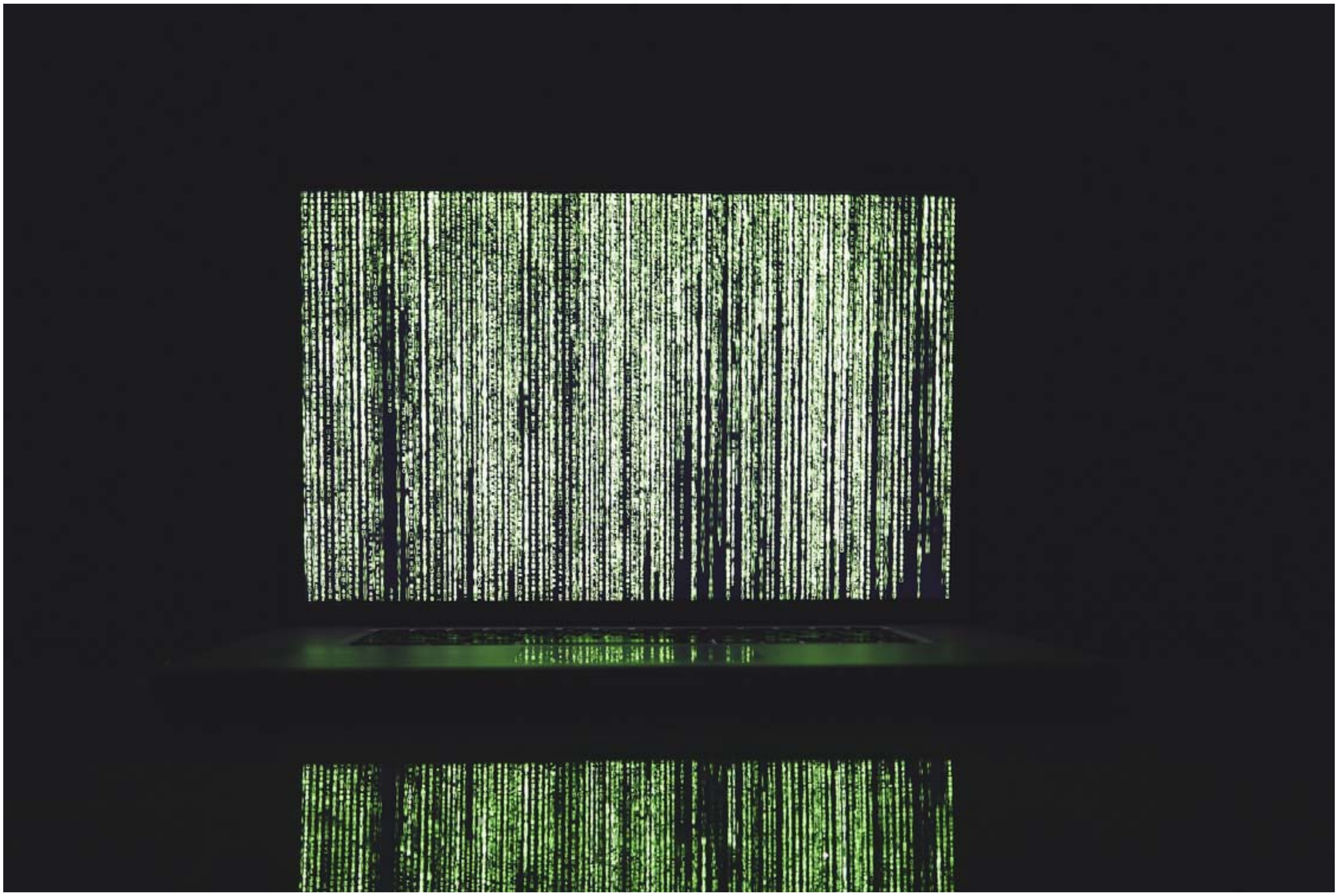
Rohith Gandhi Follow
What I cannot create, I do not understand - Richard Feynman. LinkedIn: www.linkedin.com/in/grohith327
May 6 · 5 min read

