Machine Learning: Programming Assignment #2

Due on March 27, 2019 at 11:59pm

Professor Sriraam Natarajan Section 005

Viraj Mavani - VDM180000, Meet Shah - MXS170053

Part A

Construct four models for each combination of maximum depth d = 3, 5 and bag size (k = 10, 20). Report the confusion matrix for these four settings.

The confusion matrices for Part A can be found in the figures below.

Bagging: Max Depth 3 and Bag Size 10

	Classifier Positive	Classifier Negative
Actual Positive	815.0	29.0
Actual Negative	57.0	1130.0

Figure 1: Bagging: Max Depth 3 and Bags 10

Bagging: Max Depth 3 and Bag Size 20

	Classifier Positive	Classifier Negative
Actual Positive	815.0	29.0
Actual Negative	57.0	1130.0

Figure 2: Bagging: Max Depth 3 and Bags 20

Bagging: Max Depth 5 and Bag Size 10

	Classifier Positive	Classifier Negative
Actual Positive	844.0	0.0
Actual Negative	4.0	1183.0

Figure 3: Bagging: Max Depth 5 and Bags 10

Bagging: Max Depth 5 and Bag Size 20

	Classifier Positive	Classifier Negative
Actual Positive	844.0	0.0
Actual Negative	4.0	1183.0

Figure 4: Bagging: Max Depth 5 and Bags 20

Part B

Construct four models for each combination of maximum depth d= 1,2 and bag size (k= 20,40). Report the confusion matrix for these four settings.

The confusion matrices for Part B can be found in the figures below.

Boosting: Max Depth 1 and Number of Stumps 20

	Classifier Positive	Classifier Negative
Actual Positive	793.0	51.0
Actual Negative	176.0	1011.0

Figure 5: Boosting: Max Depth 1 and Number of Stumps 20

Boosting: Max Depth 1 and Number of Stumps 40

	Classifier Positive	Classifier Negative
Actual Positive	793.0	51.0
Actual Negative	176.0	1011.0

Figure 6: Boosting: Max Depth 1 and Number of Stumps 40

Boosting: Max Depth 2 and Number of Stumps 20

	Classifier Positive	Classifier Negative
Actual Positive	840.0	4.0
Actual Negative	0.0	1187.0

Figure 7: Boosting: Max Depth 2 and Number of Stumps 20

Boosting: Max Depth 2 and Number of Stumps 40

	Classifier Positive	Classifier Negative
Actual Positive	840.0	4.0
Actual Negative	0.0	1187.0

Figure 8: Boosting: Max Depth 2 and Number of Stumps 40

Part C

This section of the assignment, we are directed to training the scikit-learn implementations of Part A and Part B tasks and are expected to report the confusion matrices.
The matrices for bagging using scikit-learn are as follows:

Sklearn Bagging: Max Depth 3 and Bag Size 10

	Classifier Positive	Classifier Negative
Actual Positive	727.0	117.0
Actual Negative	2.0	1185.0

Figure 9: Sklearn Bagging: Max Depth 3 and Bags 10

Sklearn Bagging: Max Depth 3 and Bag Size 20

	Classifier Positive	Classifier Negative
Actual Positive	727.0	117.0
Actual Negative	2.0	1185.0

Figure 10: Sklearn Bagging: Max Depth 3 and Bags 20

Sklearn Bagging: Max Depth 5 and Bag Size 10

	Classifier Positive	Classifier Negative
Actual Positive	818.0	26.0
Actual Negative	8.0	1179.0

Figure 11: Sklearn Bagging: Max Depth 5 and Bags 10

Sklearn Bagging: Max Depth 5 and Bag Size 20

	Classifier Positive	Classifier Negative
Actual Positive	818.0	26.0
Actual Negative	8.0	1179.0

Figure 12: Sklearn Bagging: Max Depth 5 and Bags 20

The matrices for boosting using scikit-learn are as follows:

Sklearn Boosting: Max Depth 1 and Number of Stumps 20

	Classifier Positive	Classifier Negative
Actual Positive	842.0	2.0
Actual Negative	0.0	1187.0

Figure 13: Sklearn Boosting: Max Depth 1 and Number of Stumps 20

Sklearn Boosting: Max Depth 1 and Number of Stumps 40

	Classifier Positive	Classifier Negative
Actual Positive	844.0	0.0
Actual Negative	0.0	1187.0

Figure 14: Sklearn Boosting: Max Depth 1 and Number of Stumps 40

Sklearn Boosting: Max Depth 2 and Number of Stumps 20

	Classifier Positive	Classifier Negative
Actual Positive	844.0	0.0
Actual Negative	0.0	1187.0

Figure 15: Sklearn Boosting: Max Depth 2 and Number of Stumps 20

Sklearn Boosting: Max Depth 2 and Number of Stumps 40

	Classifier Positive	Classifier Negative
Actual Positive	844.0	0.0
Actual Negative	0.0	1187.0

Figure 16: Sklearn Boosting: Max Depth 2 and Number of Stumps 40

From the reported numbers we can say that our implementation of Bagging is closer to implementation of Scikit learn. Except the case of 10 Bags and Depth of Decision tree 3, we are out performing Scikit learn in all cases.
The difference between the results of using scikit-learn's implementation and our implementation is that our implementation is prone to underflow errors because of using normalized weights at each iteration of the AdaBoost algorithm.