Programming Assignment #4

Ensemble Methods

April 9th, 2023

Gautam Sapre (*gss170001*)

Sanjeev Penupala (*sxp170022*)

a) Bagging

MONKS-1 *DEPTH=3, BAG_SIZE=10* *test_error = 18.06%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	198	18
	Negative	60	156

MONKS-1 *DEPTH=3, BAG_SIZE=20* *test_error = 16.90%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	212	4
	Negative	69	147

MONKS-1 *DEPTH=5, BAG_SIZE=10* *test_error = 10.88%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	191	25
	Negative	22	194

MONKS-1 *DEPTH=5, BAG_SIZE=20* *test_error = 9.95%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	185	31
	Negative	12	204

b) Boosting

MONKS-1 *DEPTH=1, BAG_SIZE=20* *test_error = 25.00%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	216	0
	Negative	108	108

MONKS-1 *DEPTH=1, BAG_SIZE=40* *test_error = 25.00%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	216	0
	Negative	108	108

MONKS-1 *DEPTH=2, BAG_SIZE=20* *test_error = 25.00%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	216	0
	Negative	108	108

MONKS-1 *DEPTH=2, BAG_SIZE=40* *test_error = 25.00%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	216	0
	Negative	108	108

c) scikit-learn

Bagging

MONKS-1 *DEPTH=3, BAG_SIZE=10* *test_error = 8.33%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	204	12
	Negative	24	192

MONKS-1 *DEPTH=3, BAG_SIZE=20* *test_error = 7.18%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	213	3
	Negative	28	188

MONKS-1 *DEPTH=5, BAG_SIZE=10* *test_error = 14.12%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	209	7
	Negative	54	162

MONKS-1 *DEPTH=5, BAG_SIZE=20* *test_error = 5.09%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	210	6
	Negative	16	200

Boosting

MONKS-1 *DEPTH=1, BAG_SIZE=20* *test_error = 29.63%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	163	53
	Negative	75	141

MONKS-1 *DEPTH=1, BAG_SIZE=40* *test_error = 30.79%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	159	57
	Negative	76	140

MONKS-1 *DEPTH=2, BAG_SIZE=20* *test_error = 0.00%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	216	0
	Negative	0	216

MONKS-1 *DEPTH=2, BAG_SIZE=40* *test_error = 0.00%*		**Classifier Prediction**	
		Positive	Negative
Actual Value	Positive	216	0
	Negative	0	216

Performance

We can say sklearn classifies the MONKS dataset a lot better than our implementation of the ensemble methods. Even over many runs of the same program (due to randomness of the bootstraipping), sklearn will come out on top with better confusion matrices than our own implementations. There could be several reasons for this:

- The hyperparameters on the sklearn ensemble methods might be better suited for our dataset than our own implementation.
- Our implementation of boosting is slightly bugged for not accounting the weights properly (since we are getting a constant test error regardless of depth and bag size)
- Sklearn classifiers might have better bootstrapping methods, which results in higher accuracies
- Sklearn is not necessarily limited to binary splits, it can do multi-way splits, which may improve the decision tree boundaries