BOOSTING ALGORITHMS

Saranya S

AGENDA

Types

Ada Booster Algorithm

XG Boosting Algorithm

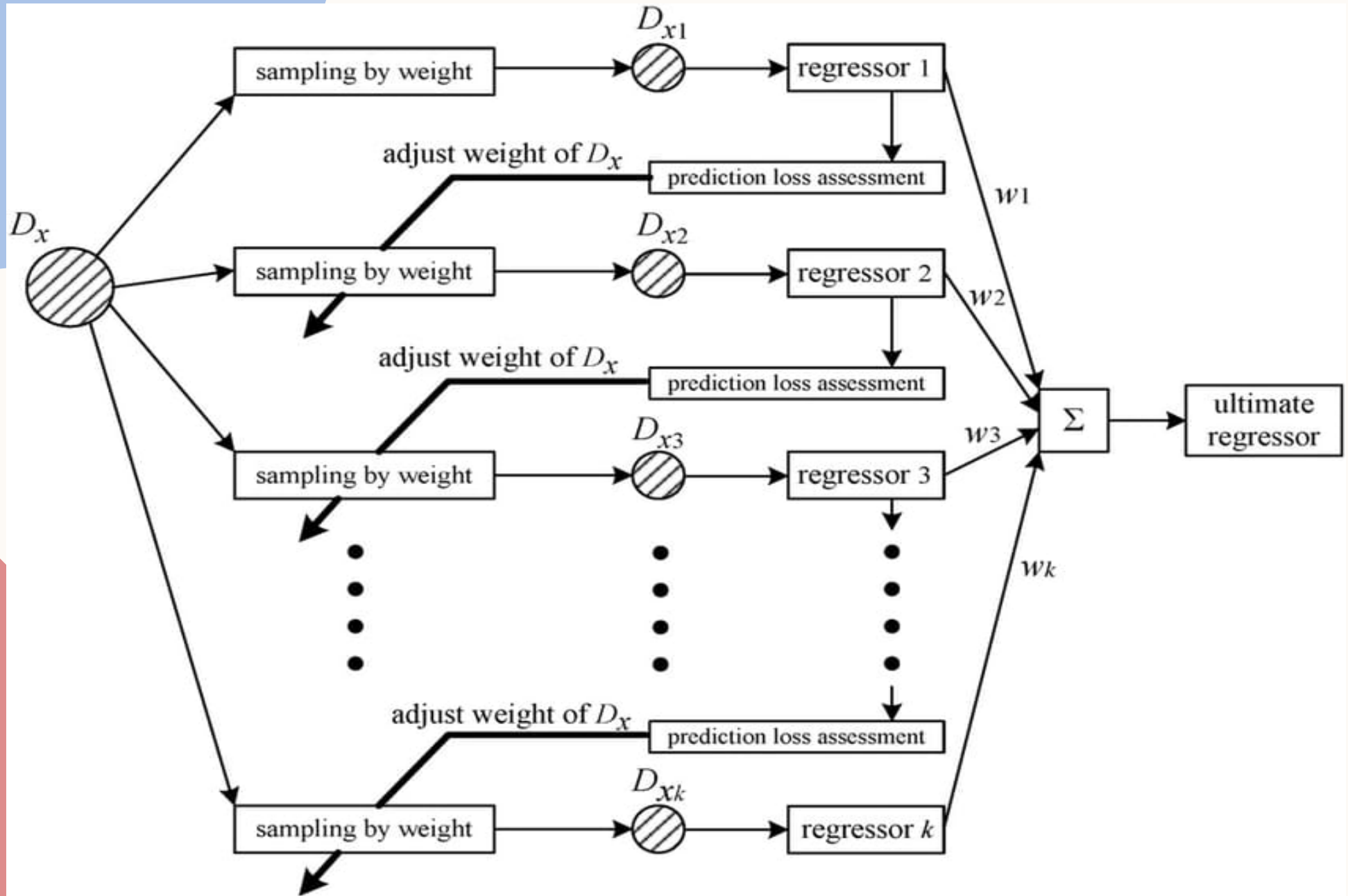
LG Boosting Algorithm

TYPES OF BOOSTING ALGORITHMS

