label,'MinLeafSize',10);
tree_Node15= fitctree(pred,label,'MinLeafSize',15);
tree_Node20= fitctree(pred,label,'MinLeafSize',20);
tree_Node25= fitctree(pred,label,'MinLeafSize',25);
```

**1a) Show the tree for the value 25. Comment on what you notice about the five trees.**

```
%Show the tree for the value 25. Comment on what you notice about the five
trees.
view(tree_Node25,'Mode','graph')
test_pred = test_Data2(:,1:6);
test_pred = table2array(test_pred);
```




Number of leaf node decreases as minimum records per leaf increases, i.e.
For, minimum records per leaf = 25, number of node are 4 as seen in above figure, the number of attribute involved for making decision will be less and tree depth will be less
Whereas, for minimum records per leaf = 5, number of node would be 13 as seen in above figure. Hence, number of attribute involved for making decision will be more and tree depth will be more.

1b)

**For each tree compute and report the accuracy, precision, and recall values:**

```
p = length(test_label(idx1));
n = length(test_label(idx2));
N = p+n;
tp = sum(test_label(idx1)==test_predict(idx1));
tn = sum(test_label(idx2)==test_predict(idx2));
fp = n-tn;
fn = p-tp;
accuracy = (tp+tn)/N;
acc(i) = accuracy;
precision = tp/(tp+fp);
prec(i) = precision;
recall = tp/p;
rec(i) = recall;
```

**OUTPUT:**

For Decision tree with minimum records per leaf = 5
accuracy = 0.830000.... precision = 0.800000.... recall = 0.625000

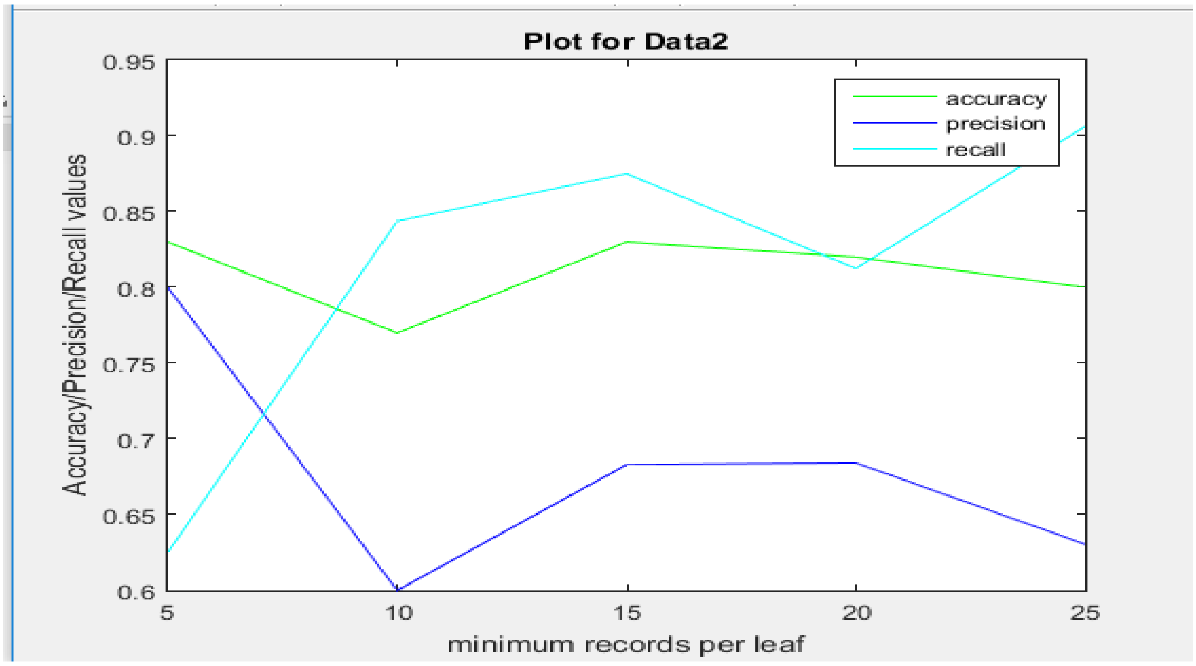
For Decision tree with minimum records per leaf = 10
accuracy = 0.770000.... precision = 0.600000.... recall = 0.843750

For Decision tree with minimum records per leaf = 15
accuracy = 0.830000.... precision = 0.682927.... recall = 0.875000

For Decision tree with minimum records per leaf = 20
accuracy = 0.820000.... precision = 0.684211.... recall = 0.812500

For Decision tree with minimum records per leaf = 25
accuracy = 0.800000.... precision = 0.630435.... recall = 0.906250

**Comment on the comparison of these values and show these values on a plot.**




Accuracy, Precision and recall vary w.r.t to minimum records per leaf, and it changes with each and every run.

**1c)**

**Now limit yourself to the case of 10 minimum records per leaf node. Repeat the tree learning exercise five times by randomly choosing different sets of 210 training instances. Report the accuracy, precision, and recall values for each run and also their averages and standard deviations. Comment on the variability of the values as the random sample changes**

**Accuracy, precision, and recall values for each run**

