Assignment 3 - (Spring 2021)

Due Date: April 16, Friday 11:59PM

Submit your document electronically via Canvas.

1. Consider the training examples shown in the table below for a binary classification problem.



* 1. Compute the Gini index for the overall collection of training examples.

Answer:

**Class C0 = 10**

**Class C1 = 10**

Gini Index =

= 1

= 1

* 1. Compute the Gini index for the **Car Type** attribute using multi-way split.

Answer:

For Car Type We have three Values : Family, Sports and Luxury . We will calculate Gini Index for all Three values first.

**Family:**

**C0 = 1**

**C1 = 3**

Gini Index =

= 1

= 1

**Sports:**

**C0 = 8**

**C1 = 0**

Gini Index =

= 1

= 1

**Luxury:**

**C0 = 1**

**C1 = 7**

Gini Index =

= 1

= 1

**Weighted Average:**

= 0.1622

* 1. Compute the Gini index for the **Shirt Size** attribute using multi-way split.

Answer:

So, we have 4 different values for shirt size. Small, Medium, Large and Extra Large.

**Small:**

**C0 = 3**

**C1 = 2**

Gini Index =

= 1

= 1

**Medium:**

**C0 = 3**

**C1 = 4**

Gini Index =

= 1

= 1

**Large:**

**C0 = 2**

**C1 = 2**

Gini Index =

= 1

= 1

**Extra-Large:**

**C0 = 2**

**C1 = 2**

Gini Index =

= 1

= 1

**Weighted Average:**

**= 0.491**

* 1. Which attribute is better, Car Type, or Shirt Size?

Answer: To determine the better attribute among the Car Type and Shirt Size we will check the Gini Index for both the attribute and as we can see above the Gini Index is lower for Car Type in comparison to Shirt Size which suggested that the distribution of sample is better for Car Type. Hence the **Car Type** is better attribute.

1. Consider the following data set for a binary class problem.



Calculate the information gain when

1. splitting on attribute *A,* and

2)splitting on attribute *B*.

Which attribute would the decision tree induction algorithm choose?

Answer:

We first check the impurity for parent node:

= 4

*Epartent = −0.4 log 0.4 − 0.6 log 0.6 =* ***0.9710***

Now lets calculate the information gain after splitting on A is:

For all T in A:

= ***0.9852***

For all F in A:

= **0**

Now the information gain:

*=*

***= 0.2813***

Similarly, now lets calculate the information gain after splitting on B:

For all T in B:

= ***0.8113***

For all F in B:

***= 0.6492***

Now the information gain:

*=*0.9710-[0.4] **=*0.2565***

We get best result with attribute A . Hence, attribute A is right choice to split the node.

The following questions require you to use the Weka tool for the analysis work.

1. Decision Trees:

Run the Decision Tree algorithm (J48) on the breast-cancer.arff data with different specifications below and show your results.

1. Percentage Split

Run the J48 classifier on the Breast Cancer dataset by selecting 3 different training-testing splits: 50%-50%, 66%-34%, and 80%-20%, respectively, with the parameters binarySplits = True, minNumObj = 5, and other parameters in default values. Report the Accuracy, Precision, Recall, and F-measure scores of different runs in a table. Does the percentage of training data used affect the classifier’s performance? Which of the three splits has the highest percentage of correctly classified instances (i.e., accuracy) ?

Answer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train Test Split | Accuracy(%) | Precision | Recall | F-measure |
| 50 -50 | 68.53 | .664 | .685 | .667 |
| 66-34 | 75.26 | .747 | .753 | .736 |
| 80-20 | 68.42 | .693 | .684 | .631 |

**50-50 Accuracy:**

98/143 = 68.53

**66-34 Accuracy:**

73/97 = 75.26

**80-20 Accuracy:**

39/57 = 68.42

Yes the percentage of training data affects the classifier’s performance. We can clearly see the accuracy of different dataset size/ percentage having different rate of correctly classified instance. The highest accuracy is achieved by 66- 34 % training testing split.

1. Cross-Validation

Run the J48 classifier on the breast-cancer.arff data by 4 different number of folds (for example, perform 4 different runs using 5, 8, 10, and 20 folds in each run, respectively), with other parameters in default values. Report Accuracy, Precision, Recall, and F-measure scores of different runs in a table. Does the number of folds affect the performance scores? What is the best number of folds in this dataset that gives you the best accuracy?