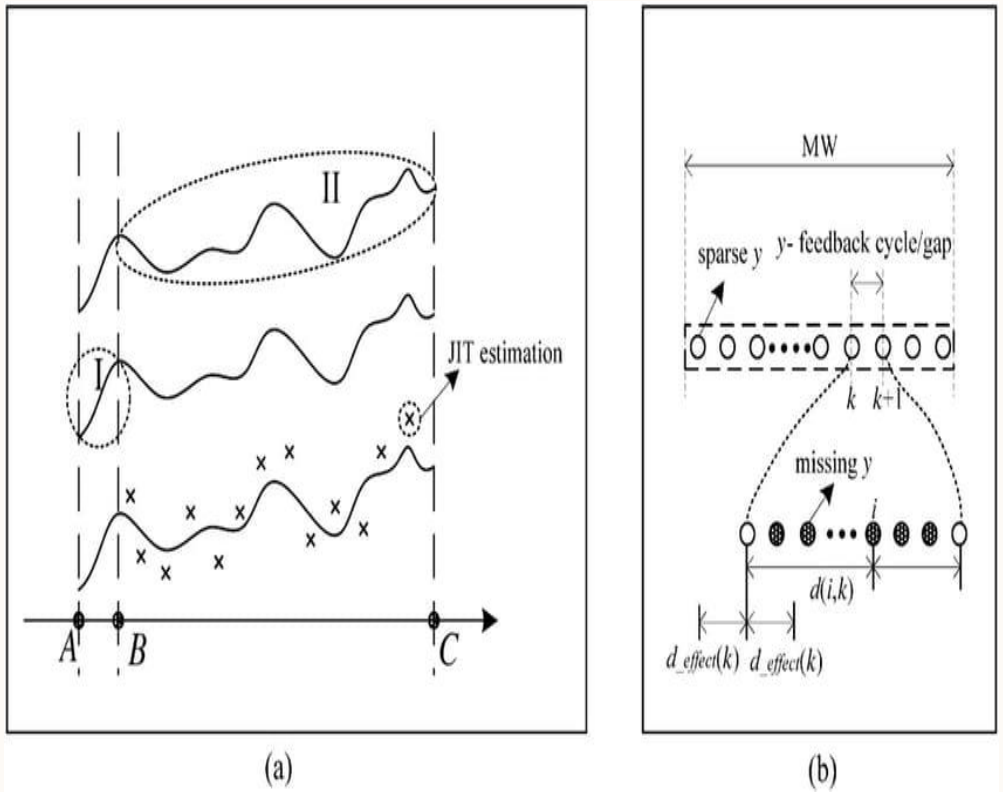
- Ada Booster Algorithm
- XG(Gradient Boosting)Algorithm
- LG(Light gain Boosting Algorithm
- Hist Gradient Algorithm
- Catboost Algorithm