```
for i = 1:5
    fprintf('For %d random run\n',i);
    [trainInd,valInd,testInd] = dividerand(310,210,0,100);
    train_Data2 = Data2(trainInd,:);
    test_Data2  = Data2(testInd,:);
    pred = train_Data2(:,1:6);
    pred = table2array(pred);
    label = train_Data2(:,7);
    label = table2array(label);
    tree_Node10= fitctree(pred,label,'MinLeafSize',10);

    test_pred = test_Data2(:,1:6);
    test_pred = table2array(test_pred);
    test_predict = predict(tree_Node10,test_pred);
    test_label = test_Data2(:,7);
    test_label = table2array(test_label);

    idx1 = (test_label == 'Normal');
    idx2 = (test_label == 'Abnormal');
    p = length(test_label(idx1));
    n = length(test_label(idx2));
    N = p+n;
    tp = sum(test_label(idx1)==test_predict(idx1));
    tn = sum(test_label(idx2)==test_predict(idx2));
    fp = n-tn;
    fn = p-tp;
    accuracy = (tp+tn)/N;
    acc(i) = accuracy;
    precision = tp/(tp+fp);
    prec(i) = precision;
    recall = tp/p;
    rec(i) = recall;
    fprintf('accuracy =  %f \t precision = %f \t recall =
%f\n\n',accuracy,precision,recall);
end
```

**OUTPUT:**

For 1 random run

accuracy =  0.810000 precision = 0.621622 recall = 0.821429

For 2 random run

accuracy =  0.810000 precision = 0.615385 recall = 0.857143

For 3 random run

accuracy =  0.810000 precision = 0.655172 recall = 0.678571

For 4 random run

accuracy =  0.830000 precision = 0.725000 recall = 0.828571

For 5 random run

accuracy =  0.790000 precision = 0.708333 recall = 0.548387

**Averages and standard deviations**

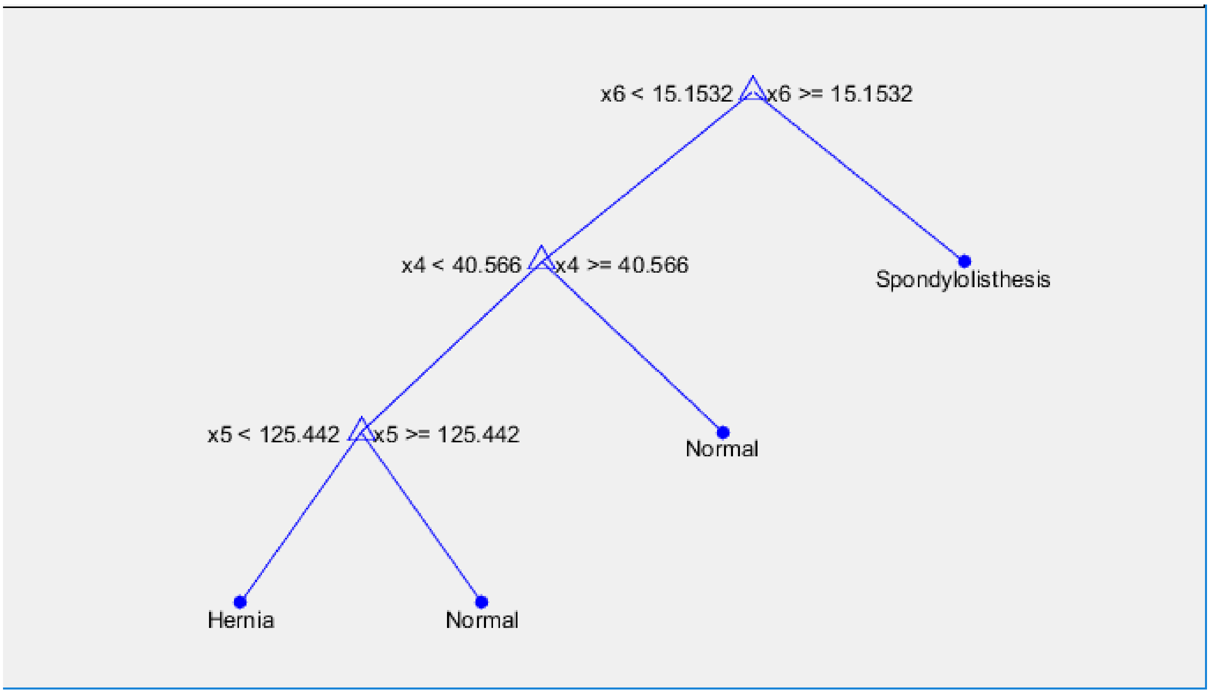
Average and standard Daviation calculation
average of accuracy = 0.810000 average of precision = 0.665102 average of recall = 0.548387
std of accuracy = 0.014142 std of precision = 0.049794 std ofrecall = 0.130819

**Comment on the variability of the values as the random sample changes**

The accuracy remains same for few runs and varies by small value for other runs. As can be seen from standard deviation, it is very less for accuracy and precision.

**2) Repeat the same tasks as done in Question-1 above for Data3.**

**2a) Show the tree for the value 25. Comment on what you notice about the five trees.**




Number of leaf node decreases as minimum records per leaf increases, i.e.
For, minimum records per leaf = 25, number of node are 4 as seen in above figure, the number of attribute involved for making decision will be less.
Whereas for minimum records per leaf = 5, number of node would be 9 as seen in above figure, hence number of attribute involved for making decision will be more.

**2b) For each tree compute and report the accuracy, precision, and recall values:**