Answer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. of Folds | Accuracy(%) | Precision | Recall | F-measure |
| 5 | 74.13 | .733 | .741 | .691 |
| 8 | 74.13 | .726 | .741 | .699 |
| 10 | 75.52 | .752 | .755 | .713 |
| 20 | 75.17 | .742 | .752 | .710 |

Accuracy calculation from confusion matrix:

5 folds : 212/286 = 74.1258

8 Folds: 212/286 = 74.1258

10 Folds: 216/286 = 75.5244

20 Folds: 215/286 = 75.1748

It would be right to say that the number of folds for testing and training datasets does affect accuracy. They are affecting it by 1 %.

The best accuracy was given by 10 Folds i:e 75.52.

1. Tree Pruning

Run the J48 classifier with different settings below and use the 10-fold cross-validation.

Question 1: Run J48 classifier with binary split only and without requiring the binary split (you may set parameter binarySplits as True or False in different runs) with other parameters in default values. Report Accuracy, Precision, Recall, and F-measure scores of different runs in respective tables. Visualize and output the trees (Right-click the result in “Results list” and choose the option “Visualize the tree”). Does enforcing binary split affect the classifier’s accuracy and the size of the tree? Show your result.

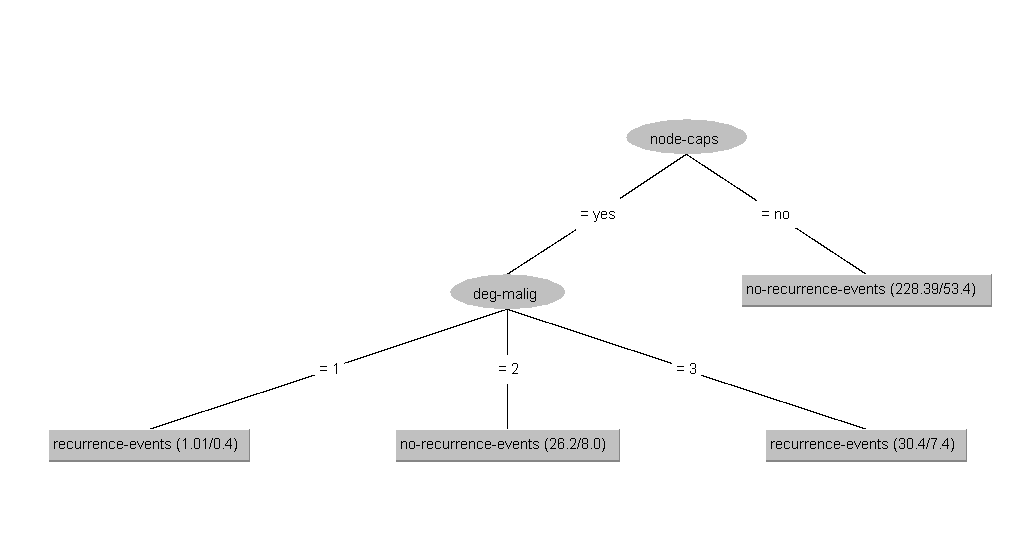
Answer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BinarySplit | Accuracy(%) | Precision | Recall | F-measure |
| True | 72.03 | .692 | .720 | .688 |
| False | 75.52 | .752 | .755 | .713 |

