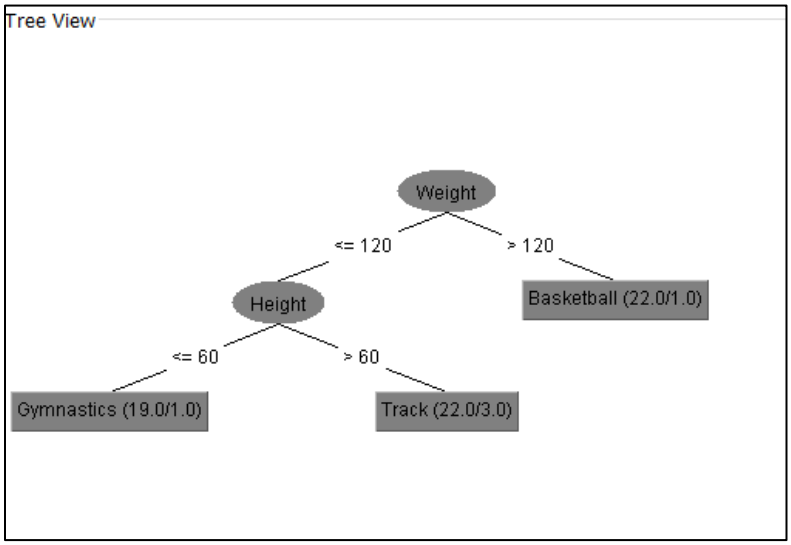
Homework #5
MEM 410, Managerial Analytics, Winter 2018
Due: Start of Class 3/1/18
Electronic Submission Only

1. [9 points] Download both the "Female Athletes" data and the "Female Athletes Test" data files from Canvas, and perform the following tasks using Weka
   a. Build a decision tree with the Female Athlete data. This is trying to predict which sport is played based on height and weight. Test your model with the test data. How do you compare the quality of the model when performed on the training sample versus the validation sample?



Predictions using training data

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          54               85.7143 %
Incorrectly Classified Instances         9               14.2857 %
Kappa statistic                          0.7857
Mean absolute error                      0.1338
Root mean squared error                  0.2987
Relative absolute error                 30.0744 %
Root relative squared error             63.3072 %
Total Number of Instances               63

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.952     0.071      0.87      0.952     0.909      0.931    Basketball
                 0.81      0.095      0.81      0.81      0.81       0.821    Track
                 0.81      0.048      0.895     0.81      0.85       0.888    Gymnastics
Weighted Avg.    0.857     0.071      0.858     0.857     0.856      0.88

=== Confusion Matrix ===

  a  b  c   <-- classified as
 20  1  0 |  a = Basketball
  2 17  2 |  b = Track
  1  3 17 |  c = Gymnastics
```

Predictions with test data :

```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          21               87.5    %
Incorrectly Classified Instances         3               12.5    %
Kappa statistic                          0.813
Mean absolute error                      0.1203
Root mean squared error                  0.2747
Relative absolute error                 27.0634 %
Root relative squared error             58.2707 %
Total Number of Instances               24

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.118      0.778     1         0.875      0.941    Basketball
                 0.75      0.063      0.857     0.75      0.8        0.781    Track
                 0.889     0          1         0.889     0.941      0.978    Gymnastics
Weighted Avg.    0.875     0.055      0.888     0.875     0.875      0.902

