AdaBoost (Adaptive Boosting)

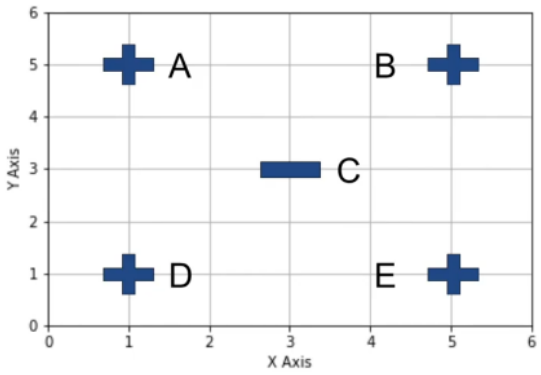
1) In AdaBoost, trees are just a node with two leaves. They are called stumps
2) Stumps are weak learners as it will only use one variable(feature) to make decisions
3) In AdaBoost many different Decision stumps are combined sequentially.
4) In Random Forest each tree has an equal vote(weightage) in the final classification
5) In AdaBooost made up of stumps, some stumps get more weight in the final classification than others
6) In Random Forest, each decision tree is made independent of the others.
7) In a Forest of stumps made up of AdaBoost, order is important. The errors that the 1st stump makes influences the 2nd stump, the errors that the 2nd stump makes influences the 3rd stump and so on.

To summarize

a) AdaBoost combines a lot of weak learners(Stumps) to make classifications
b) Some stumps have more say in the classification than others.
c) Each stump is made by taking previous stump mistake into account.

Example

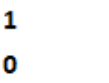
1) Consider 5 data points
2) Out of these 5 points, 4 belong to **class 1 (denoted by + sign)** and 1 point belong to class **0 (denoted by – sign)**.
3) Assign all the training points with Weights = 1/N, where N= total number of training points.



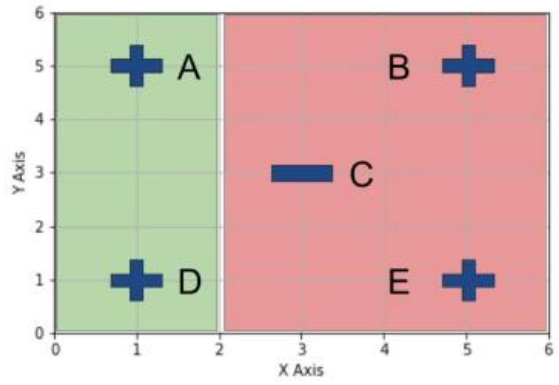
Points	Weight
Wa	1/5
Wb	1/5
Wc	1/5
Wd	1/5
We	1/5

4) AdaBoost uses **Decision Stump (A decision tree with just one split)**
5) Model considers multiple **Decision Stumps** and compares their error rates.
Let's consider the different classifiers and evaluate their error rates

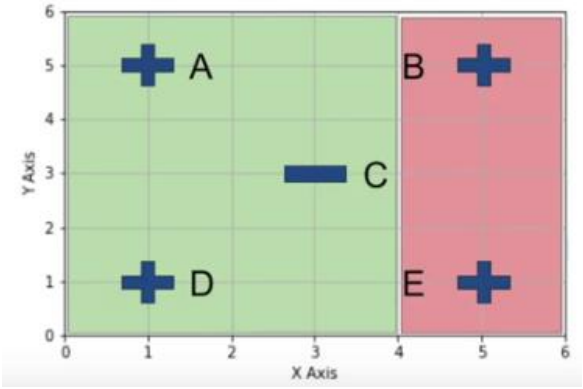
1st Iteration



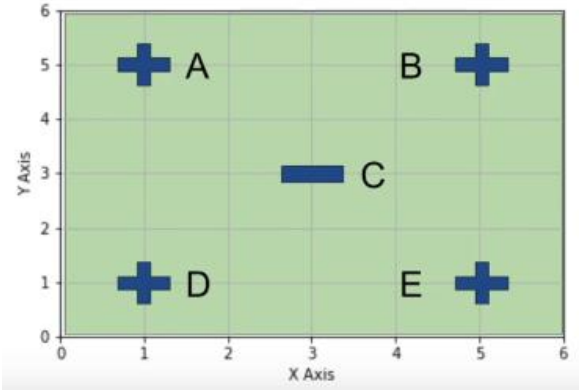
Classifier 1 (X<2)