**BinarySplit = False**

Accuracy = 216/286 = 75.5244

**Tree**:

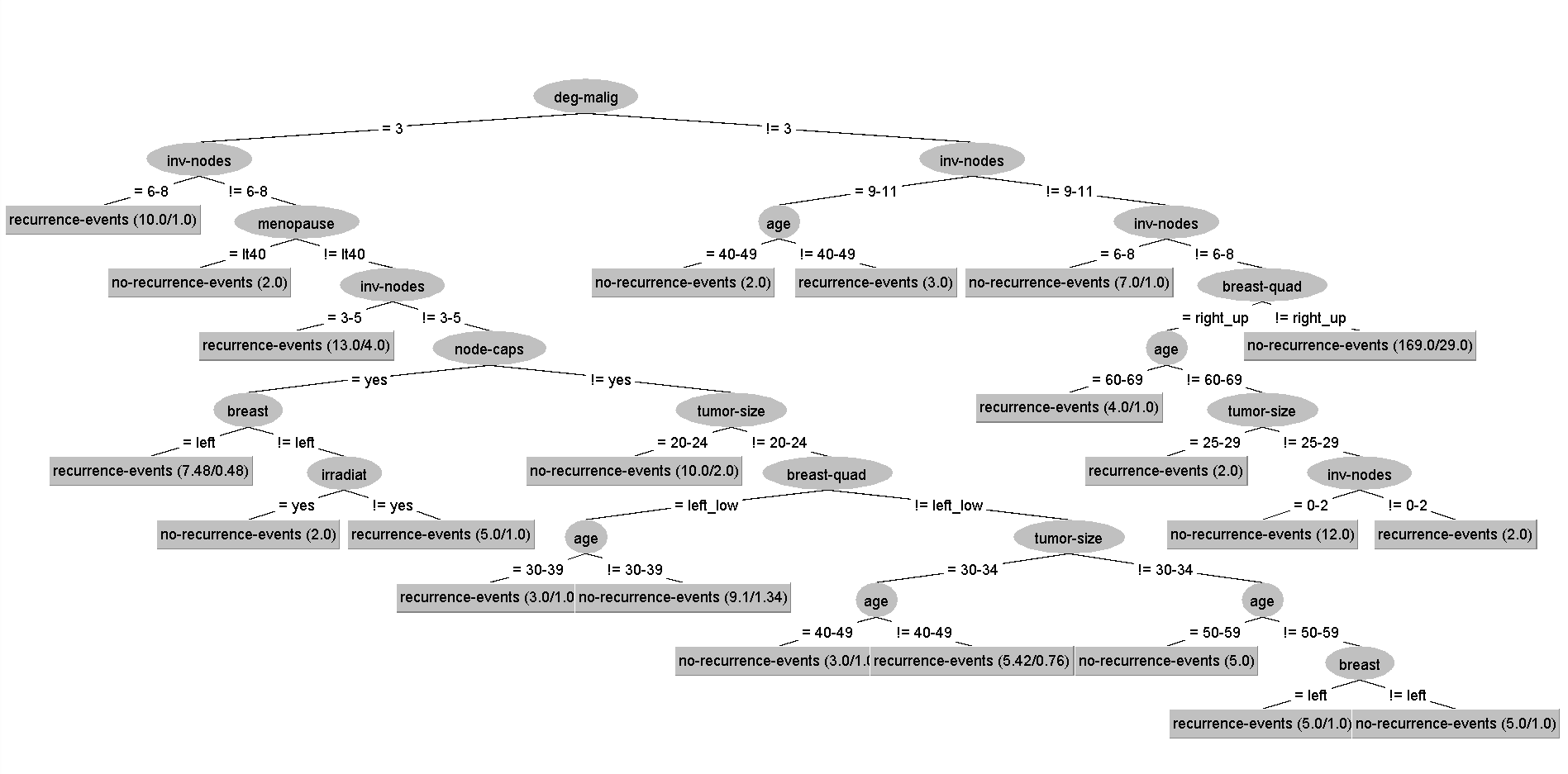


**BinarySplit = True**

Accuracy = 206/286 = 72.0279

**Tree**

Next Page:



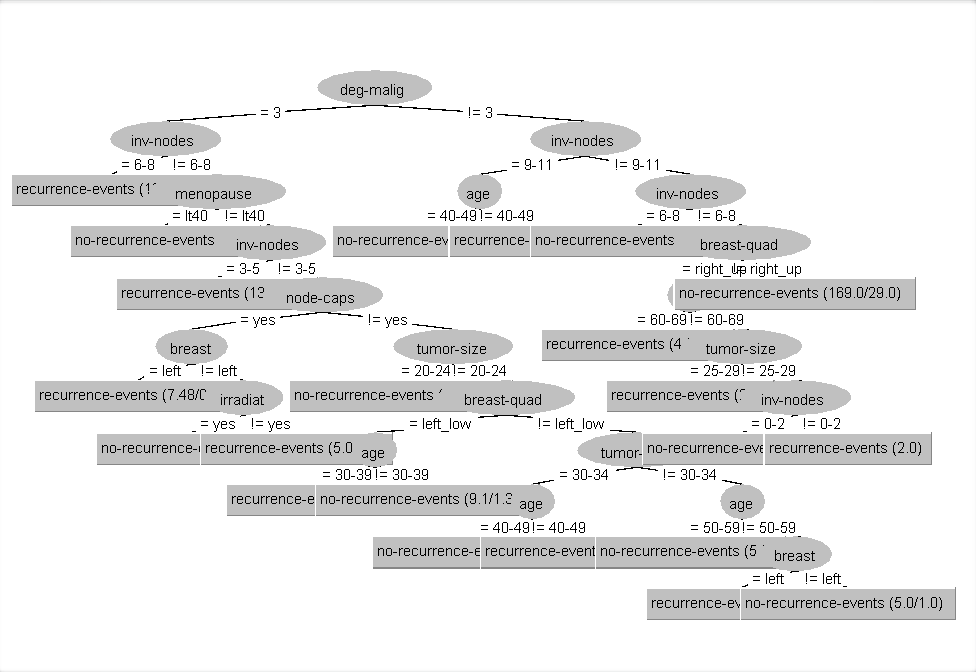
Yes, Binary split affects both the size and accuracy of classifier. When the split is true the accuracy is low and tree size is bigger.

