AdaBoost

Boosting:

The basic idea underlying boosting techniques is that we build a model on the training dataset first, and then we build a second model to correct the faults in the first model. This process is repeated until the mistakes are reduced and the dataset is accurately predicted.

Let's look at an example: imagine you created a decision tree algorithm on the Titanic dataset and got an accuracy of 80%. Following that, you apply a new method and verify the accuracy, which comes out to be 75% for KNN and 70% for Linear Regression.

When we developed a separate model on the same dataset, we saw that the accuracy differed. But what if we combine all of these algorithms to come up with a final prediction? Taking the average of the outcomes from various models will give us more accurate results. In this method, we may improve forecast accuracy.

Introduction:

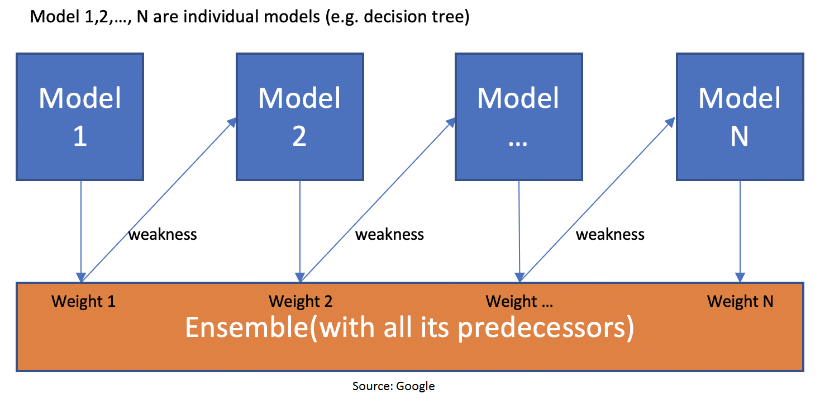
AdaBoost, also known as Adaptive Boosting, is a Machine Learning approach that is employed as part of an Ensemble Method. Decision trees with one level, or Decision trees with only one split, are the most popular algorithm used with AdaBoost. Decision Stumps is another name for these trees.

This approach creates a model by assigning equal weights to all of the data points. It then gives points that are incorrectly categorised a higher weight. In the next model, all points with greater weights are given more importance. It will continue to train models till a lowe error is received.

Understanding how the AdaBoost Algorithm works:

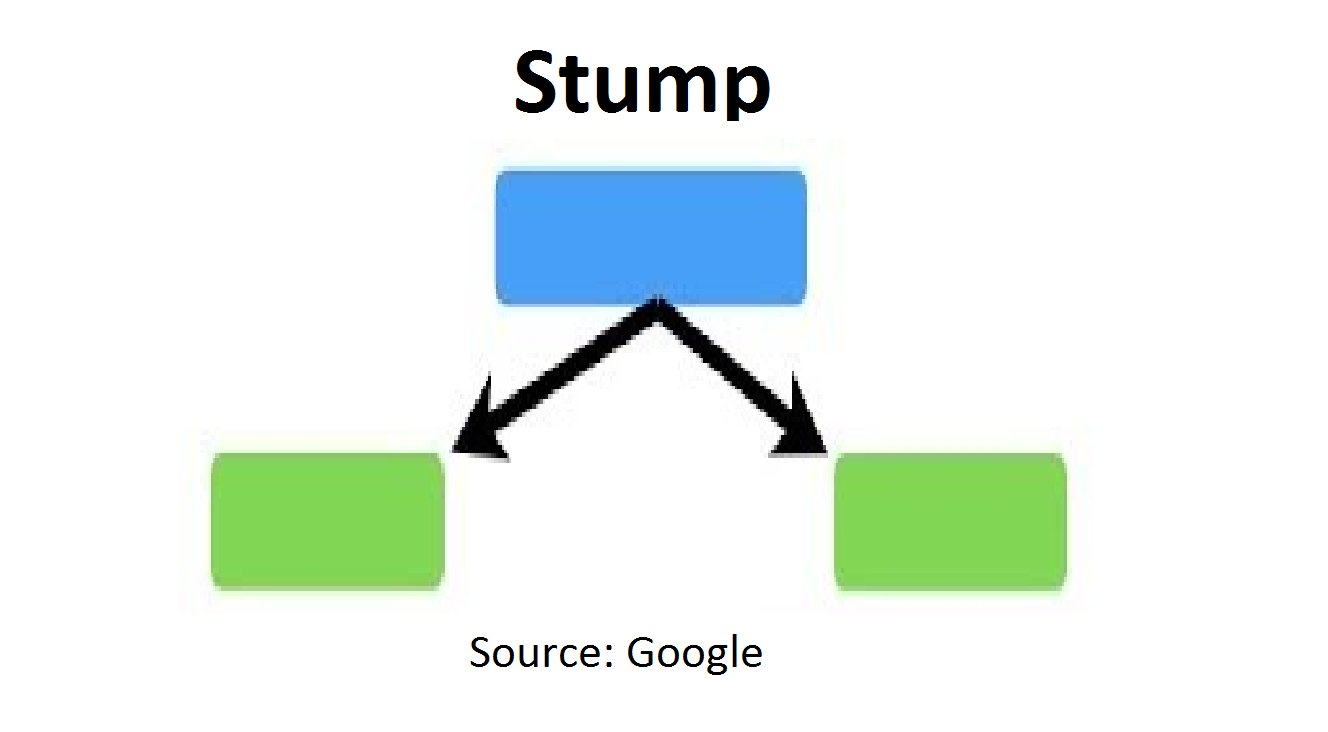
Let's start with an explanation of how boosting works. During the data training phase, it creates a certain number of decision trees. The improperly categorised record in the first model is given precedence when the first decision tree/model is constructed. Only these records are supplied to the second model as input. The procedure continues until we have decided on a number of base learners to develop. Remember that all boosting strategies allow for record repetition.