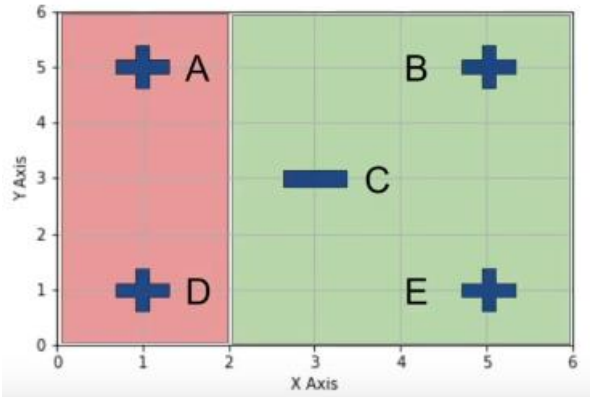
Classifier 2 (X<4)



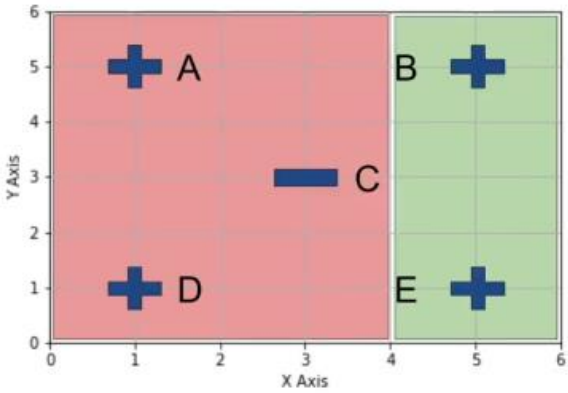
Classifier 3 (X<6)



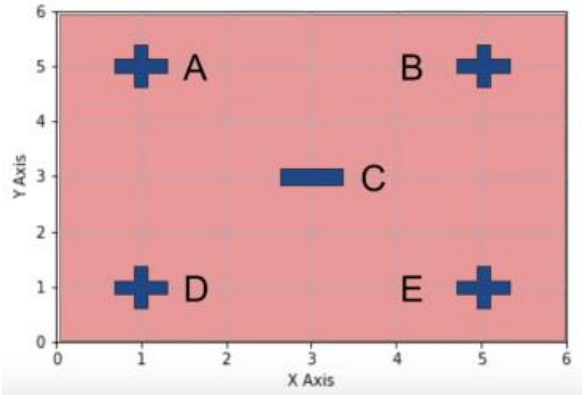
Classifier 4 (X>2)



Classifier 5 (X>4)



Classifier 4 (X>6)



Classifier	Wrong data points	Error(ε)
X<2	B, E	2/5
X<4	B, C, E	3/5
X<6	C	1/5
X>2	A, D, C	3/5
X>4	A, D	2/5
X>6	A, B, D, E	4/5

6) Select classifier with lowest Error rate => **X<6 is selected, with ε = 1/5**
7) Compute Voting power of the Classifier

   Voting Power (α) = (½)ln((1- ε)/ ε)
8) **α = (½)ln((1-(1/5))/(1/5)) = (½)ln4 = 0.5*1.3862 = 0.6931**
9) **Update Weights using the formula**

   For Incorrect Samples = W_{old}* e^{α} = (1/5)*$e^{0.6931}$ = (1/5)*1.9999 = 0.4
   For Correct Samples = W_{old}* $e^{-\alpha}$ = (1/5)*$e^{-0.6931}$ = (1/5)*(0.500) = 0.1

Points	Updated Weight	Normalized Weight
A	0.1	0.1/0.8=0.125
B	0.1	0.1/0.8=0.125
C	0.4	0.4/0.8=0.5
D	0.1	0.1/0.8=0.125
E	0.1	0.1/0.8=0.125
Total	0.8	1

10) Model generates Buckets(range values) based on Normalized weights as follows.

Points	Updated Weight	Normalized Weight	Buckets(Range)
A	0.1	0.1/0.8=0.125	0-0.125
B	0.1	0.1/0.8=0.125	0.125-0.250
C	0.4	0.4/0.8=0.5	0.250-0.750
D	0.1	0.1/0.8=0.125	0.750-0.875
E	0.1	0.4/0.8=0.125	0.875-1.0
Total Sum	0.8	1	

11) Model starts with new but empty dataset that is the same size as original, model then picks a random number between 0 and 1. This process is repeated n times where n is

the number of the records in the original dataset. The bucket into which that random number will fall will decide the row which will to be selected to be a part of the new dataset considered for 2nd stump. So in this way the chances are high for the misclassified data point to be a member of new dataset. This process is like Bootstrapping, so the misclassified record can be repeated as well. The new dataset is the same size as original.

12) So this way the sample which is misclassified, will have higher say in the 2nd stump and so on. That is how the errors that the first stump makes influence how the 2nd stump is made and so on.