Question 2: Run J48 classifier with different minimum numbers of objects allowed (for example, set parameter minNumObj as 2, 10, 20, and 40 in 4 different runs) while fixing binarySplits as True and other parameters in default values. Report Accuracy, Precision, Recall, and F-measure scores of different runs in respective tables. Visualize(output) the trees. Does the number of minimum objects allowed affect the classifier’s accuracy and the size of the tree? Show your result.

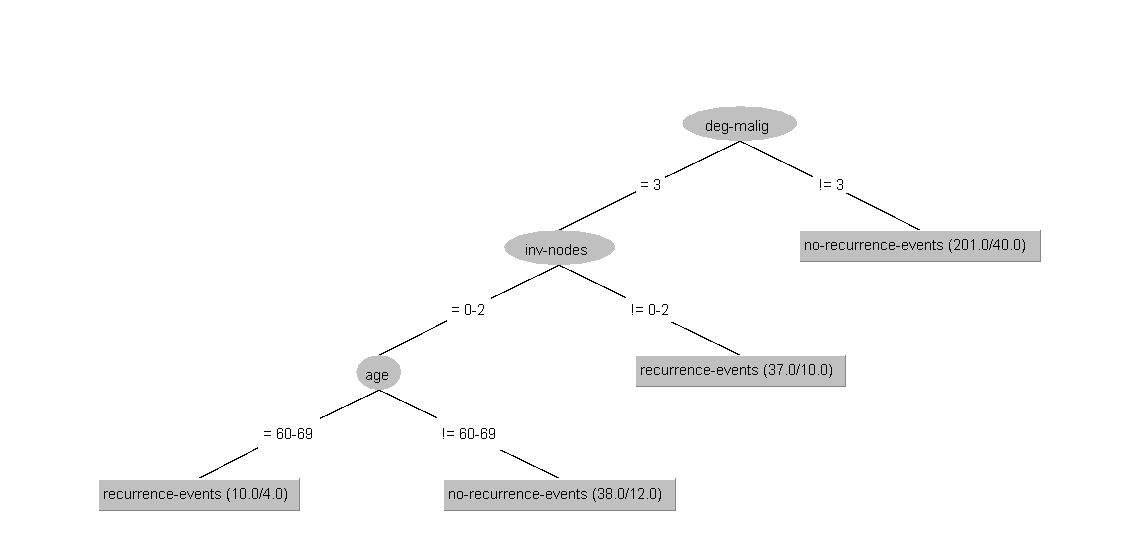
Answer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| minNumObj | Accuracy(%) | Precision | Recall | F-measure |
| 2 | 72.03 | .692 | .720 | .688 |
| 10 | 72.73 | .702 | .727 | .691 |
| 20 | 75.52 | .744 | .755 | .722 |
| 40 | 65.73 | .643 | .657 | .649 |

