# Topic: AdaBoost- Extreme Gradient Boosting

**Problem Statement: -**

A sample of global companies and their ratings are given for the cocoa bean production along with the location of bean being used by the companies. Identify the important features in the analysis and accurately classify the companies based on their ratings and draw insights from your model. Perform Ensemble methodology such as Bagging, Boosting, Stacking, voting algorithms on the dataset given.

Business objective:

To classify type of cocoa bean by ensemble models .

Solution:

Load the Coco\_Rating\_Ensemble.xlsx data file which contains all the data about .

Exploring Data

After you have loaded the dataset, you might want to know a little bit more about it. You can check feature and target names.



inspection of data types of each column

#converting categorical columns to numeric by using label encoder . Label Encoder is obtained importing from sklearn.preprocessing

converting class variable column to numeric by label encoder

By using unique( ) and values\_count() methods all unique entries and their count are measured in target data



#### Splitting Data

To understand model performance, dividing the dataset into a training set and a test set is a good strategy.

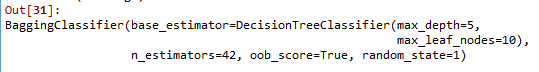
Let's split dataset by using function train\_test\_split(). You need to pass 3 parameters features, target, and test\_set size. Additionally, you can use random\_state to select records randomly.

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(predictors,target,test\_size=0.2)

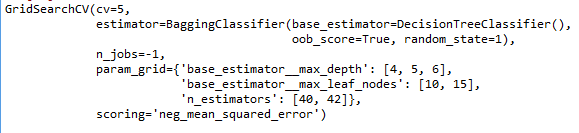
## Bagging technique:

Bagging technique is implemented by importing Bagging Classifier which is imported from sklearn.ensemble



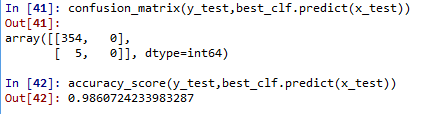
in which base estimator taken here as DecisionTreeClassifier which is imported from sklearn.ensemble

Hyper parameters tuning is done by using GridSearchCV which cross validates all hyper parameters and give best parameters



by passing obtained parameters from grid search to bagging classifier and data is trained.

confusion matrix and accuracy score for test data



so it considered as good model

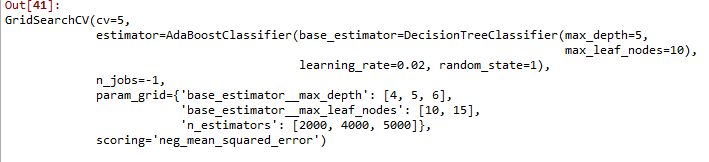
## Ada Boosting technique:

Bagging technique is implemented by importing AdaBoostClassifier which is imported from sklearn.ensemble



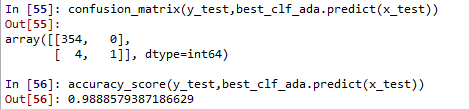
in which base estimator taken here as DecisionTreeClassifier which is imported from sklearn.ensemble

Hyper parameters tuning is done by using GridSearchCV which cross validates all hyper parameters and give best parameters



by passing obtained parameters from grid search to ada boosting classifier and data is trained.

confusion matrix and accuracy score for test data



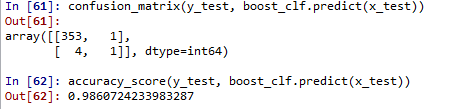
so the model is considered as good

## Gradient Boosting technique:

Bagging technique is implemented by importing GradientBoostingClassifier which is imported from sklearn.ensemble

boost\_clf = GradientBoostingClassifier()

confusion matrix and accuracy score for test data



## Stacking classification:

All the ensemble learners bagging ,ada boost,gradient boosting are listed in a base\_learners variable

for cross validation kfold technique is used and meta data is framed



each classifier accuracy scores are stacked