Boosting Algorithms: AdaBoost, Gradient Boosting and XGBoost

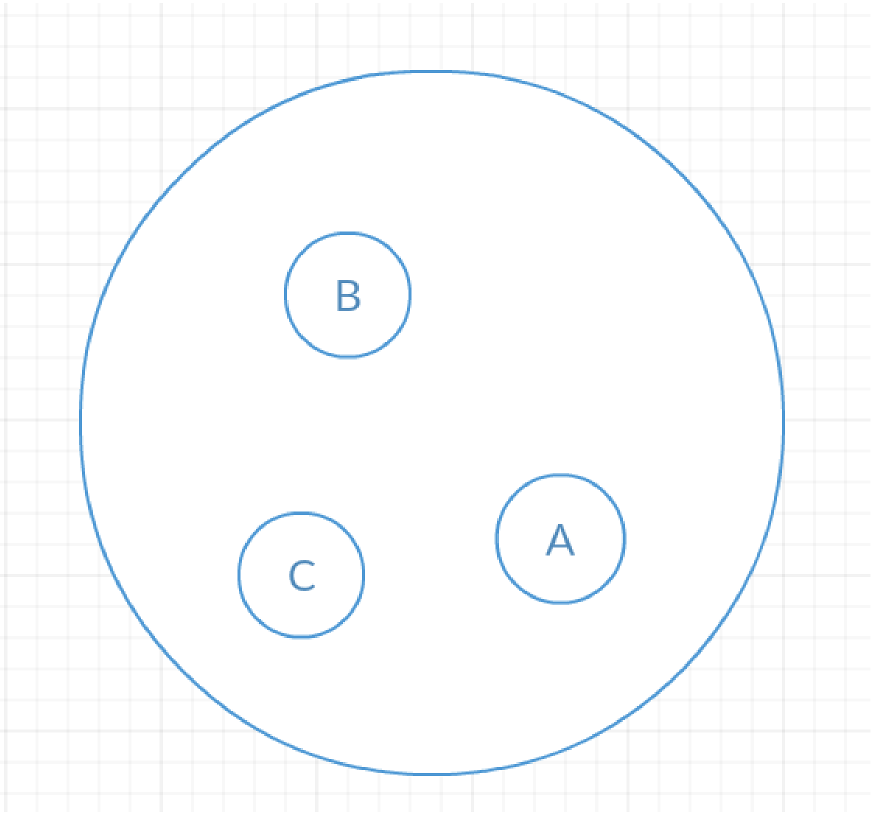


- Neural networks and Genetic algorithms are our naive approach to imitate nature. They work well for a class of problems but they do have various hurdles such as overfitting, local minima, vanishing gradient and much more. There is another set of algorithms that do not get much recognition(in my opinion) compared to others and they are boosting algorithms.

What is Boosting?

Boosting is a method of converting a set of weak learners into strong learners. Suppose we have a binary classification task. A weak learner has an error rate that is slightly lesser than 0.5 in classifying the object, i.e the weak learner is slightly better than deciding from a coin toss. A strong learner has an error rate closer to 0. To convert a weak learner into strong learner, we take a family of weak learners, combine them and vote. This turns this family of weak learners into strong learners.

The idea here is that the family of weak learners should have a minimum correlation between them.



Here, let A, B and C be different classifiers. Their area A represents where the classifier A misclassifies(goes wrong) and area B represents where the classifier B misclassifies and area C represents where the classifier C misclassifies. Since there is no correlation between the errors of each classifier, combining them and using a technique of democratic voting to classify each object, this family of classifiers will never go wrong. I guess this would've provided a basic understanding of boosting. Moving on to types of boosting algorithm.

Types of boosting algorithms:

I would like to explain different boosting algorithms without any of the math involved in it cause I feel it would complicate things and defeat the purpose of this article, which is simplicity(hopefully). The different types of boosting algorithms are:

AdaBoost

Gradient Boosting

XGBoost

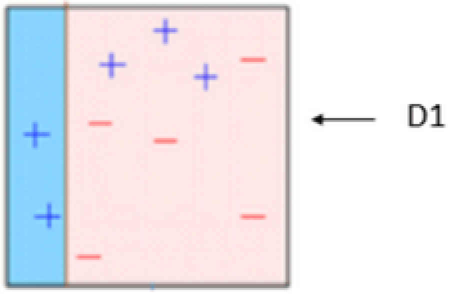
These three algorithms have gained huge popularity, especially XGBoost, which has been responsible for winning many data science competitions.

AdaBoost(Adaptive Boosting):

The Adaptive Boosting technique was formulated by Yoav Freund and Robert Schapire, who won the Gödel Prize for their work. AdaBoost works on improving the areas where the base learner fails. The base learner is a machine learning algorithm which is a weak learner and upon which the boosting method is applied to turn it into a strong learner. Any machine learning algorithm that accept weights on training data can be used as a base learner. In the example taken below, Decision stumps are used as the base learner.