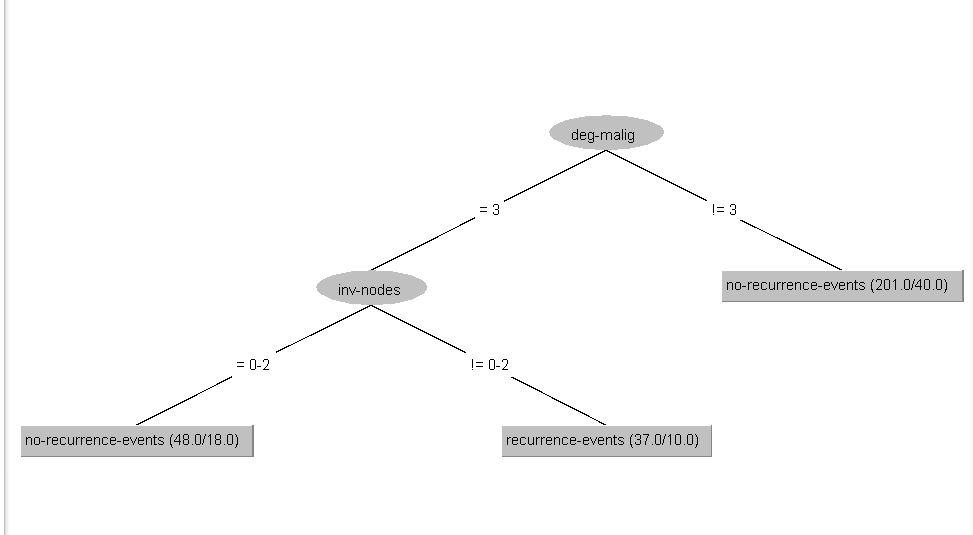
**Tree For minNumObj = 2**



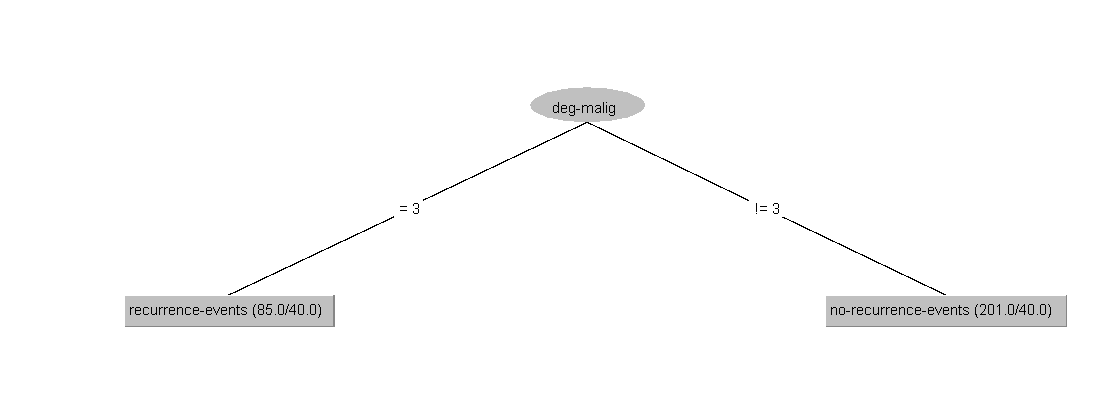
**Tree For minNumObj = 10**



**Tree For minNumObj = 20**



**Tree For minNumObj = 40**

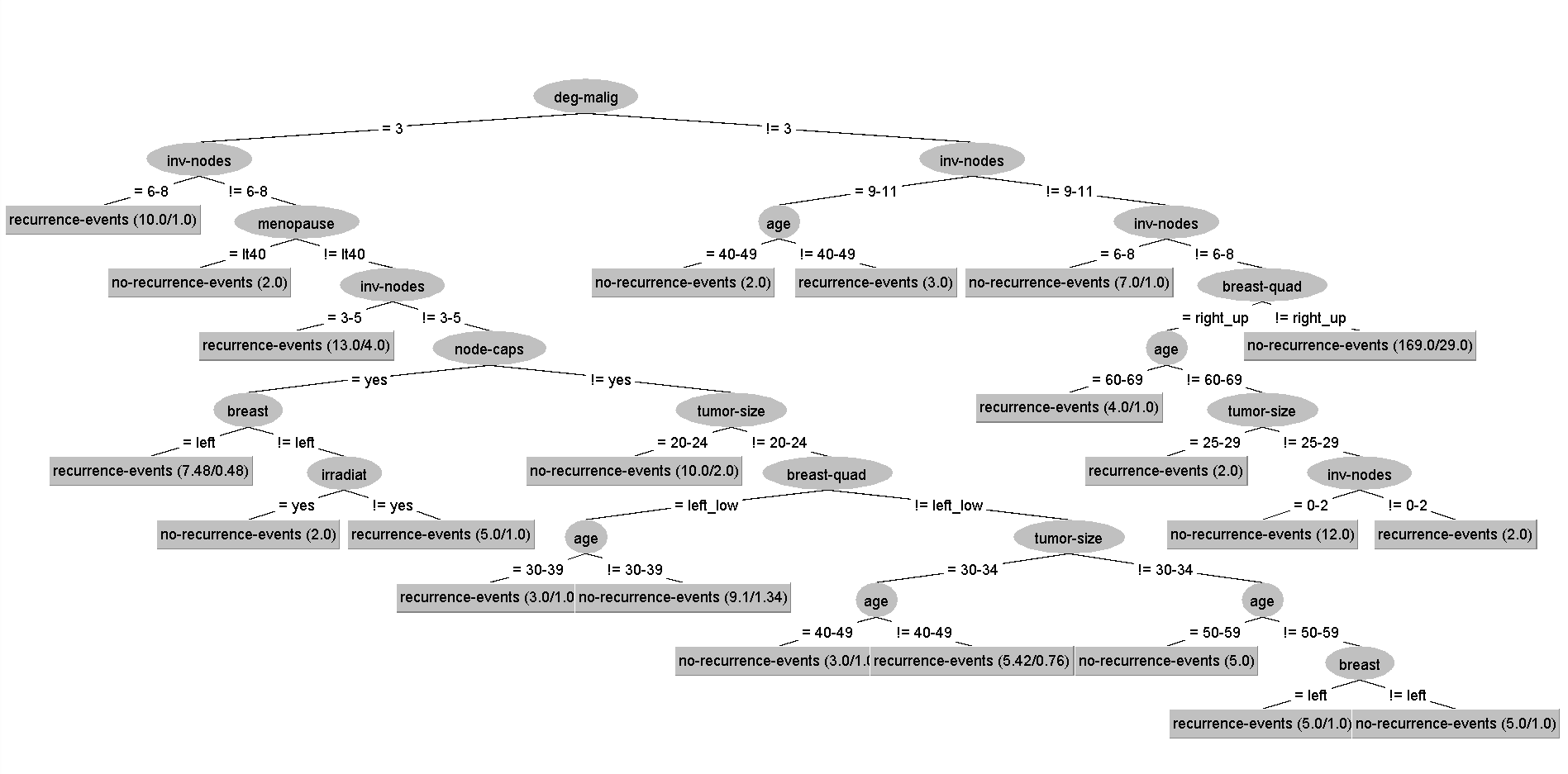


Yes, Indeed the minimum number of objects allowed affects the accuracy of classifier and tree size both. We can see that the best accuracy and tree size is achieved with minNumObj equals to 20. First the accuracy increases till minNumObj = 20 after that it started to decrease.