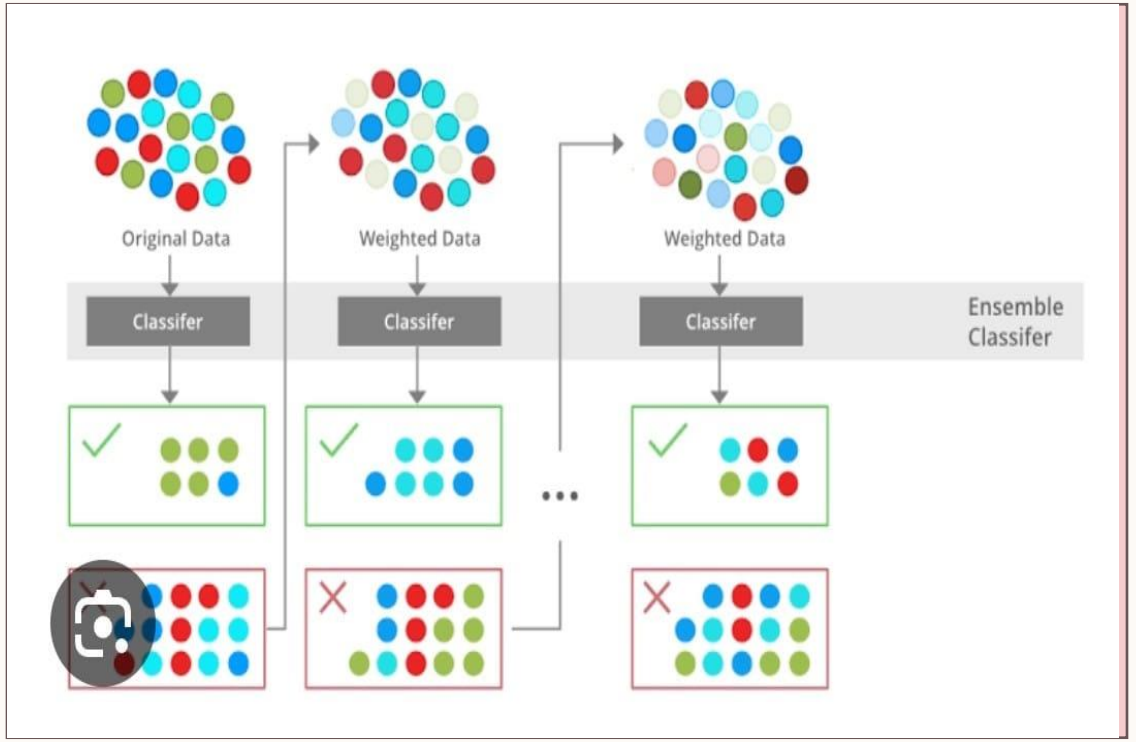
ADA BOOSTER ALGORITHM

- It is a meta-estimator that fits a regressor on the original dataset
- It works on the principle of stagewise addition method
- Multiple weak learners are used for getting strong learners
- The alpha parameter calculated is related to the error of the weak learners


D_x
sampling by weight
D_{x1}
regressor 1
adjust weight of D_x
prediction loss assessment
$w1$
sampling by weight
D_{x2}
regressor 2
$w2$
adjust weight of D_x
prediction loss assessment
sampling by weight
D_{x3}
regressor 3
$w3$
Σ
ultimate regressor
w_k
adjust weight of D_x
prediction loss assessment
sampling by weight
D_{xk}
regressor k