The AdaBoost algorithm will be simple to comprehend given that we grasp the boosting idea. Let's have a look at how AdaBoost works. When the random forest algorithm is applied, it creates a 'n' number of trees. It creates correct trees with a root node and numerous leaf nodes. Although some trees are larger than others, there is no set depth in a random forest. However, using AdaBoost, the method only creates a Stump node, which has two leaves.

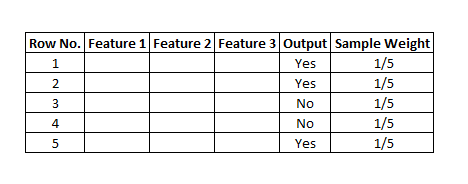


This figure shows how the first model is made and errors from the first model are noted by the algorithm. The record which is incorrectly classified is used as input for the next model. This process is repeated until the specified condition is met. As you can see in the figure, there are ‘n’ number of models made by taking the errors from the previous model. This is how boosting works. The models 1,2, 3,…, N are individual models that can be known as decision trees. All types of boosting models work on the same principle.

Since we now know the boosting principle, it will be easy to understand the AdaBoost algorithm. Let’s dive into AdaBoost’s working. When the random forest is used, the algorithm makes an ‘n’ number of trees. It makes proper trees that consist of a start node with several leaf nodes. Some trees might be bigger than others, but there is no fixed depth in a random forest. With AdaBoost, however, the algorithm only makes a node with two leaves, known as Stump.



The figure here represents the stump. It can be seen clearly that it has only one node with two leaves. These stumps are weak learners and boosting techniques prefer this. The order of stumps is very important in AdaBoost. The error of the first stump influences how other stumps are made. Let’s understand this with an example.



Dummy Dataset

Here’s a sample dataset consisting of only three features where the output is in categorical form. The image shows the actual representation of the dataset. As the output is in binary/categorical form, it becomes a classification problem. In real life, the dataset can have any number of records and features in it. Let us consider 5 datasets for explanation purposes. The output is in categorical form, here in the form of Yes or No. All these records will be assigned a sample weight. The formula used for this is ‘W=1/N’ where N is the number of records. In this dataset, there are only 5 records, so the sample weight becomes 1/5 initially. Every record gets the same weight. In this case, it’s 1/5.

How Does the Algorithm Decide Output for Test Data?

Suppose with the above dataset, the algorithm constructed 3 decision trees or stumps. The test dataset will pass through all the stumps which have been constructed by the algorithm. While passing through the 1st stump, the output it produces is 1. Passing through the 2nd stump, the output generated once again is 1. While passing through the 3rd stump it gives the output as 0. In the AdaBoost algorithm too, the majority of votes take place between the stumps, in the same way as in random trees. In this case, the final output will be 1. This is how the output with test data is decided.

How to Code AdaBoost in Python?

In Python, coding the AdaBoost algorithm takes only 3-4 lines and is easy. We must import the AdaBoost classifier from the sci-kit learn library. Before applying AdaBoost to any dataset, one should split the data into train and test. After splitting the data into train and test, the training data is ready to train the AdaBoost model. This data has both the input as well as output. After training the data, our algorithm will try to predict the result on the test data. Test data consists of only the inputs. The output of test data is not known by the model. Accuracy can be checked by comparing the actual output of the test data and the output predicted by the model. This can help us conclude how our model is performing and how much accuracy can be considered, depending on the problem statement. If it’s a medical problem, then accuracy should be above 90%. Usually, 70% accuracy is considered good. Accuracy also depends on factors apart from the type of model. The figure below shows the code used to implement AdaBoost.

Adaboost Algorithm

Adaptive Boosting is a good ensemble technique and can be used for both Classification and Regression problems. In most cases, it is used for classification problems. It is better than any other model as it improves model accuracy which can be checked by going in sequence. One can first try decision trees and then go for the random forest to finally apply the boost and implement AdaBoost. Accuracy keeps increasing as we follow the above sequence. The weight-assigning technique after every iteration makes the AdaBoost algorithm different from all other boosting algorithms and that is the best thing about it.