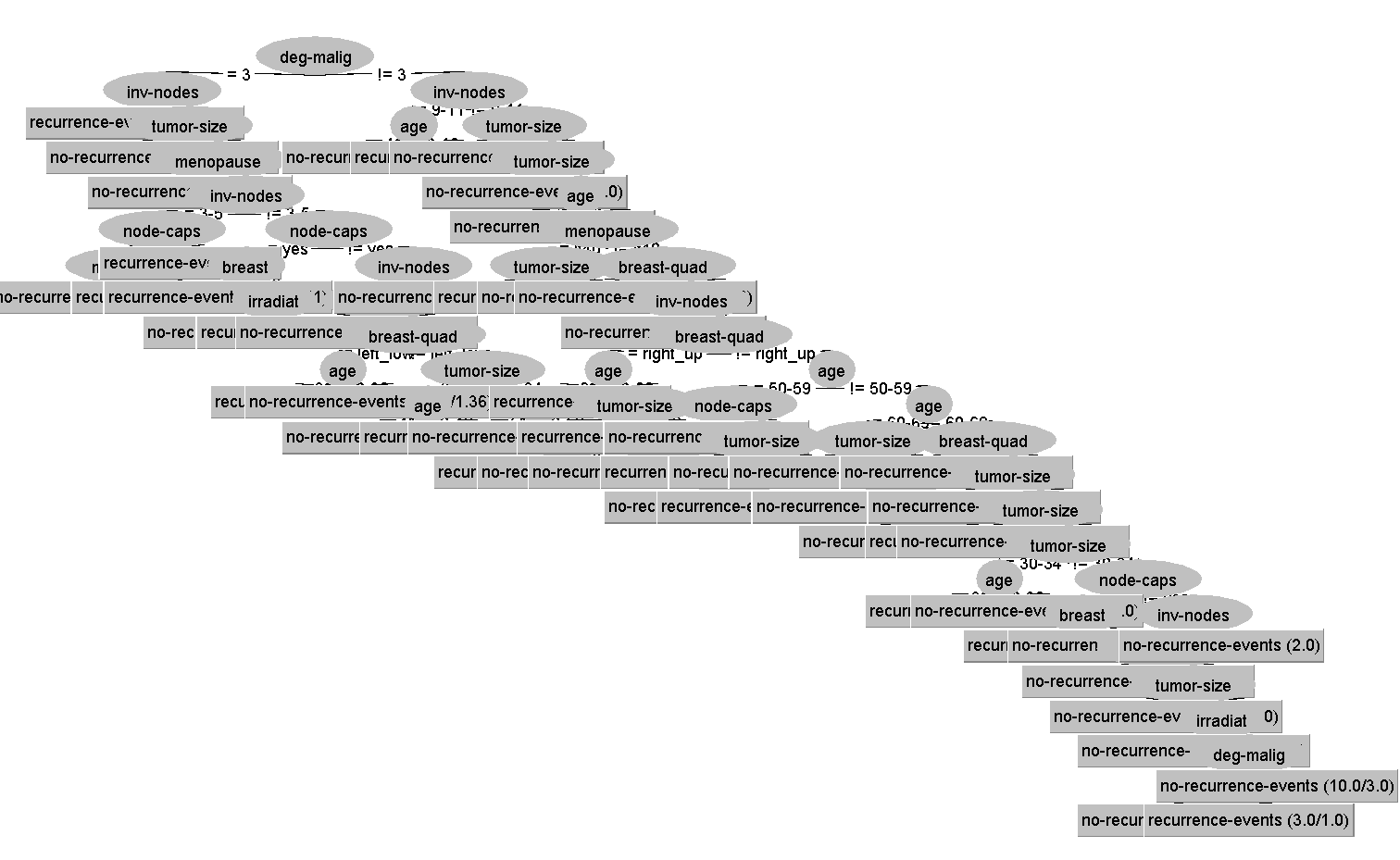
Question 3: Run J48 classifier with and without pruning (for example, set the parameter **unpruned** as False or True in 2 different runs) while fixing binarySplits as True and other parameters in default values. Report Accuracy, Precision, Recall, and F-measure scores of different runs in respective tables. Visualize (output) the trees. Does pruning the tree affect the classifier’s accuracy and the size of the tree? Show your result.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pruned | Accuracy(%) | Precision | Recall | F-measure |
| True | 72.03 | .692 | .720 | .688 |
| False | 62.58 | .606 | .626 | .615 |

When **Pruned True** Tree looks like as below:



When **unpruned** Tree looks like as below:



Yes, Pruning and pruning of Tree affects both the accuracy of classifier and size of Tree. When using pruned Tree, the size of tree is small and convenient to understand, also the accuracy is higher in comparison to the unpruned tree.

4 K-Nearest Neighbor classifier

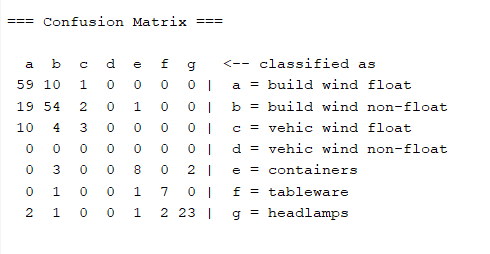
Run the K-Nearest Neighbor algorithm (IBk) on the following data set and answer the questions.

1. Run the IBK classifier on the Glass dataset by selecting 4 different values for *K* (i.e., the number of nearest neighbors considered): 1, 3, 5 , and 7, with the number of folds =10, and Euclidean as the distance metric. Does *K* affect the classifier’s accuracy? What is the value of *k* that leads to the best performance on this dataset? For the results obtained, explain the confusion matrix values for the one with the highest accuracy.