=== Confusion Matrix ===

 a b c   <-- classified as
 7 0 0 | a = Basketball
 2 6 0 | b = Track
 0 1 8 | c = Gymnastics
```

The correctly classified instance or the accuracy rate for training data is 85.7%. When tested against validation sample, model yields 87.5% accuracy which implies it is a good model. The model is expected to classify correctly and the model will not collapse with unknown data.

b. Now build a kNN model and compare these results to the decision tree. What's an optimal k in this case? How would you compare the two models?

For K=1,

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          55               87.3016 %
Incorrectly Classified Instances         8               12.6984 %
Kappa statistic                          0.8095
Mean absolute error                      0.1206
Root mean squared error                  0.3005
Relative absolute error                 27.1183 %
Root relative squared error             63.6954 %
Total Number of Instances               63

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.952     0.024      0.952     0.952     0.952      0.957    Basketball
                 0.81      0.095      0.81      0.81      0.81       0.827    Track
                 0.857     0.071      0.857     0.857     0.857      0.855    Gymnastics
Weighted Avg.    0.873     0.063      0.873     0.873     0.873      0.88

=== Confusion Matrix ===

  a  b  c   <-- classified as
 20  1  0 |  a = Basketball
  1 17  3 |  b = Track
  0  3 18 |  c = Gymnastics
```

For K=3,

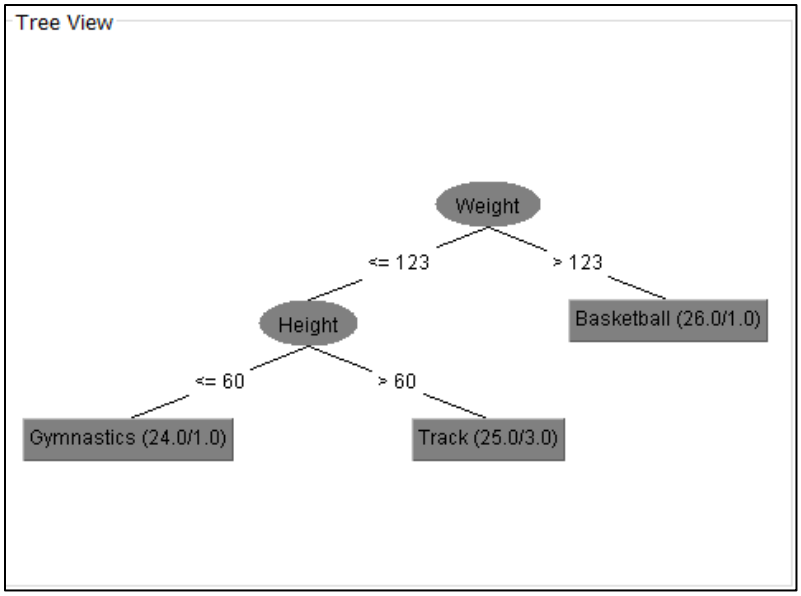
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          57               90.4762 %
Incorrectly Classified Instances         6                9.5238 %
Kappa statistic                          0.8571
Mean absolute error                      0.1064
Root mean squared error                  0.2645
Relative absolute error                 23.9172 %
Root relative squared error             56.0651 %
Total Number of Instances               63

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.952     0          1         0.952     0.976      0.976    Basketball
                 0.952     0.119      0.8       0.952     0.87       0.884    Track
                 0.81      0.024      0.944     0.81      0.872      0.858    Gymnastics
Weighted Avg.    0.905     0.048      0.915     0.905     0.906      0.906

=== Confusion Matrix ===

  a  b  c   <-- classified as
 20  1  0 |  a = Basketball
  0 20  1 |  b = Track
  0  4 17 |  c = Gymnastics
```

Since for K=3, Correctly classified instances are 90%, optimal value of K=3
Prediction on test data,

```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          24              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0.0774
Root mean squared error                  0.1541
Relative absolute error                 17.4202 %
Root relative squared error             32.6987 %
Total Number of Instances               24