```
fprintf('For %d random run\n',i);
[trainInd,valInd,testInd] = dividerand(310,210,0,100);
train_Data3 = Data3(trainInd,:);
test_Data3  = Data3(testInd,:);
pred = train_Data3(:,1:6);
pred = table2array(pred);
label = train_Data3(:,7);
label = table2array(label);
tree_Node10= fitctree(pred,label,'MinLeafSize',10);

test_pred = test_Data3(:,1:6);
test_pred = table2array(test_pred);
test_predict = predict(tree_Node10,test_pred);
test_label = test_Data3(:,7);
test_label = table2array(test_label);

idx1 = (test_label == 'Hernia');
idx2 = (test_label == 'Spondylolisthesis');
idx3 = (test_label == 'Normal');

H = length(test_label(idx1));
S = length(test_label(idx2));
Nr = length(test_label(idx3));
N = H + S + Nr;
th = sum(test_label(idx1)==test_predict(idx1));
ts = sum(test_label(idx2)==test_predict(idx2));
tn = sum(test_label(idx3)==test_predict(idx3));
fpH = S + Nr - ts - tn;
fpS = H + Nr - th - tn;
fpN = S + H - ts - th;
accuracy = (th + ts + tn)/ N;
acc(i) = accuracy;
precision_Hernia = th/(th+fpH);
prec(i)= precision_Hernia;
precision_Spondy = ts/(ts+fpS);
prec_S(i) = precision_Spondy;
precision_Normal = tn/(tn+fpN);
prec_N(i) = precision_Normal;
recall_Hernia = th/H;
rec(i)= recall_Hernia;
recall_Spondy = ts/S;
rec_S(i) = recall_Spondy;
recall_Normal = tn/Nr;
rec_N(i) = recall_Normal;

