Accuracies using binary Univariate split:

(entropy, no pruning, no_of_th = 30)

```
accuracy-fold 1 : 0.7939
accuracy-fold 2 : 0.8265
accuracy-fold 3 : 0.7857
accuracy-fold 4 : 0.8061
accuracy-fold 5 : 0.8265
accuracy-fold 6 : 0.8143
accuracy-fold 7 : 0.8245
accuracy-fold 8 : 0.8102
accuracy-fold 9 : 0.8323
accuracy-fold 10 : 0.8160
Cross validation Accuracy: 0.8136
```

Improvements

1. *Gini Index:*

   Using Gini index over entropy improves the accuracy, but very slightly.

2. *Pre-Pruning:*

   This helps in improving accuracy and generalisation ability of the tree. The logic can be explained as follows: Suppose a node gets a set of points such that, p(class 0) is very close to 0(or 1). This means most of the points are of class 1(or class 0), and very few are from the other class. These minority points might be noise or outliers in the data, which need to be handled. Instead of setting a hard rule of p(class 0)=0(or 1) for terminating, we can give a range of [0,0.05](or [0.95,1]). This way, we can take care of such data and avoid overfitting. Another advantage of pre-pruning is that the algorithm is faster, as it does not go till the leaf level.

3. *reducing no_of_th:*

   This action seemed to improve accuracy considerably. no_of_th is the number of threshold values which I am checking for a given attribute, to find an optimal threshold for splitting data based on that attribute. Initially, when I ran my code for no_of_th=30, my code gave accuracy in folds between 78%-82%. Overall cross validation accuracy was also around 80%. But when I reduced this number, the accuracy improved consistenly. For no_of_th=10, some folds even touch 85%-86% accuracy. I think this happens because, when we try to increase no_of_th, we are overfitting the threshold values for which, data points split into subtrees. By taking lesser no_of_th, we can get a better idea of the splitting threshold values of attributes, and hence, generalise better. Reducing no_of_th also reduces the time taken by the code, as it now has to check 1/3 (if we reduce it from 30 to 10) of the values as before.

Accuracies using binary Univariate split:

(Gini index, pre-pruning, no_of_th = 10)

```
accuracy-fold 1 : 0.8408
accuracy-fold 2 : 0.8592
accuracy-fold 3 : 0.8102
accuracy-fold 4 : 0.8184
accuracy-fold 5 : 0.8469
accuracy-fold 6 : 0.8102
accuracy-fold 7 : 0.8327
accuracy-fold 8 : 0.8327
accuracy-fold 9 : 0.7975
accuracy-fold 10 : 0.8078
Cross validation Accuracy: 0.8256
```