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Basketball
                 1         0          1         1         1          1        Track
                 1         0          1         1         1          1        Gymnastics
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

 a b c   <-- classified as
 7 0 0 | a = Basketball
 0 8 0 | b = Track
 0 0 9 | c = Gymnastics
```

When KNN model is compared with decision tree, correctly classified instances for KNN is 90.47%. However, it is 100% for k-value of 3. To compare the two models, we use Correctly Classified instances which is higher for KNN. This implies KNN model will sustain with future data. Hence, it is a better model.

c. Now repeat the above with “Female Athletes 2” and “Female Athletes Test 2”. The difference between this version and the previous one is that half of the original “Female Athletes Test” data is now sitting in the “Female Athletes 2”. Compare the result against the original “Female Athletes” model. Do you think the second decision tree model is a better predictive model than the first decision tree model, why or why not?
Decision Tree on training set:



Predictions using training set.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          66               88      %
Incorrectly Classified Instances         9               12      %
Kappa statistic                          0.8199
Mean absolute error                      0.1125
Root mean squared error                  0.2728
Relative absolute error                 25.2788 %
Root relative squared error             57.7736 %
Total Number of Instances               75

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.96      0.04       0.923     0.96      0.941      0.957    Basketball
                 0.885     0.061      0.885     0.885     0.885      0.937    Gymnastics
                 0.792     0.078      0.826     0.792     0.809      0.833    Track
Weighted Avg.    0.88      0.06       0.879     0.88      0.879      0.91

=== Confusion Matrix ===

  a  b  c   <-- classified as
 24  0  1 |  a = Basketball
  0 23  3 |  b = Gymnastics
  2  3 19 |  c = Track
```

Predictions with test data,

```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          11               91.6667 %
Incorrectly Classified Instances         1                8.3333 %
Kappa statistic                          0.871
Mean absolute error                      0.0956
Root mean squared error                  0.2181
Relative absolute error                 21.4704 %
Root relative squared error             46.1761 %
Total Number of Instances               12

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Basketball
                 0.75      0          1         0.75      0.857      0.922    Gymnastics
                 1         0.143      0.833     1         0.909      0.929    Track
Weighted Avg.    0.917     0.06       0.931     0.917     0.915      0.944
```

The model on training data set has accuracy rate of 88% for the training set. When checked with testing data, the model yields accuracy of 91.6% which means that the model is a good fit.

KNN predictions using training data, K=3:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          69               92      %
Incorrectly Classified Instances         6                8      %
Kappa statistic                          0.88
Mean absolute error                      0.0909
Root mean squared error                  0.2399
Relative absolute error                 20.4226 %
Root relative squared error             50.8149 %
Total Number of Instances               75

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.96      0          1         0.96      0.98       0.99     Basketball
                 0.885     0.041      0.92      0.885     0.902      0.896    Gymnastics
                 0.917     0.078      0.846     0.917     0.88       0.91     Track
Weighted Avg.    0.92      0.039      0.923     0.92      0.921      0.932
```

KNN Predictions using test data

```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          12              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0.0789
Root mean squared error                  0.1583
Relative absolute error                 17.7347 %
Root relative squared error             33.5101 %
Total Number of Instances               12

