ECS766 Assignment 3 `

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]

	pruned	unpruned
Size of Trees	93	175
Number of Leaves	61	121

In case of pruned Decision Tree(DT) we see a lower number of leave nodes and size of tree is also small while in case of unpruned DT we see a large number of leave nodes and size of tree is also big. This affects the training data for both the models as chances of overfitting can happen for unpruned DT and underfitting for pruned DT.

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]

	pruned	unpruned
Size of Trees	93	175
Number of Leaves	61	121
Accuracy(%)	90.51	86.6

With a large size of tree and leaf nodes, we see a lower accuracy for unpruned Decision Tree(DT) where as in case of pruned DT a higher accuracy is obtained on smaller tree size and less leaf nodes. In case of unpruned DT, we here see overfitting, one way to stop is by pruning. When we do pruning, we stop splitting when nodes get small and optimize the number of actual nodes which contribute to classification of different classes.

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

Virginica (Class 2) is being heavily mis-classified(60% test misclassification). This is because of a smaller number of training sample available. As the logistic regression, P(A)P(1-A), fails to estimate the likelihood of presence of data sample due to less likely chance of them occurring as there are very few samples available to learn from.

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]

The Normalized test confusion matrix gives us some insight

```
[[0.96 0.04 0.  ]
 [0.   1.   0.  ]
 [0.   0.6  0.4 ]]
```

here the Accuracy of class 2(Virginica) is 40% and the other class which has been confused with it is class 1(Versicolour).

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]

The new accuracy obtained for region b using model of region a for class 2(virginica) is 44% which is barely any improvement over previous 40% as the model has only learnt in region and it picks the rules learnt in region a.

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]

Since we are moving to region b then we should use the priors of region b on model built on region a. This means for region a

P(class=0) = 25/55

P(class=1) = 5/25

P(class=2) = 25/55

By using these priors, we see increase in accruracy for class=2 which was 44% in case of logistic regression to 76%.

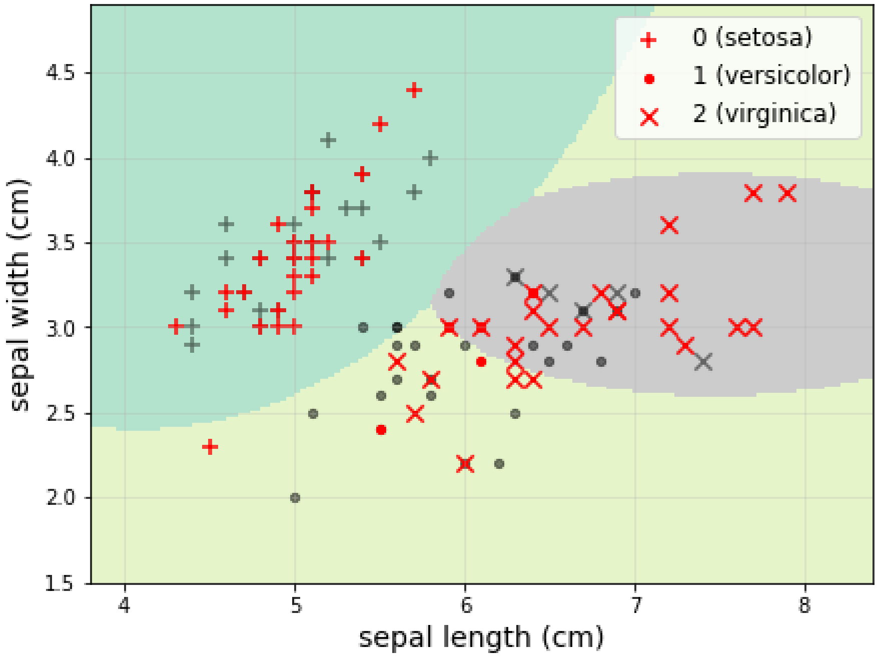
```
# [25/55,5/55/,25/55]
gnb_A_with_uniform_priors = GaussianNB(priors=np.array([25/55,5/55,25/55]))
gnb_A_with_uniform_priors.fit(XtrImbalanced_A, YtrImbalanced_A)
print_classifier_report(XtrImbalanced_A, YtrImbalanced_A, XteImbalanced_B, YteImbalanced_B,
gnb_A_with_uniform_priors, True)
```

```
Train accuracy = 0.782
Train confusion matrix:
[[25  0  0]
[ 3 13  9]
[ 0  0  5]]


Test accuracy = 0.818
Test confusion matrix:
[[24  1  0]
 [ 0  2  3]
 [ 0  6 19]]


Normalised train confusion matrix:
[[1.   0.   0.  ]
 [0.12 0.52 0.36]
 [0.   0.   1.  ]]


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]

```
|    2-dim        |Logistic Regression|Naive Bayes|

| --------------- |:-----------------:| ---------:|

|Train Accuracy   |     78            |        78 |

|Test Accuracy    |     72            |        72 |


|    200-dim      |Logistic Regression|Naive Bayes|
```

Train Accuracy	100	100
Test Accuracy	78	100

From above we clearly see the case of overfitting. As we increase the number of dimensions, we are increasing the number of features from which our model can learn from, but the number of instances are constant. Too many dimensions cause every observation in the dataset to appear equidistant from all the others, which is why it is easily separable and thus the above results. It is also known as the curse of dimensionality, as we increase the dimensions the performance of the model after rising to a peak decreases rapidly.