Boosting Algorithm

Boosting Algorithms can be used in regression as well as classification problems.

AdaBoost:

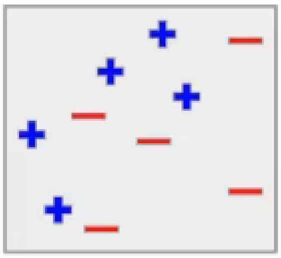
The Adaptive Boosting algorithm. It is more commonly known as AdaBoost. AdaBoost is the first designed boosting algorithm with a particular loss function.

AdaBoost regressor:

AdaBoost regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction.

AdaBoost Classification:

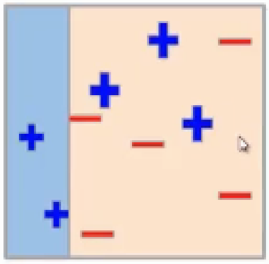
In the case of AdaBoost, higher points are assigned to the data points which are miss-classified or incorrectly predicted by the previous model. This means each successive model will get a weighted input.



I have the blue positives and red negatives. Now the first step is to build a model to classify this data.

Model 1:

Suppose the first model gives the following result, where it is able to classify two blue points on the left side and all red points correctly.

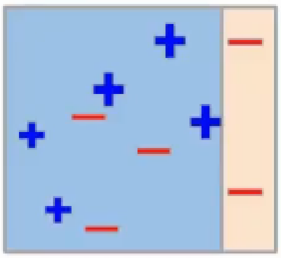


Model 1

But the model also miss-classifies the three blue points here.

Now, these miss- classified data points will be given higher weight. So these three blue positive points will be given higher weights in the next iteration. Giving higher weights to these points means my model is going to focus more on these values. Now we will build a new model.

Model 2:

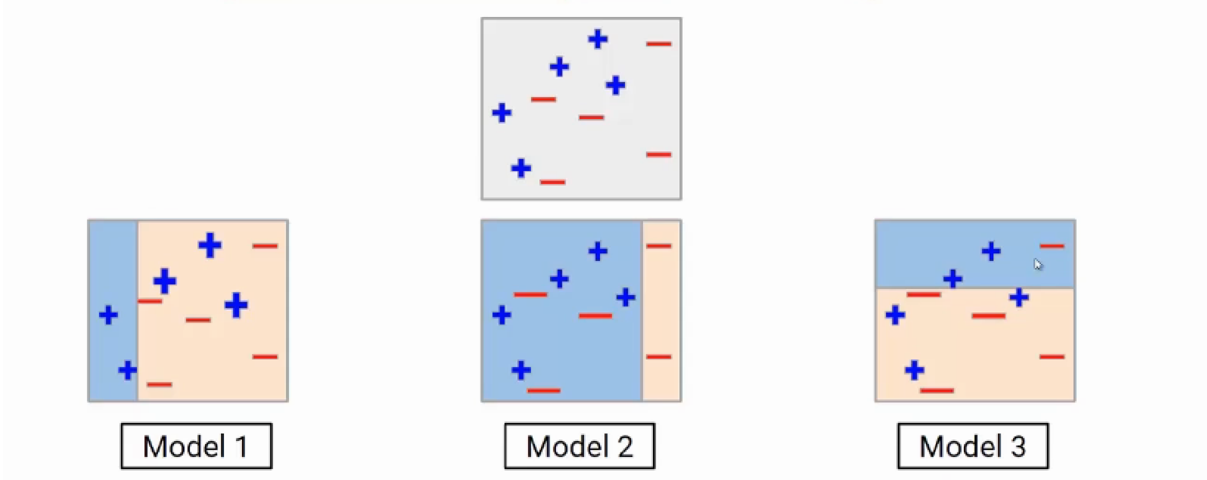


Model 2

In the second model you will see, the model boundary has been shifted to the right side in order to correctly classify the higher weighted points. Still, it's not a perfect model. You will notice three red negatives are miss-classified by model 2.