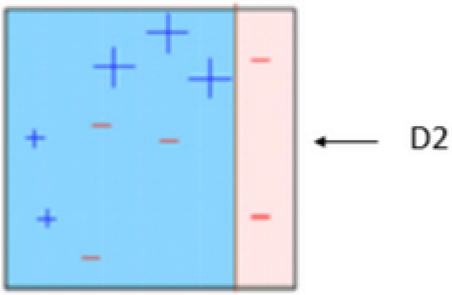
We take the training data and randomly sample points from this data and apply decision stump algorithm to classify the points. After classifying the sampled points we fit the decision tree stump to the complete training data. This process iteratively happens until the complete training data fits without any error or until a specified maximum number of estimators.

After sampling from training data and applying the decision stump, the model fits as showcased below.



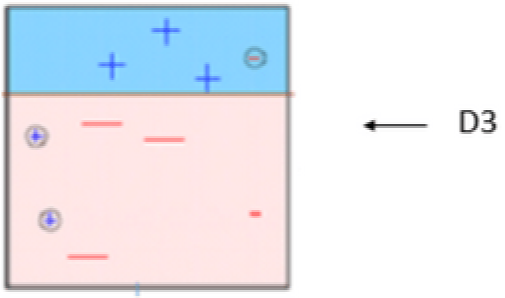
Decision Stump 1

We can observe that three of the positive samples are misclassified as negative. Therefore, we exaggerate the weights of these misclassified samples so that that they have a better chance of being selected when sampled again.



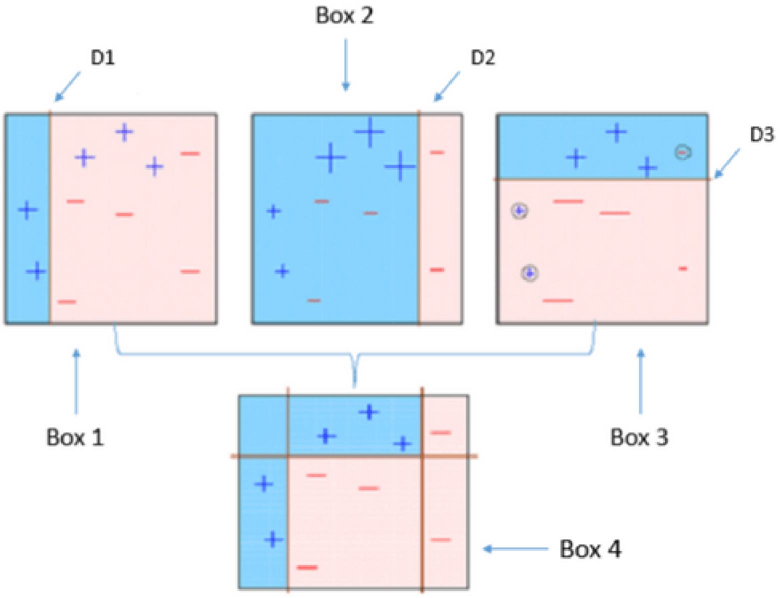
Decision Stump 2

When data is sampled next time, the decision stump 2 is combined with decision stump 1 to fit the training data. Therefore we have a miniature ensemble here trying to fit the data perfectly. This miniature ensemble of two decision stumps misclassifies three negative samples as positive. Therefore, we exaggerate the weights of these misclassified samples so that that they have a better chance of being selected when sampled again.



Decision Stump 3

The previously misclassified samples are chosen and decision stump 3 is applied to fit the training data. We can find that two positive samples are classified as negative and one negative sample is classified as positive. Then the ensemble of three decision stumps(1, 2 and 3) are used to fit the complete training data. When this ensemble of three decision stumps are used the model fits the training data perfectly.



Ensemble of 3 Decision Stumps

The drawback of AdaBoost is that it is easily defeated by noisy data, the efficiency of the algorithm is highly affected by outliers as the algorithm tries to fit every point perfectly. You might be wondering since the algorithm tries to fit every point, doesn't it overfit? No, it does not. The answer has been found out through experimental results, there has been speculations but no concrete reasoning available.

Code:

```
# AdaBoost Algorithm
from sklearn.ensemble import AdaBoostClassifier


clf = AdaBoostClassifier()
# n_estimators = 50 (default value)
# base_estimator = DecisionTreeClassifier (default value)
clf.fit(x_train,y_train)
clf.predict(x_test)
```

Continuing to explain Gradient Boosting and XGBoost will further increase the length of this already pretty long article. Therefore I have decided to write them as another article. Please follow the link below.

Link: https://medium.com/@grohith327/gradient-boosting-and-xgboost-90862daa6c77

References:

https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/