fprintf('For Hernia class,            accuracy =  %f \t precision = %f
\t recall = %f\n',accuracy,precision_Hernia,recall_Hernia);
fprintf('For Spondylolistesis class, accuracy =  %f \t precision = %f
\t recall = %f\n',accuracy,precision_Spondy,recall_Spondy);
fprintf('For Normal class,            accuracy =  %f \t precision = %f
\t recall = %f\n\n',accuracy,precision_Normal,recall_Normal);
```

**OUTPUT:**

For Decision tree with minimum records per leaf = 5

For Hernia class, accuracy = 0.820000 precision = 0.523810 recall = 0.578947

For Spondylolistesis class, accuracy = 0.820000 precision = 0.746269 recall = 0.980392

For Normal class, accuracy = 0.820000 precision = 0.700000 recall = 0.700000

For Decision tree with minimum records per leaf = 10

For Hernia class, accuracy = 0.840000 precision = 0.578947 recall = 0.578947

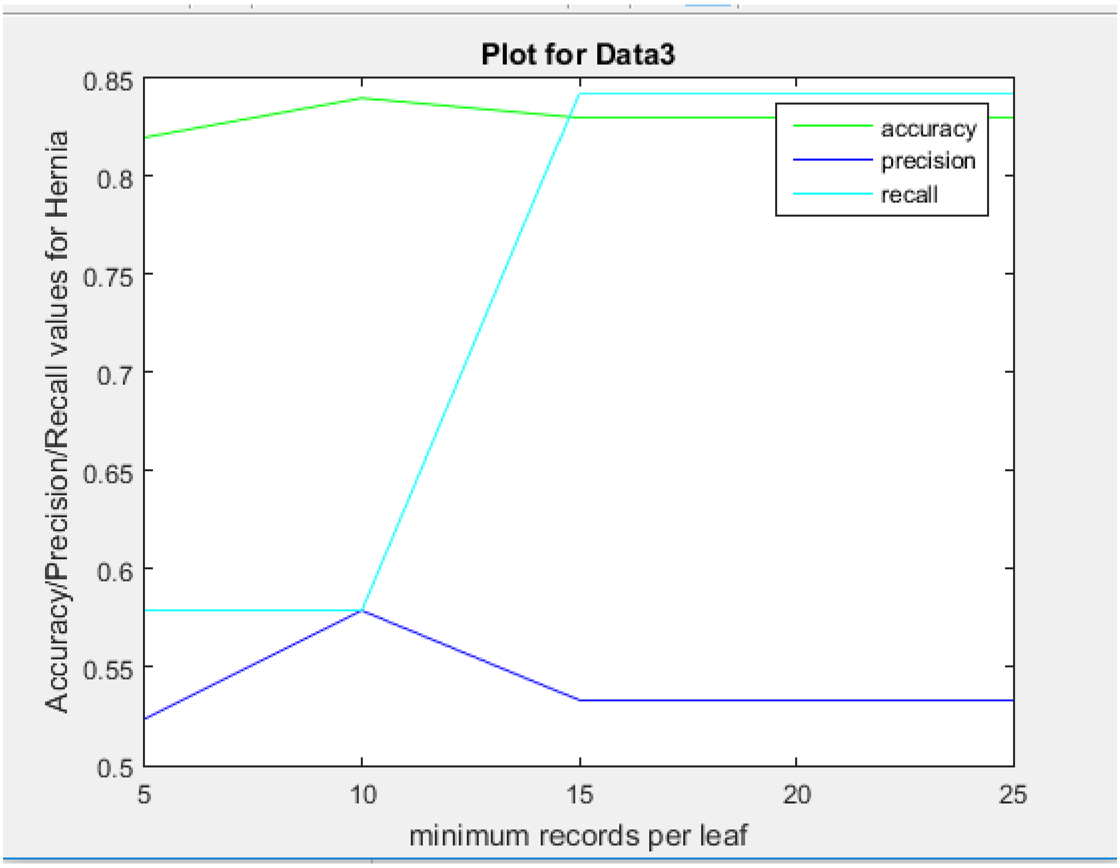
For Spondylolistesis class, accuracy = 0.840000 precision = 0.769231 recall = 0.980392

For Normal class, accuracy = 0.840000 precision = 0.718750 recall = 0.766667

For Decision tree with minimum records per leaf = 15
For Hernia class, accuracy = 0.830000 precision = 0.533333 recall = 0.842105
For Spondylolistesis class, accuracy = 0.830000 precision = 0.757576 recall = 0.980392
For Normal class, accuracy = 0.830000 precision = 0.809524 recall = 0.566667

For Decision tree with minimum records per leaf = 20
For Hernia class, accuracy = 0.830000 precision = 0.533333 recall = 0.842105
For Spondylolistesis class, accuracy = 0.830000 precision = 0.757576 recall = 0.980392
For Normal class, accuracy = 0.830000 precision = 0.809524 recall = 0.566667

For Decision tree with minimum records per leaf = 25
For Hernia class, accuracy = 0.830000 precision = 0.533333 recall = 0.842105
For Spondylolistesis class, accuracy = 0.830000 precision = 0.757576 recall = 0.980392
For Normal class, accuracy = 0.830000 precision = 0.809524 recall = 0.566667




Accuracy, Precision and recall vary w.r.t to minimum records per leaf, and it changes with each and every run. It depends on the random data selected.

**2c)**

**Now limit yourself to the case of 10 minimum records per leaf node. Repeat the tree learning exercise five times by randomly choosing different sets of 210 training instances. Report the accuracy, precision, and recall values for each run and also their averages and standard deviations. Comment on the variability of the values as the random sample changes**

**Accuracy, precision, and recall values for each run**

