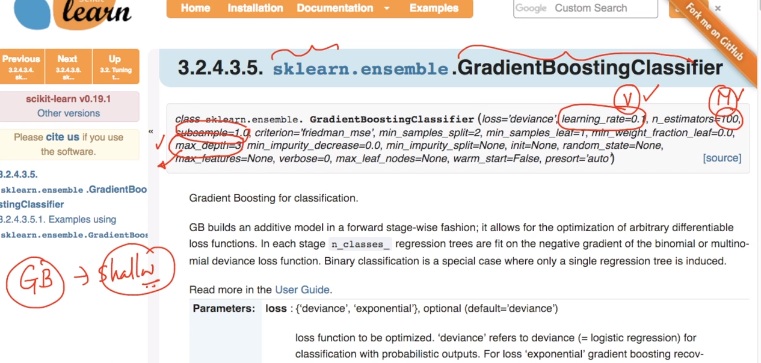
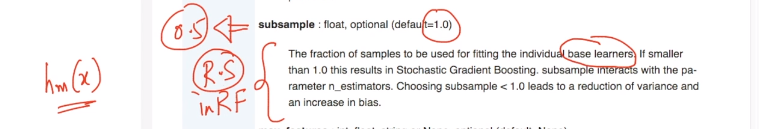
So Gradient Boosting Classifiers can be found very easily in sklearn and when we talk about its hyper params it has few like learning rate i.e. v , n\_estimators i.e. M, or say max\_depth can also be seen as hyper params and to find all of this hyper params we will use grid search to find its optimal value.

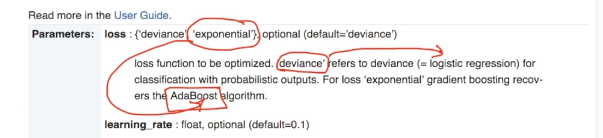


It has another parameter i.e. sub-sample



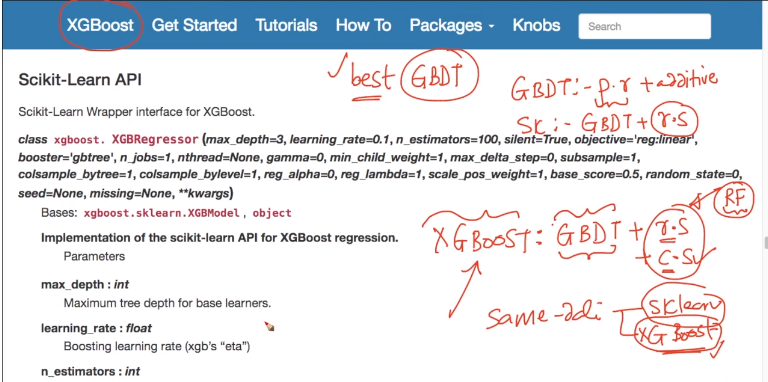
Sub-sample can be seen as row sampling in RF because when we say sub-sample = 0.5 so it wont use whole data in calculating hm(x) and use partial data which is just like Row sampling.

Another parameter is loss which is much similar to RF or others as we have seen before.



If you say loss as deviance so it means it is just **logisitic regression** and if you say it as exponential it means as **AdaBoost**.

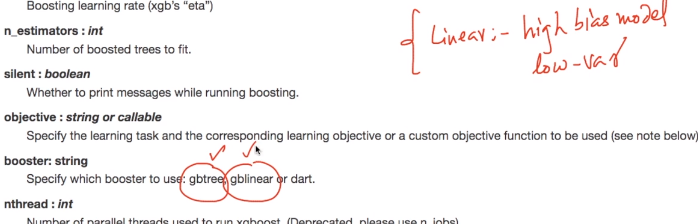
But sklearn implementation for GBDT is quite slow and much more enhanced implementation of GBDT which is better and publically available is **XGBoost**.



It basically uses GBDT + RS like sklearn implementation but additionally is also uses Column Sampling.

i.e. it is process combined of **GBDT + RS + CS**.

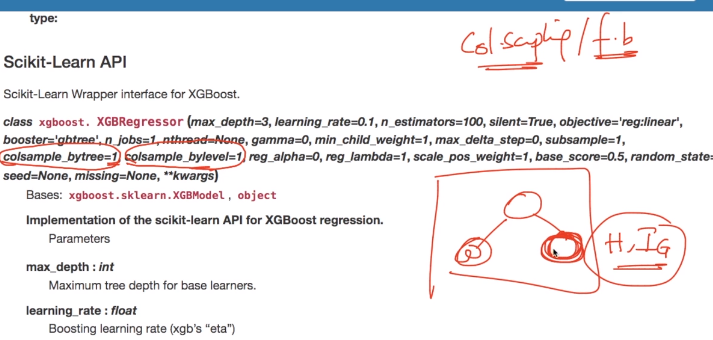
And so it performs much better than sklearn implementation.



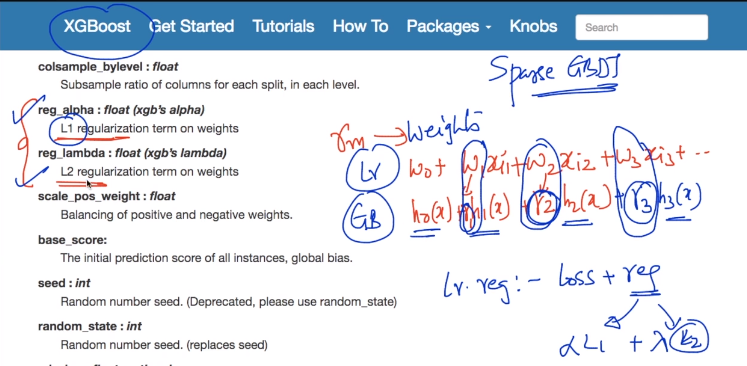
It allows you to choose between tree and linear models.

It has another parameters like colsample\_bytree and colsample\_bylevel

i.e. it allows you to use column sampling at both level i.e. for constructing tree and by level means at different levels of tree also to calculate Gine impurity and Entropy it allows column sampling.



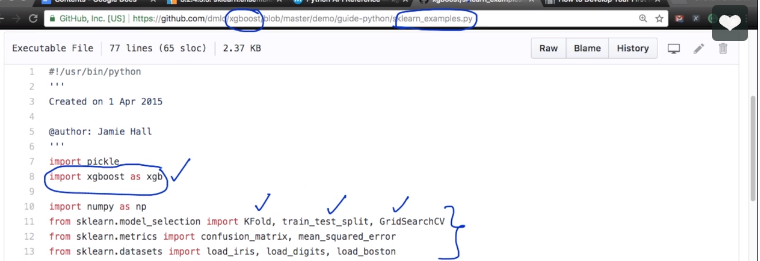
Another params like reg\_alpha and reg\_lambda are also very important parameters here.



So as in GBDT we have multiple Gamma values and so we can use regularization here to control its value and if we want sparse GBDT so as to reduce latency or anything like that we can also do that so that is the beauty of XGBoost implementation of GBDT.

XGBoost can also be interfaced with sklearn library.

There are just few minor changes.



**Comments:**