Answer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K value | Accuracy(%) | Precision | Recall | F-measure |
| 1 | 70.56 | .709 | .706 | .704 |
| 3 | 71.96 | .718 | .720 | .708 |
| 5 | 67.76 | .635 | .678 | .651 |
| 7 | 64.02 | .636 | .640 | .617 |

The best accuracy is achieved when K value = 3 below is the screenshot of confusion Matrix for K value = 3.



The above confusion matrix helps us to understand how many samples are classified correctly or incorrectly for individual classes on the testing data. We can read from confusion matrix first row that the total instances of class “a” is 70, among these 70 correctly classified instances are 59 ,whereas 11 instances of class “a” is misclassified. To know in which class these misclassified instances are classified, we read the table further in same row and we can see that the 10 instances are classified as class “b” and 1 as class “c”.

Just as explained above we can read all other class entries row wise for b ,c,d,e,f and g .With the help of all the information we can calculate the accuracy by dividing the sum of all the diagonal values of confusion matrix by total no of instances. Confusion matrix diagonal reading give the correctly classified instances count for all the classes together.

So, from above confusion matrix we can conclude below are correctly classified instances of each class:

a – 59

b – 54

c – 3

d – 0

e – 8

f – 7

g – 23

The accuracy is

From the confusion matrix we come to know about that for class “d -vehic wind non-float” there is no data in test dataset. It also gives us the idea how many data we have for individual classes in test set, also we can predict that our classifier is classifying the maximum instances in class “a” with count of 90.

1. Run the IBK classifier on the Glass dataset with the same above K values (i.e., 1, 3, 5, and 7) and applying 3 different distance metrics (Euclidean, Chebyshev and Manhattan Distance). Consider the number of folds = 10. Does the distance metric affect the classifier’s accuracy? For the results obtained , which distance metric(s) provide the best accuracy for a given value of K (for k = 1, 3, 5, 7)? List out your results.

Answer:

Yes, the distance metrics affects the accuracy of classifiers. It is clear that for K = 1 , 5 and 7 as the distance metric is changed the accuracy of KNN classifier changes too, whereas for K=3 the accuracy is same in case of Chebyshev and Manhattan distance metrics but it has changed for Euclidean.

**K = 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Distance Metrics | Accuracy(%) | Precision | Recall | F-measure |
| Euclidean | 70.56 | .709 | .706 | .704 |
| Chebyshev | 67.757 | .669 | .678 | .672 |
| Manhattan | 73.3645 | .741 | .734 | .733 |

Here **the best accuracy is achieved by Manhattan Distance** Metrics classifier.

**K =3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Distance Metrics | Accuracy(%) | Precision | Recall | F-measure |
| Euclidean | 71.96 | .718 | .720 | .708 |
| Chebyshev | 71.4953 | .704 | .715 | .698 |
| Manhattan | 71.4953 | .705 | .715 | .700 |

Here the **best accuracy is achieved by Euclidean Distance** Metrics classifier.

**K =5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Distance Metrics | Accuracy(%) | Precision | Recall | F-measure |
| Euclidean | 67.76 | .635 | .678 | .651 |
| Chebyshev | 65.4206 | .606 | .654 | .625 |
| Manhattan | 69.6262 | .685 | .696 | .677 |

Here the **best accuracy is achieved by Manhattan Distance** Metrics classifier.

**K = 7**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Distance Metrics | Accuracy(%) | Precision | Recall | F-measure |
| Euclidean | 64.02 | .636 | .640 | .617 |
| Chebyshev | 61.6822 | .574 | .617 | .583 |
| Manhattan | 68.6916 | .683 | .687 | .668 |

Here the **best accuracy is achieved by Manhattan Distance** Metrics classifier.

1. K-means clustering

Run the K-means algorithm (SimpleKMeans) on the glass.arff data set using different **k** values (for example, k= 3, 6, 9, 15 in 4 different runs), and use the Euclidean distance metric and other default parameter values. Report the within-cluster sum of squared errors in different runs in a table. For the results obtained, what is the value of **k** that produces the lowest sum of squared errors?

|  |  |
| --- | --- |
| K value | sum of squared errors |
| 3 | 77.1242 |
| 6 | 52.1858 |
| 9 | 48.0487 |
| 15 | 32.5696 |

For K = 15 we have achieved the lowest sum of errors.