ADABOOST Report

Introduction to Adaboost and algorithm used:

Adaboost is defined as a booster and it works in a number of rounds. At round the 't' booster defines a distribution of weights over training examples, and the weak learner produces a weak hypothesis this implementation can be defined using an algorithm.

Algorithm:

input:
training set S = (x1, y1), . . . ,(xm, ym)
weak learner WL
number of rounds T
initialize D(1) = (1/m , . . . ,1/m).
for t = 1, . . . , T:
invoke weak learner ht = WL(D(t), S)
compute error_t=sum_i_m(Di(t)*1[yi!=ht(xi)])
let weight_t =1/2*log(1-et/et)
update D(t+1)i = D(t)i*exp(−wt*yi*ht(xi))/sum j=1 to m D(t)j*exp(−wt*yj*ht(xj))for all i = 1, . . . , m
output the hypothesis hs(x) = sign(sum t=1 to T weight_t*ht(x)).

By using the above defined algorithm, to calculate the error and update the weight, weak learners can be used in adaboost to produce a much better hypothesis.

Implementation of Code:

1. In the code defined I have defined a class which contains the different methods used for fitting the model and predicting the output of the defined model.
2. Used Decision tree classifiers as weak classifiers at each iteration to create appropriate decision stumps.
3. Created a function named plot_adaboost to plot the decision boundary of the adaboost classifier.
4. Used make_gaussian_quantiles to create a random data set of defined data points.
5. **For different values of data points change the value of variable 'n' and for different numbers of weak classifiers change the value of variable 'i'**
6. **For checking the value of error and weights set iteraion_information to True during model creation, otherwise set False.**

Experiments Conducted and Observations:

1. **Comparison with AdaBoostClassifier defined in sklearn.ensemble:** By comparing the two models for constant value of data points and iterations. It is observed that both the models have the same performance.

2. **Different Values of Iterations:** While using different values of iterations or weak learners it is observed that with a constant data points number of weak learners can vary but they are mostly less than n/2 where n is number of data points, also in some cases the number of weak learners is very less around 10 or 14 and in some cases it is around n/4.

Example:

1. For n = 160 the number of weak learners for 100% accuracy is around 47 which is nearly greater than n/4.
2. For n = 50 the number of weak learners for highest accuracy is around 14 which is nearly greater than n/4.