```
fprintf('For Decision tree with 10 minimum records per leaf\n');
for i = 1:5
    fprintf('For %d random run\n',i);
    [trainInd,valInd,testInd] = dividerand(310,210,0,100);
```

```
    train_Data3 = Data3(trainInd,:);
    test_Data3  = Data3(testInd,:);
    pred = train_Data3(:,1:6);
    pred = table2array(pred);
    label = train_Data3(:,7);
    label = table2array(label);
    tree_Node10= fitctree(pred,label,'MinLeafSize',10);

    test_pred = test_Data3(:,1:6);
    test_pred = table2array(test_pred);
    test_predict = predict(tree_Node10,test_pred);
    test_label = test_Data3(:,7);
    test_label = table2array(test_label);

    idx1 = (test_label == 'Hernia');
    idx2 = (test_label == 'Spondylolisthesis');
    idx3 = (test_label == 'Normal');

    H = length(test_label(idx1));
    S = length(test_label(idx2));
    Nr = length(test_label(idx3));
    N = H + S + Nr;
    th = sum(test_label(idx1)==test_predict(idx1));
    ts = sum(test_label(idx2)==test_predict(idx2));
    tn = sum(test_label(idx3)==test_predict(idx3));
    fpH = S + Nr - ts - tn;
    fpS = H + Nr - th - tn;
    fpN = S + H - ts - th;
    accuracy = (th + ts + tn)/ N;
    acc(i) = accuracy;
    precision_Hernia = th/(th+fpH);
    prec(i)= precision_Hernia;
    precision_Spondy = ts/(ts+fpS);
    prec_S(i) = precision_Spondy;
    precision_Normal = tn/(tn+fpN);
    prec_N(i) = precision_Normal;

    recall_Hernia = th/H;
    rec(i)= recall_Hernia;
    recall_Spondy = ts/S;
    rec_S(i) = recall_Spondy;
    recall_Normal = tn/Nr;
    rec_N(i) = recall_Normal;

