Name : Chandramohan Natarajan

Assignment – 3

Student id – 190617866

Module – ECS766P

Exercise 0: Compare the Number of Leaves and Size of Tree of both trees (i.e. with and without pruning) and explain any differences observed. [0.5 mark]

Pruning is reducing the size of trees to work on efficiency of classification. After pruning, branches are reduced. The main objective behind the pruning is to keep tab on over fitting.

J48 pruned tree

Scheme: weka. classifiers. trees. J48 -C 0.25 -M 2

Relation: soybean

Instances: 683

Attributes: 36

Number of Leaves: 61

Size of the tree: 93

Correctly Classified Instances 210 90.5172 %

J48 unpruned tree

Number of Leaves: 121

Size of the tree: 175

Correctly Classified Instances 201 86.6379 %

Scheme - Weka classifiers -J.48 0.25 -M 2			
Instances		683	
Attributes		36	
Type	Number of leaves	Size of the tree	Accuracy
Pruned	61	93	90.52%
Unpruned	121	175	86.64%

Exercise 1: Compare the Test Accuracy of both trees. Which tree shows a better performance? Explain your observation based on the notion of tree pruning. [0.5 mark]

Scheme - Weka classifiers -J.48 0.25 -M 2			
Instances		683	
Attributes		36	
Type	Number of leaves	Size of the tree	Accuracy
Pruned	61	93	90.52%
Unpruned	121	175	86.64%

The prime objective behind the pruning process is reduce the size of decision trees by removing sections of the tree that provide little power to classify instances. Pruning reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting. The tree shows a better performance after pruning is done on it.
When tree was unpruned, test accuracy was recorded as 86.5% and after it was pruned this accuracy jumped up to 90.5%. The outcome is quite evident from the parameters of accuracy.

Exercise 2: Which class is being heavily mis-classified? Why has this happened? [1 mark]

As there are more instances of some classes, the model classified the instances to the available dataset and displayed impressive accuracy. In the presented dataset, number of instances for Setosa and Versicolor is high as compared to very few of Virginica. It is observed that Virginica class is heavily mis-classified in imbalanced classes, even though the overall clearly palpable that accuracy of the algorithm is a 90% which is just due to a lack of enough data instances of the Virginica class. As there are less samples, our model is not able to determine the quality or the factors of a virginica flower. The model requires more instances to zero in on the classification aspect about a particular sample to be Virginica. Imbalanced classes, because of more instances for some classes in proportion to others is prime reason for the misleading output . Our model cannot estimate the likelihood of presence of a data sample. In our dataset, the difference in number of instances for each class is high, that is why it affects the accuracy. So with such availability, its highly misleading to classify just based on accuracy parameter.

Exercise 3: Obtain the accuracy for this class from the test dataset and identify the other class that it is being confused with. [1 mark]

```
Test confusion matrix:
[[24  1  0]
 [ 0 25  0]
 [ 0  3  2]]
```

The accuracy for virginica is 40 percent. Due to unavailability of enough samples, which leads to imbalanced classes, virginica has been identified incorrect due to the presence in different decision region i.e versicolor.

Exercise 4: What is the new accuracy for class 2 (virginica)? Compare this accuracy with the accuracy obtained in the previous section and explain any discrepancies. [1 mark]

```
Normalised test confusion matrix:

[[0.96 0.04 0.  ]
 [0.   1.   0.  ]
 [0.   0.56 0.44]]
```

Though the geographic region is changed , the presence of number of samples does not have any change in classification. This is evident from insignificant growth in accuracy of class 2,irrespective of increased presence in data set. The accuracy is highly marginal as 11 in a set of 25 instances were classified correctly and the rest were still classified as class 2. The training pattern of algorithm still holds the toll and, being the change in region does not matter to the model, it is trained with parameters in model.40 % percent accuracy has very least impact though provided with more instances with change in region.

Exercise 5: What prior should you use to get maximum accuracy in region B? What accuracy do you get by using this value? [1 mark]

Possible priors are:

1) Prior 1 used are as follows..

P(0) = 4/11 → For class 0
P(1) = 1/11 → For class 1
P(2) = 6/11 → For class 2

With this priors, overall test accuracy is 81.8%. For class 2, it is observed an increase to 76% correctly classified instances.

```
gnb_A_with_uniform_priors = GaussianNB(priors=np.array([4/11,1/11,6/11]))
gnb_A_with_uniform_priors.fit(XtrImbalanced_A, YtrImbalanced_A)


print_classifier_report(XtrImbalanced_A, YtrImbalanced_A, XteImbalanced_B, YteImbalanced
_B, gnb_A_with_uniform_priors, True)
-------------------------------------------------------------------------------------
Test accuracy = 0.818
Test confusion matrix:
 [[24  1  0]
 [ 0  2  3]
 [ 0  6 19]]
-------------------------------------------------------------------------------------

Normalised test confusion matrix:
 [[0.96 0.04 0.  ]
 [0.   0.4  0.6 ]
 [0.   0.24 0.76]]
```

2) Prior 2 used are as follows..

P(0) = 5/11 → For class 0
P(1) = 1/11 → For class 1
P(2) = 5/11 → For class 2

```
gnb_A_with_uniform_priors = GaussianNB(priors=np.array([5/11,1/11,5/11]))
gnb_A_with_uniform_priors.fit(XtrImbalanced_A, YtrImbalanced_A)

print_classifier_report(XtrImbalanced_A, YtrImbalanced_A, XteImbalanced_B, YteImbalanced
_B, gnb_A_with_uniform_priors, True)
-----------------------------------------------------------------------------------------
Test confusion matrix:
 [[24  1  0]
 [ 0  2  3]
 [ 0  6 19]]
-----------------------------------------------------------------------------------------
Normalised test confusion matrix:
 [[0.96 0.04 0.  ]
 [0.   0.4  0.6 ]
 [0.   0.24 0.76]]
```

Exercise 6: Compare the performance of both classifiers in the 2-feature scenario with the performance in the 200-feature scenario and explain any differences you might observe. [1 mark]

Classification	Parameters	n=2	n=200
Logistic Regression	Train accuracy	0.78	1
	Test accuracy	0.72	0.78
Naïve Bayes	Train accuracy	0.78	1
	Test accuracy	0.72	1

When there is increase in dimensions, ie 2 to 200, there is possible scenario of overfitting which is quite **evident from facts. At the same time, less number of dimensions** may lead to underfitting . More dimensions infer more parameters/features which makes classification quite easy for model to perform. Similarly less dimensions are also not good for the model.