

Programming Assignment #2

Team Members:

Tarun Punhani – txp190029

Kiran Devaraj Raj – kxr190038

a) Bagging Results:

```
Bagging results for max_depth = 3 and bag_size = 10
Training Accuracy=73.87165599868702, Testing Accuracy=74.1999015263417

Confusion Matrix for Bagging with max_depth = 3 and bag_size=10
[[663 524]
 [  0 844]]
Bagging results for max_depth = 3 and bag_size = 20
Training Accuracy=73.87165599868702, Testing Accuracy=74.1999015263417

Confusion Matrix for Bagging with max_depth = 3 and bag_size=20
[[663 524]
 [  0 844]]
Bagging results for max_depth = 5 and bag_size = 10
Training Accuracy=88.83965205974069, Testing Accuracy=88.38010832102412

Confusion Matrix for Bagging with max_depth = 5 and bag_size=10
[[986 201]
 [ 35 809]]
Bagging results for max_depth = 5 and bag_size = 20
Training Accuracy=88.83965205974069, Testing Accuracy=88.38010832102412

Confusion Matrix for Bagging with max_depth = 5 and bag_size=20
[[986 201]
 [ 35 809]]
```

b) Boosting Results:

```
Boosting results for max_depth = 1 and bag_size = 20
Training Accuracy=71.03233218447397, Testing Accuracy=70.90103397341213

Confusion Matrix for Boosting with max_depth = 1 and bag_size=20
[[647 540]
 [ 51 793]]
Boosting results for max_depth = 1 and bag_size = 40
Training Accuracy=71.03233218447397, Testing Accuracy=70.90103397341213

Confusion Matrix for Boosting with max_depth = 1 and bag_size=40
[[647 540]
 [ 51 793]]
Boosting results for max_depth = 2 and bag_size = 20
Training Accuracy=71.03233218447397, Testing Accuracy=70.90103397341213

Confusion Matrix for Boosting with max_depth = 2 and bag_size=20
[[647 540]
 [ 51 793]]
Boosting results for max_depth = 2 and bag_size = 40
Training Accuracy=71.03233218447397, Testing Accuracy=70.90103397341213

Confusion Matrix for Boosting with max_depth = 2 and bag_size=40
[[647 540]
 [ 51 793]]
```

c) Scikit-learn Results:

Bagging:

```
Boosting results for max_depth = 3 and bag_size = 10
Testing Accuracy=100.0

Confusion Matrix for Boosting with max_depth = 3 and bag_size=10
[[1187   0]
 [   0 844]]
Boosting results for max_depth = 3 and bag_size = 20
Testing Accuracy=100.0

Confusion Matrix for Boosting with max_depth = 3 and bag_size=20
[[1187   0]
 [   0 844]]
Boosting results for max_depth = 5 and bag_size = 10
Testing Accuracy=100.0

Confusion Matrix for Boosting with max_depth = 5 and bag_size=10
[[1187   0]
 [   0 844]]
Boosting results for max_depth = 5 and bag_size = 20
Testing Accuracy=100.0

Confusion Matrix for Boosting with max_depth = 5 and bag_size=20
[[1187   0]
 [   0 844]]
```

Scikit-learn Boosting

Adaboost:

```
Boosting results for max_depth = 1 and bag_size = 20
Testing Accuracy=93.205317577548

Confusion Matrix for Boosting with max_depth = 1 and bag_size=20
[[1061  126]
 [   12 832]]
Boosting results for max_depth = 1 and bag_size = 40
Testing Accuracy=93.205317577548

Confusion Matrix for Boosting with max_depth = 1 and bag_size=40
[[1061  126]
 [   12 832]]
Boosting results for max_depth = 2 and bag_size = 20
Testing Accuracy=99.55686853766618

Confusion Matrix for Boosting with max_depth = 2 and bag_size=20
[[1187    0]
 [    9 835]]
Boosting results for max_depth = 2 and bag_size = 40
Testing Accuracy=99.55686853766618

Confusion Matrix for Boosting with max_depth = 2 and bag_size=40
[[1187    0]
 [    9 835]]
```

What can you say about the quality of your implementation's performance versus scikit's performance?

Scikit is significantly better with accuracy than our own implementation. We think it is due to the other factors like gini index which scikit-learn considers while building the decision trees.