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Basketball
                 1         0          1         1         1          1        Gymnastics
                 1         0          1         1         1          1        Track
Weighted Avg.    1         0          1         1         1          1
```

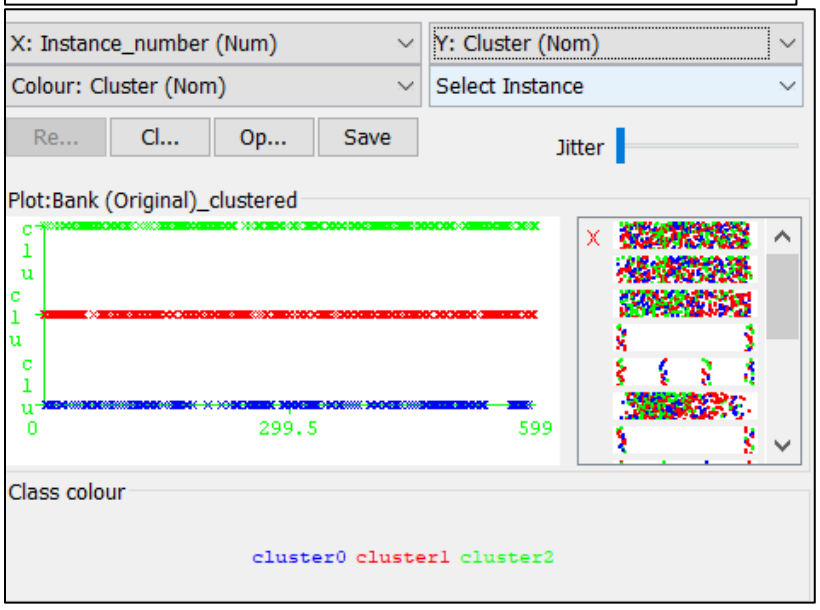
Comparing KNN model with decision tree for athlete 2 dataset, KNN model has better accuracy rate (92% vs 91.6%) and the accuracy rate of Test data set is 100%.

Comparing decision tree 1 model with decision tree 2 model, accuracy is higher for latter model and is a better fit. Hence, 2nd model is a better predictive model.

2. [9 points] Download both the “Bank (Original)” data and the “Bank (Modified)” data files from Canvas, and perform the following tasks using Weka
   a. Apply SimpleKMeans to cluster the data in Bank (Original) into 3, 4, and 6 clusters. How would you describe the difference among them using the “cluster centroid” views? If you try to visualize the cluster assignments, do you see the clusters as nicely defined as the centroids?

Cluster count : 3