    fprintf('For Hernia class,            accuracy =  %f \t precision = %f
\t recall = %f\n',accuracy,precision_Hernia,recall_Hernia);
    fprintf('For Spondylolistesis class, accuracy =  %f \t precision = %f
\t recall = %f\n',accuracy,precision_Spondy,recall_Spondy);
    fprintf('For Normal class,            accuracy =  %f \t precision = %f
\t recall = %f\n\n',accuracy,precision_Normal,recall_Normal);
end
```

For Decision tree with 10 minimum records per leaf

For 1 random run

For Hernia class,          accuracy =  0.830000        precision = 0.545455      recall = 0.631579

For Spondylolistesis class, accuracy =  0.830000  precision = 0.785714      recall = 0.964912

For Normal class,           accuracy =  0.830000        precision = 0.640000      recall = 0.666667

For 2 random run

For Hernia class,          accuracy =  0.810000        precision = 0.478261      recall = 0.611111

For Spondylolistesis class, accuracy =  0.810000  precision = 0.725806      recall = 0.957447

For Normal class,           accuracy =  0.810000        precision = 0.735294      recall = 0.714286

For 3 random run
For Hernia class, accuracy = 0.800000 precision = 0.363636 recall = 0.571429
For Spondylolistesis class, accuracy = 0.800000 precision = 0.739130 recall = 0.962264
For Normal class, accuracy = 0.800000 precision = 0.724138 recall = 0.636364

For 4 random run
For Hernia class, accuracy = 0.850000 precision = 0.611111 recall = 0.578947
For Spondylolistesis class, accuracy = 0.850000 precision = 0.775862 recall = 0.957447
For Normal class, accuracy = 0.850000 precision = 0.743590 recall = 0.852941

For 5 random run
For Hernia class, accuracy = 0.840000 precision = 0.500000 recall = 0.812500
For Spondylolistesis class, accuracy = 0.840000 precision = 0.769231 recall = 0.980392
For Normal class, accuracy = 0.840000 precision = 0.840000 recall = 0.636364

Average and standard Daviation calculation(precision and Recall for Hernia class)
average of accuracy = 0.826000 average of precision = 0.499693 average of recall = 0.641113
std of accuracy = 0.020736 std of precision = 0.091486 std ofrecall = 0.098850

Average and standard Daviation calculation(precision and Recall for Spondylolisthesis class)
average of accuracy = 0.826000 average of precision = 0.759149 average of recall = 0.964492
std of accuracy = 0.020736 std of precision = 0.025491 std ofrecall = 0.009450

Average and standard Daviation calculation(precision and Recall for Normal class)
average of accuracy = 0.826000 average of precision = 0.736604 average of recall = 0.701324
std of accuracy = 0.020736 std of precision = 0.071114 std of recall = 0.090549

**The accuracy remains same for few runs and varies by small value for other runs. As can be seen from standard deviation, it is very less for accuracy and precision. The Precision and Recall is relatively higher for Spondylolisthesis class, which has 150 count out of 310 given records, followed by Normal class and Hernia class.**

**2d) comment on the comparison of results obtained for 1c and 2c. Give your analysis for the differences in results**

**For Binary class, there is high precision value as seen in 1c for Data2. Whereas, for three class classification in 2c, there is reduction in the precision values, Which can be seen from average values. Standard deviation for Data2 recall was 0.136504, which is relatively higher than 2c i.e. 0.090549.**
**Average Recall is higher for the class Spondylolisthesis which is more in count in given dataset Data3.**

**3) Partition each column into four sets of equal widths of values. Assign these intervals as values 0, 1, 2, and 3 and replace each value by its corresponding interval value.**

```
%Take Data2 for this question.
%Partition each column into four sets of equal widths of values. Assign
these intervals as values 0, 1, 2, and 3 and replace each value by its
corresponding interval value.

Data = table2array(Data2(:,1:6));
binSize = floor(size(Data,1)/4);
c = (0:binSize:308);
c(1) = 1;
% partitioning 1st coloumn
col_1 = sort(Data(:,1));
edges1 = col_1(c);
edges1(size(edges1,1)) = max(col_1);
col_1 = discretize(sort(Data(:,1)),edges1,[0,1,2,3]);
% partitioning 2nd coloumn
col_2 = sort(Data(:,2));
edges2 = col_2(c);
edges2(size(edges2,1)) = max(col_2);
col_2 = discretize(sort(Data(:,2)),edges2,[0,1,2,3]);
% partitioning 3rd coloumn
col_3 = sort(Data(:,3));
edges3 = col_3(c);
edges3(size(edges3,1)) = max(col_3);
col_3 = discretize(sort(Data(:,3)),edges3,[0,1,2,3]);
% partitioning 4th coloumn
col_4 = sort(Data(:,4));
edges4 = col_4(c);
edges4(size(edges4,1)) = max(col_4);
col_4 = discretize(sort(Data(:,4)),edges4,[0,1,2,3]);
% partitioning 5th coloumn
col_5 = sort(Data(:,5));
edges5 = col_5(c);
edges5(size(edges5,1)) = max(col_5);
col_5 = discretize(sort(Data(:,5)),edges5,[0,1,2,3]);
% partitioning 6th coloumn
col_6 = sort(Data(:,6));
edges6 = col_6(c);
edges6(size(edges6,1)) = max(col_6);
col_6 = discretize(sort(Data(:,6)),edges6,[0,1,2,3]);
```

**3a) Show the boundaries for each interval for each attribute**.