Now, these miss-classified red points will get a higher weight. Again we will build another model and do the predictions. The task of the third model is to focus on these three red negative points.

Model 3:



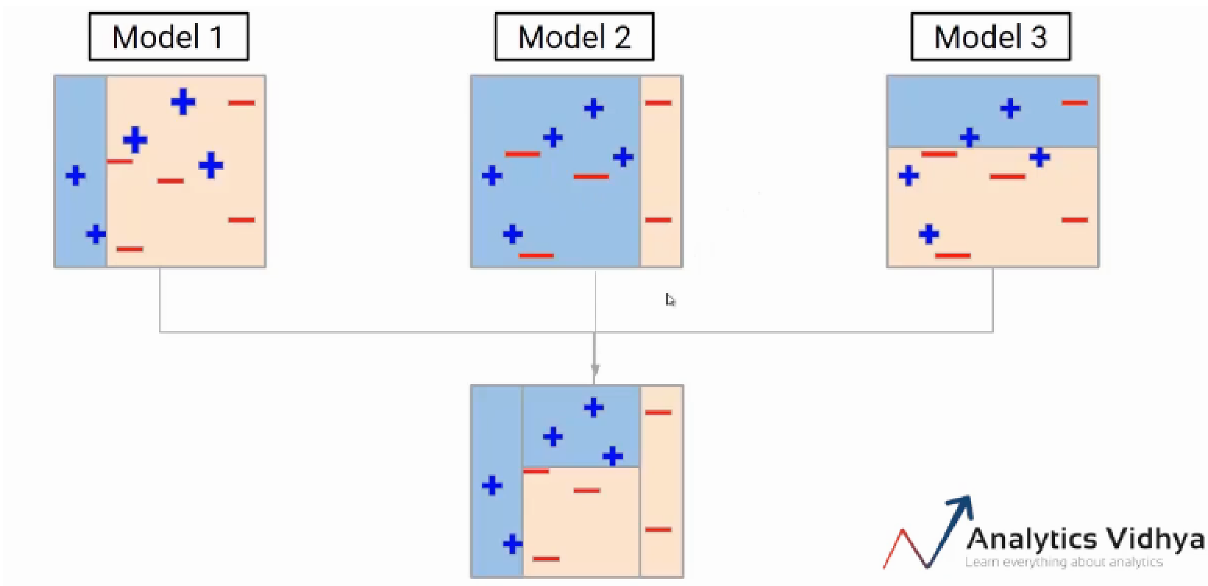
See, This new model again incorrectly predicted some data points. At this point, we can say all these individual models are not strong enough to classify the points correctly and are often called weak learners.

What should be our next step?

Easy to solve this by using "**Ensemble technique**".

Ensemble:

Well, we have to aggregate these models(M1,M2,M3). One of the ways could be taking the weighted average of the individual weak learners. So our final model will be the weighted mean of individual models.



After multiple iterations, we will be able to create the right decision boundary with the help of all the previous weak learners. As you can see the final model is able to classify all the points correctly. This final model is known as a strong learner.

Gradient Boosting Machine (GBM)

A Gradient Boosting Machine or GBM combines the predictions from multiple decision trees to generate the final predictions. Keep in mind that all the weak learners in a gradient boosting machine are decision trees.

Extreme Gradient Boosting Machine (XGBM)

Extreme Gradient Boosting or XGBoost is another popular boosting algorithm. In fact, XGBoost is simply an improvised version of the GBM algorithm! The working procedure of XGBoost is the same as GBM. The trees in XGBoost are built sequentially, trying to correct the errors of the previous trees.

- One of the most important points is that XGBM implements parallel preprocessing (at the node level) which makes it faster than GBM
- XGBoost also includes a variety of regularization techniques that reduce overfitting and improve overall performance. You can select the regularization technique by setting the hyperparameters of the XGBoost algorithm

Regularization: XGBoost has an option to penalize complex models through both L1 and L2 regularization. Regularization helps in preventing overfitting.