```
Cluster centroids:
                          Cluster#
Attribute      Full Data          0          1          2
                   (600)      (154)      (251)      (195)
=============================================================
id               ID12101    ID12104    ID12102    ID12101
age                42.395    41.9545    46.5618    37.3795
sex                FEMALE     FEMALE     FEMALE       MALE
region         INNER_CITY       TOWN INNER_CITY INNER_CITY
income        27524.0312 27284.6914 31075.1603 23142.1078
married               YES         NO        YES        YES
children           1.0117     1.2987     1.0637     0.7179
car                   NO         NO        YES         NO
save_act             YES        YES        YES         NO
current_act          YES        YES        YES        YES
mortgage              NO         NO         NO         NO
pep                   NO         NO         NO        YES
```



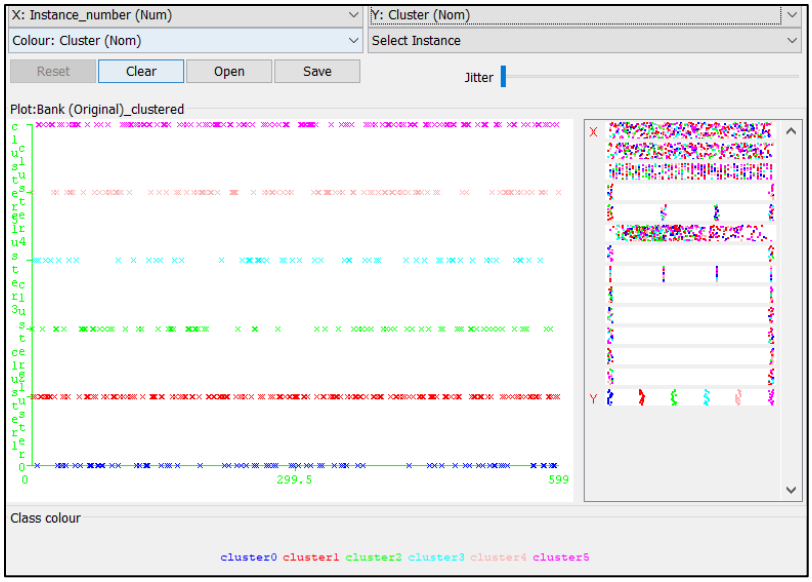
Cluster count : 4

Cluster centroids:

Attribute	Full Data (600)	Cluster# 0 (114)	1 (206)	2 (181)	3 (99)
id	ID12101	ID12107	ID12103	ID12101	ID12102
age	42.395	37.193	45.034	45.674	36.899
sex	FEMALE	MALE	FEMALE	MALE	FEMALE
region	INNER_CITY	TOWN	INNER_CITY	INNER_CITY	TOWN
income	27524.0312	23301.1535	29692.7667	31181.7101	21186.7452
married	YES	NO	YES	YES	YES
children	1.0117	1.1404	0.9515	1	1.0101
car	NO	NO	YES	NO	NO
save_act	YES	YES	YES	YES	NO
current_act	YES	YES	YES	YES	YES
mortgage	NO	NO	NO	NO	NO
pep	NO	NO	NO	YES	NO

Cluster centroids:

Attribute	Full Data (600)	Cluster# 0 (74)	1 (164)	2 (71)	3 (58)	4 (99)	5 (134)
id	ID12101	ID12107	ID12103	ID12101	ID12104	ID12102	ID12108
age	42.395	42.9324	43.7744	39.0282	37.3103	38.404	47.3433
sex	FEMALE	FEMALE	FEMALE	FEMALE	FEMALE	MALE	MALE
region	INNER_CITY	RURAL	INNER_CITY	INNER_CITY	TOWN	INNER_CITY	TOWN
income	27524.0312	28838.7605	28586.4063	20463.1273	20600.8528	25720.037	33568.3929
married	YES	NO	YES	YES	YES	YES	NO
children	1.0117	1.973	0.628	0.6901	1.6207	0.899	0.9403
car	NO	NO	NO	NO	NO	YES	YES
save_act	YES	YES	YES	NO	NO	NO	YES
current_act	YES	YES	YES	YES	YES	YES	YES
mortgage	NO	NO	NO	NO	NO	YES	NO
pep	NO	NO	NO	YES	NO	YES	YES



Clusters are difficult to read and comprehend when compared to Centroid view. Cluster centroids are the mean vectors for each cluster (so, each dimension value in the centroid represents the mean value for that dimension in the cluster). Thus, centroids can be used to characterize the clusters. For eg. Clusters 0 & 1 are female dominant cluster 2 has more males.
No, Clusters are not properly defined than the cluster centroid view.

b. Try to apply Association rule to the Bank (Original)? Why do you think you cannot perform this task?

Association rules works well with Categorical data. Since Bank(original)has fields such as age, income and children which are not categorically defined rather they are numeric in nature. Hence, Association rule cannot be applied.

c. Apply Association rule to the Bank (Modified) dataset. Make sure you remove necessary data columns so that you can apply Association rule. Submit the top 10 rules for each the following two scenarios separately, then discuss how valuable these insights are

i) Support >= 0.1 and Confidence >=0.85

```
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.1 (60 instances)
Minimum metric <confidence>: 0.85
Number of cycles performed: 18

Generated sets of large itemsets:

Size of set of large itemsets L(1): 24

Size of set of large itemsets L(2): 154

Size of set of large itemsets L(3): 221

Size of set of large itemsets L(4): 56

Size of set of large itemsets L(5): 1

Best rules found:

 1. region=INNER_CITY Age_Bucket=<30 75 ==> Income_Bucket=<20K 67    conf:(0.89)
 2. Age_Bucket=>60 90 ==> save_act=YES 77    conf:(0.86)
 3. sex=MALE Age_Bucket=<30 76 ==> Income_Bucket=<20K 65    conf:(0.86)
 4. sex=MALE Age_Bucket=46-60 75 ==> save_act=YES 64    conf:(0.85)
 5. sex=MALE Age_Bucket=46-60 75 ==> Income_Bucket=20K-50K 64    conf:(0.85)
 6. save_act=YES Age_Bucket=<30 88 ==> Income_Bucket=<20K 75    conf:(0.85)
```

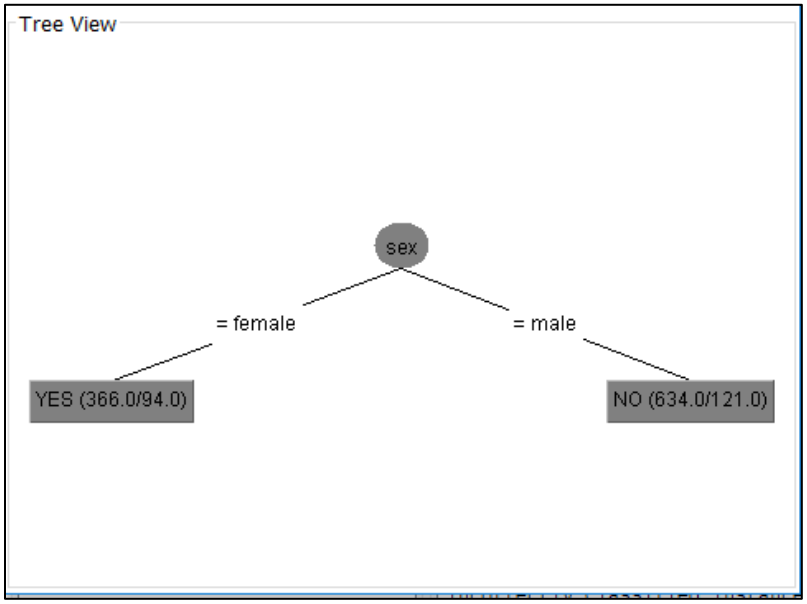
ii) Support >= 0.2 and Lift >= 1.25

```
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.2 (120 instances)
Minimum metric <lift>: 1.25
Number of cycles performed: 16

Generated sets of large itemsets:

Size of set of large itemsets L(1): 20

Size of set of large itemsets L(2): 59

Size of set of large itemsets L(3): 22

Best rules found:

 1. Age_Bucket=<30 144 ==> Income_Bucket=<20K 122    conf:(0.85) < lift:(2.58)> lev:(0.12) [74] conv:(4.21)
 2. Income_Bucket=<20K 197 ==> Age_Bucket=<30 122    conf:(0.62) < lift:(2.58)> lev:(0.12) [74] conv:(1.97)
 3. Age_Bucket=46-60 167 ==> Income_Bucket=20K-50K 133    conf:(0.8) < lift:(1.33)> lev:(0.06) [33] conv:(1.92)
 4. Income_Bucket=20K-50K 358 ==> Age_Bucket=46-60 133    conf:(0.37) < lift:(1.33)> lev:(0.06) [33] conv:(1.14)
```

When association rules are applied to the two tables, it is revealed that there is correlation between Age_buckt and Income_bucket but there is not enough evidence to prove causation. This implies the existing facts are strengthened but a hypothesis cannot be drawn from the relation between these variables.

3. [7 points] Download the “Titanic Data – Training” and the “Titanic Data – Test” datasets from Canvas. This is the actual data from the Titanic ship. You will try to predict the survival of the passengers.
   a. Apply the J48 tree classification algorithm in Weka to the dataset. Use the training set to build the tree and apply to the testing dataset to measure the prediction. How good is the prediction? Submit the resulting tree.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         774               77.4    %
Incorrectly Classified Instances       226               22.6    %
Kappa statistic                          0.51
Mean absolute error                      0.3275
Root mean squared error                  0.4093
Relative absolute error                 68.6324 %
Root relative squared error             83.8028 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.618     0.125      0.762     0.618     0.683      0.763    YES
                 0.875     0.382      0.78      0.875     0.825      0.763    NO
Weighted Avg.    0.774     0.281      0.773     0.774     0.769      0.763
```