```
edges = [edges1 edges2 edges3 edges4 edges5 edges6];
Header = Data2.Properties.VariableNames;
for j = 1:6
    fprintf('Boundaries for - %s\n',Header{j});
    for i = 1:4
        fprintf('%f - %f : %d \n',edges(i,j),edges(i + 1,j),i-1);
    end
end
```

Boundaries for - pelvic_incidence
26.147921 - 46.390260 : 0
46.390260 - 58.521623 : 1
58.521623 - 72.560702 : 2
72.560702 - 129.834041 : 3
Boundaries for - pelvic_tilt Numeric
-6.554948 - 10.540675 : 0
10.540675 - 16.208839 : 1
16.208839 - 21.931147 : 2
21.931147 - 49.431864 : 3
Boundaries for - lumbar_lordosis_angle
14.000000 - 36.679985 : 0
36.679985 - 49.278597 : 1
49.278597 - 62.859109 : 2
62.859109 - 125.742386 : 3
Boundaries for - sacral_slope
13.366931 - 33.215251 : 0
33.215251 - 42.324573 : 1
42.324573 - 52.253195 : 2
52.253195 - 121.429566 : 3
Boundaries for - pelvic_radius
70.082575 - 110.703107 : 0
110.703107 - 118.151531 : 1
118.151531 - 125.391138 : 2
125.391138 - 163.071041 : 3
Boundaries for - degree_spondylolisthesis
-11.058179 - 1.571205 : 0
1.571205 - 11.211523 : 1
11.211523 - 40.510982 : 2
40.510982 - 418.543082 : 3

**3b) Learn a decision tree with this transformed data and compute performance parameters in the same way as done for 1c and 2c.**

```
fprintf('\nFor Decision tree with 10 minimum records per leaf\n');
for i = 1:5
    fprintf('For %d random run\n',i);
    [trainInd,valInd,testInd] = dividerand(310,210,0,100);
    train_Data2 = new_Data(trainInd,:);
    test_Data2  = new_Data(testInd,:);
    pred = train_Data2;
    train_label = label(trainInd,:);
    tree_Node10= fitctree(pred,train_label,'MinLeafSize',10);
```

```
    test_pred = test_Data2;
    test_predict = predict(tree_Node10,test_pred);
    test_label = label(testInd,:);

    idx1 = (test_label == 'Normal');
    idx2 = (test_label == 'Abnormal');
    p = length(test_label(idx1));
    n = length(test_label(idx2));
    N = p+n;
    tp = sum(test_label(idx1)==test_predict(idx1));
    tn = sum(test_label(idx2)==test_predict(idx2));
    fp = n-tn;
    fn = p-tp;
    accuracy = (tp+tn)/N;
    acc(i) = accuracy;
    precision = tp/(tp+fp);
    prec(i) = precision;
    recall = tp/p;
    rec(i) = recall;
    fprintf('accuracy =  %f \t precision = %f \t recall =
%f\n\n',acc,prec,rec);
end

fprintf('Average and standard Daviation calculation for transformed
data\n')
fprintf('average of accuracy =  %f \t average of precision = %f \t average
of recall = %f\n', mean(acc),mean(prec),mean(recall));
fprintf('std of accuracy =  %f \t      std of precision = %f \t      std
ofrecall = %f\n\n',std(acc),std(prec),std(rec));
```

**OUTPUT:**

For Decision tree with 10 minimum records per leaf

For 1 random run

accuracy = 0.960000 precision = 1.000000 recall = 0.875000

For 2 random run

accuracy = 0.960000 precision = 1.000000 recall = 0.857143

For 3 random run

accuracy = 0.920000 precision = 1.000000 recall = 0.741935

For 4 random run

accuracy = 0.950000 precision = 1.000000 recall = 0.843750

For 5 random run

accuracy = 0.940000 precision = 1.000000 recall = 0.800000

Average and standard Daviation calculation for transformed data

average of accuracy = 0.946000 average of precision = 1.000000 average of recall = 0.800000

std of accuracy = 0.016733 std of precision = 0.000000 std of recall = 0.053383

**3c) Compare these results with those obtained for 1c. Analyze the differences in performance and give your intuitive reasons why these differences are observed.**

**Accuracy and precision value are higher for transformed data than original data in 1c. Also, precision turn out to be 1 for all the random run for transformed data. This means the model doesn't predict any false positive values. This is because, in transformed data, the data is uniformly distributed for each attribute and has uniform values, hence the decision can be made using very few columns. Hence, generated decision will have higher accuracy and precision**