If you are using the XGBM algorithm, you don't have to worry about imputing missing values in your dataset. The XGBM model can handle the missing values on its own. During the training process, the model learns whether missing values should be in the right or left node.

The Power of XGBoost:

The beauty of this powerful algorithm lies in its scalability, which drives fast learning through parallel and distributed computing and offers efficient memory usage.

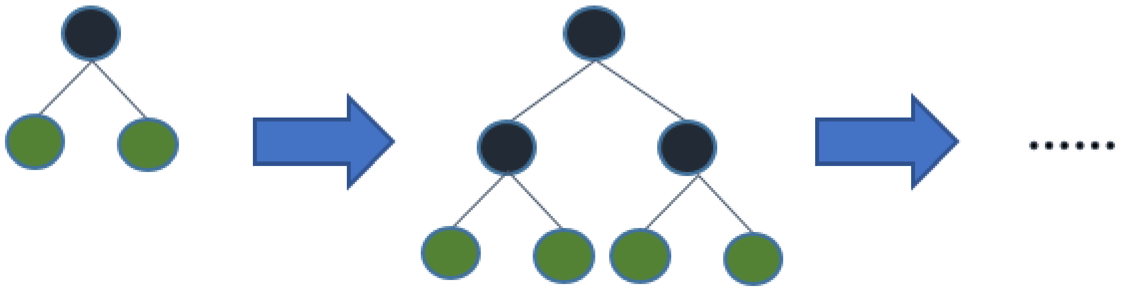
Advantages of XGBoost:

- ❖ Regularization
- ❖ Parallel Processing
- ❖ Handling Missing Values
- ❖ Cross Validation
- ❖ Effective Tree Pruning

LightGBM

The LightGBM boosting algorithm is becoming more popular by the day due to its speed and efficiency. LightGBM is able to handle huge amounts of data with ease. But keep in mind that this algorithm does not perform well with a small number of data points.

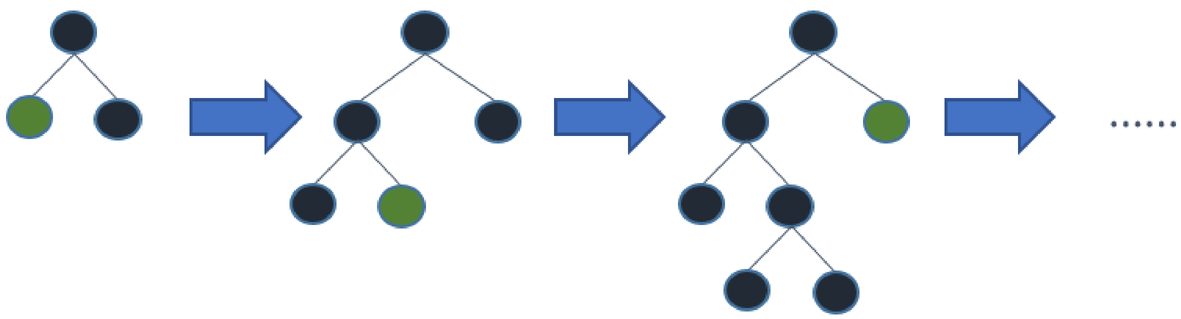
Since it is based on decision tree algorithms, it splits the tree leaf wise with the best fit whereas other boosting algorithms split the tree depth wise or level wise rather than leaf-wise.



Level-wise tree growth

Leaf wise splits lead to increase in complexity and may lead to overfitting and it can be overcome by specifying another parameter max-depth which specifies the depth to which splitting will occur.

So when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms. Also, it is surprisingly very fast, hence the word 'Light'.



Leaf-wise tree growth

Advantages of Light GBM

- Faster training speed and higher efficiency.
- Lower memory usage.
- Better accuracy than any other boosting algorithm.
- Compatibility with Large Datasets.
- Parallel learning supported.