CS 422/622 Project 3

Due 11/15 11:59pm

Logistics: You must implement everything stated in this project description that is marked with an **implement** tag. Whenever you see the **write-up** tag, that is something that must be addressed in the write-up for the project. You may import the numpy, math, and random libraries. For this project, you are allowed to use the scikit learn decision tree library to build a tree with a max_depth=1.

```
from sklearn import tree
```

You should only need three functions from the library.

```
DecisionTreeClassifier(max_depth=1)

_.fit(X,Y)

_.predict([[x,y]])
```

A test_script.py file is provided to test your functions with. **You must implement everything else from scratch.**

Deliverables: You should submit a single ZIP file, containing your project code (*.py files) and your writeup (PDF or README.txt). Your zip file should be named lastname1_firstname2_project3.zip. For example, a zip file for Sara Smith might look like smith_sarah_project3.zip. Your code should run without errors on the ECC linux machines. If your code does not run for a particular problem, you will lose 50% on that problem. You should submit one py file, named accordingly (e.g. adaboost.py).

Extra Credit: Students in 422 who use LaTeX for their write-up and submit their LaTeX source files will get extra points.

1 Training Adaboost (65 points)

File name: **adaboost.py**

Implement a function in python:

```
adaboost_train(X, Y, max_iter)
```

that takes sample data, sample labels, and an iteration count as inputs. The function should return variables f and alpha. f is an array of trained decision tree stumps (a tree with a max depth of 1). alpha is a 1D array of calculated alpha values. To indicate sample weights in subsequent iterations, create a new dataset with duplicate samples. For example, if we have a dataset with 4 samples (s1, s2, s3, s4) with weights (1/8, 1/8, 1/4, 1/2), on the next iteration we will create a new dataset with 8 samples (s1, s2, s3, s3, s4, s4, s4, s4). Now s4 is half of all the samples for the dataset, while s1 and s2 only contribute 1/8 each to the whole dataset. At each iteration train a decision tree and get the predicted values from the test data, then compare to the actual labels. Use this to calculate the alpha and update the data with the new weights for the subsequent iterations.

Write-Up: What is the reason for using a decision tree stump rather than a decision tree with a greater depth? How does this differentiate adaboost from a random forest ensemble method?

2 Testing Adaboost (25 points)

File name: **adaboost.py**

Implement a function in python:

```
adaboost_test(X, Y, f, alpha)
```

that takes sample data, sample labels, the previously calculated array f, and the previously calculated array alpha as inputs. The function should return an accuracy indicating the overall performance of the adaboost algorithm.

Write-Up: What would need to change to run an adaboost algorithm with a perceptron rather than a decision tree?

3 Report (10 points)

622 is required to submit their report using LaTeX as a PDF and the source file. 422 can submit a README.txt or a LaTeX with source files for extra credit.