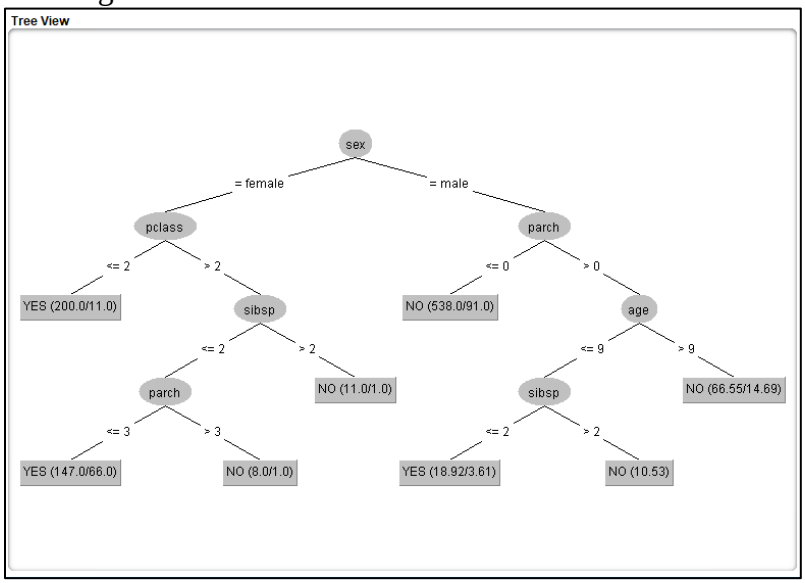



```
=== Summary ===

Correctly Classified Instances         236               76.3754 %
Incorrectly Classified Instances        73               23.6246 %
Kappa statistic                          0.47
Mean absolute error                      0.3442
Root mean squared error                  0.4219
Relative absolute error                 73.6739 %
Root relative squared error             88.2369 %
Total Number of Instances              309

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.626    0.163    0.670      0.626    0.647      0.471    0.731     0.549     YES
               0.837    0.374    0.809      0.837    0.822      0.471    0.731     0.783     NO
Weighted Avg.  0.764    0.301    0.761      0.764    0.762      0.471    0.731     0.702
```

Correctly classified instances in training data is 77.4%. When testing data is tested using the model, although it is quite close to the model, data accuracy is not as desired. Also, it is important to note that there are 13 attributes in the file. However, only 2 are used to construct the decision tree. Hence, We would not consider this a very good prediction.

b. While machine learning is a powerful tool, it is still unproductive to throw all kinds of variables into the mix and “let machine determine” the outcome. Examine the set variables carefully. Remove those that

may not make sense to be included here until you get a better tree. How good is the prediction? Submit the resulting tree.




```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         802               80.2    %
Incorrectly Classified Instances       198               19.8    %
Kappa statistic                          0.5786
Mean absolute error                      0.281
Root mean squared error                  0.3809
Relative absolute error                 58.8846 %
Root relative squared error             77.9813 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.705     0.135     0.772      0.705     0.737      0.580     0.819     0.772     YES
               0.865     0.295     0.819      0.865     0.841      0.580     0.819     0.827     NO
Weighted Avg.  0.802     0.232     0.800      0.802     0.800      0.580     0.819     0.805
```

```
=== Summary ===

Correctly Classified Instances         252               81.5534 %
Incorrectly Classified Instances        57               18.4466 %
Kappa statistic                          0.5843
Mean absolute error                      0.2788
Root mean squared error                  0.3751
Relative absolute error                 59.6785 %
Root relative squared error             78.4664 %
Total Number of Instances              309

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.692     0.119     0.755      0.692     0.722      0.586     0.802     0.706     YES
               0.881     0.308     0.844      0.881     0.862      0.586     0.802     0.831     NO
Weighted Avg.  0.816     0.243     0.813      0.816     0.813      0.586     0.802     0.788
```

After removing fields like – customer number, name, ticket #, cabin, fare, embarked and home destination, the model accuracy rate has increased to 80%. This shows that we cannot let machine decide the outcome by feeding all the variables at once. It can lead to bad predictive models. By reducing the fields to only those that should matter, model has improved its accuracy rate by 4% and testing model also is a better fit to improved model(81% accuracy rate).