Boosting

It is an effective method to deal with UNDERFITTING! 🡺 Boost the scores!

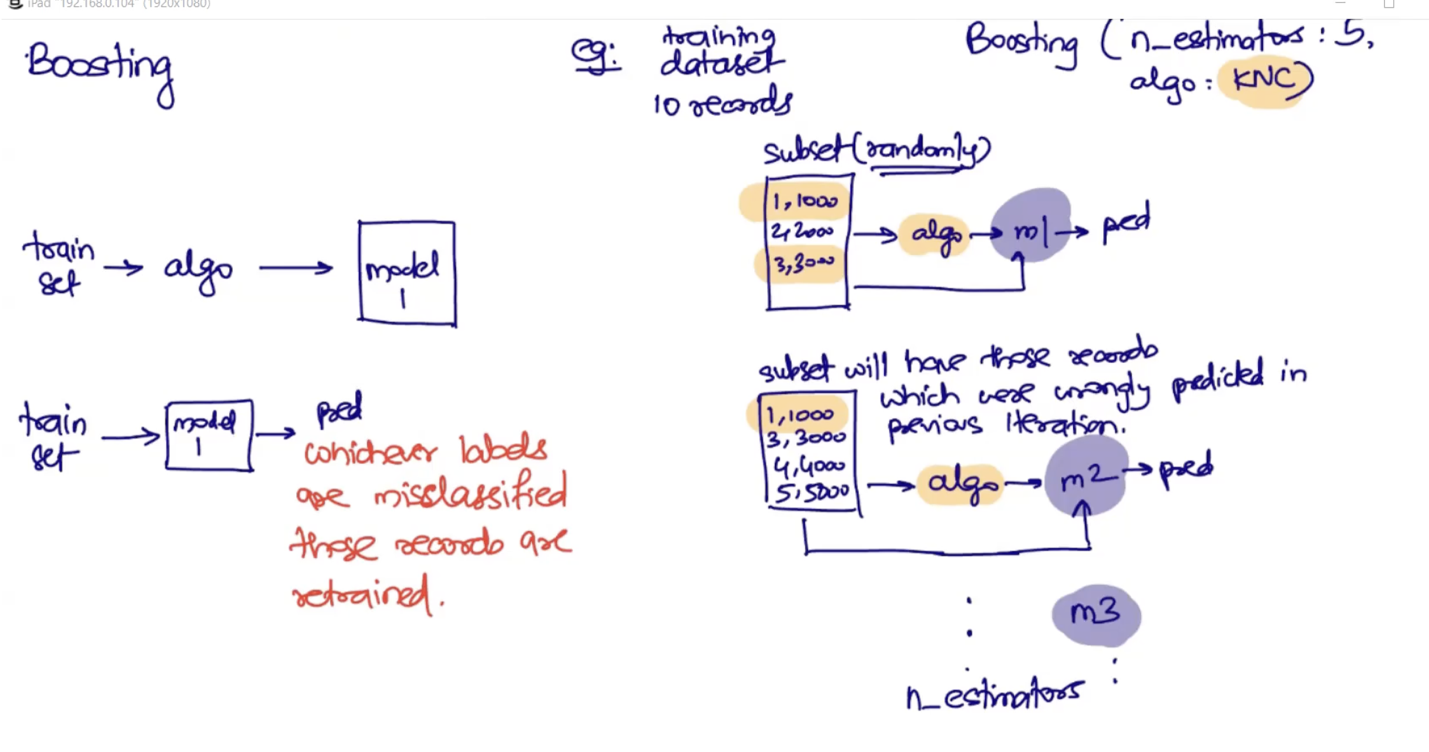
Boosting also create set of weak learners but the approach is different.

Train Set 🡺 algorithm 🡺 model 1

Train set 🡺 model 1 🡺 prediction. whichever labels are misclassified those records are retrained.

Boosting creating a stronger model with the help of weak learners.

Bagging all weak learners participate to create a stronger prediction.



In Boosting the weak learners are responsible to create a strong learner

Boosting may introduce overfitting issues which can be solved by involving Bagging.

**XGBoost**

XGBoost can be used for both classification and regression

Classification : XGBClassifier

Regression: XGBRegressor

This classification/regression uses Decision Tree as the core algo to create mini models

We all know Decision Tree can overfit the model, thus you can control the overfitting

behavior using learning\_rate hyperparameter

Ideal range of learning\_rate is between 0 to 1

Suggest to start learning\_rate with value 0.001 and so on....

XGBOOST requires seperate installation

pip install xgboost

IF you use XGBoost and you don’t get a generalized model, try use learning\_ rate parameter with it: 🡺 XGBClassifier(learning\_rate=0.001)

If XGBoost doesn’t work with learning\_rate, you can use: 🡺 XGBRFClassifier

When we use XGBoost, it internally use Decision Tree

When we use XGBRFClassifier, it internally use Random Forest.

**Stacking**

Using Boosting and Bagging at the same time to address overfitting and underfitting problem at the same time.

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingClassifier.html