ADA BOOST ALGORITHM

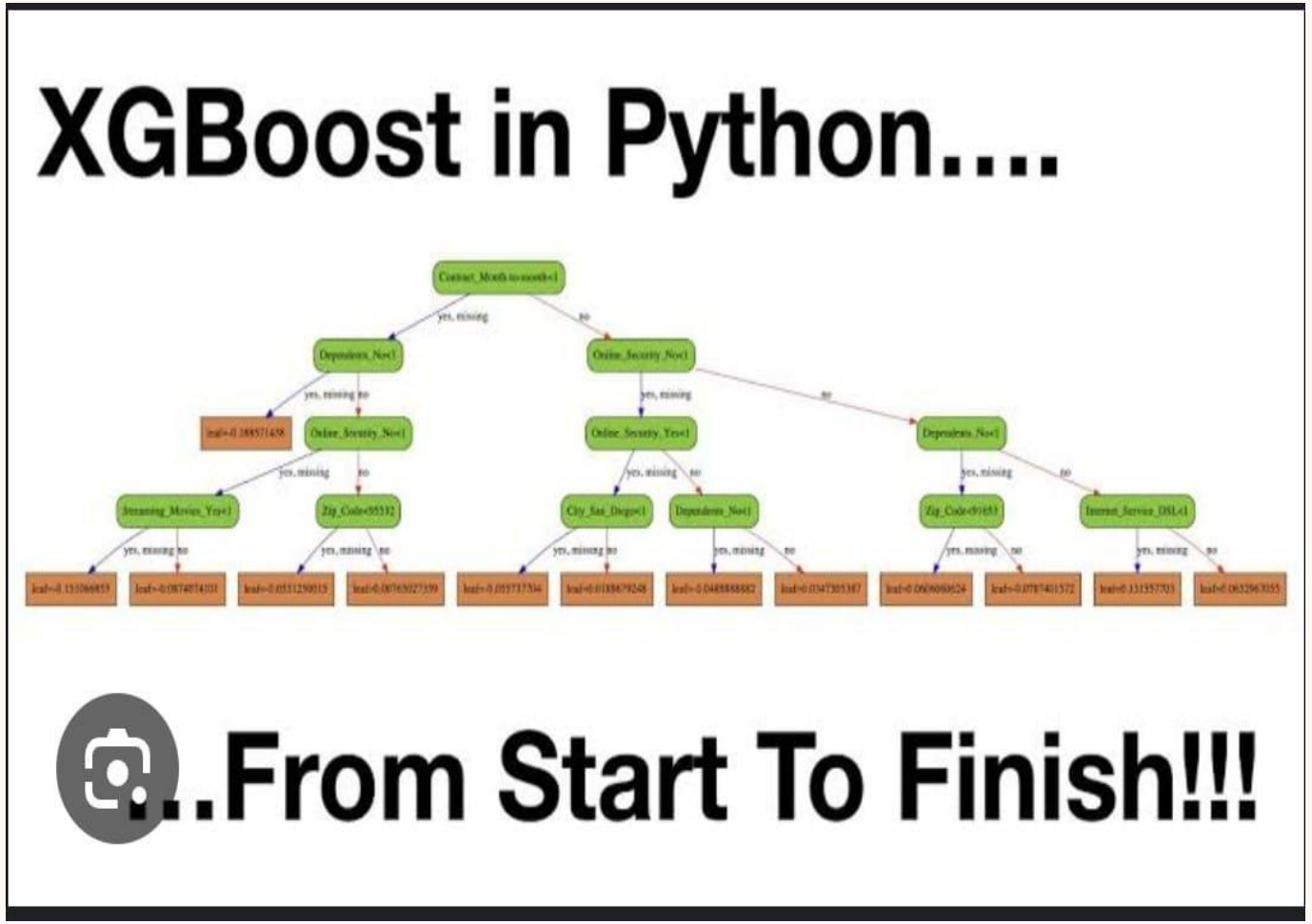


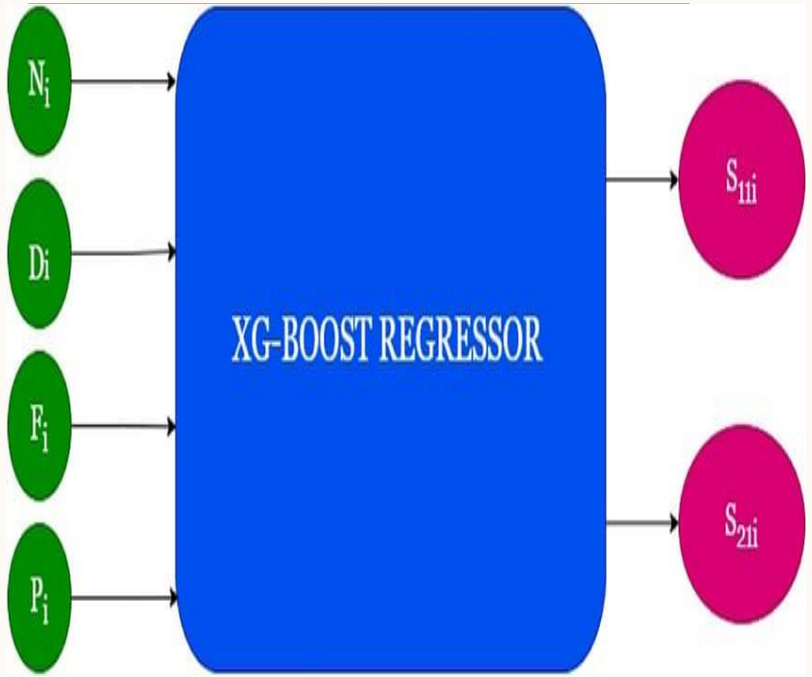


XG BOOSTING ALGORITHM

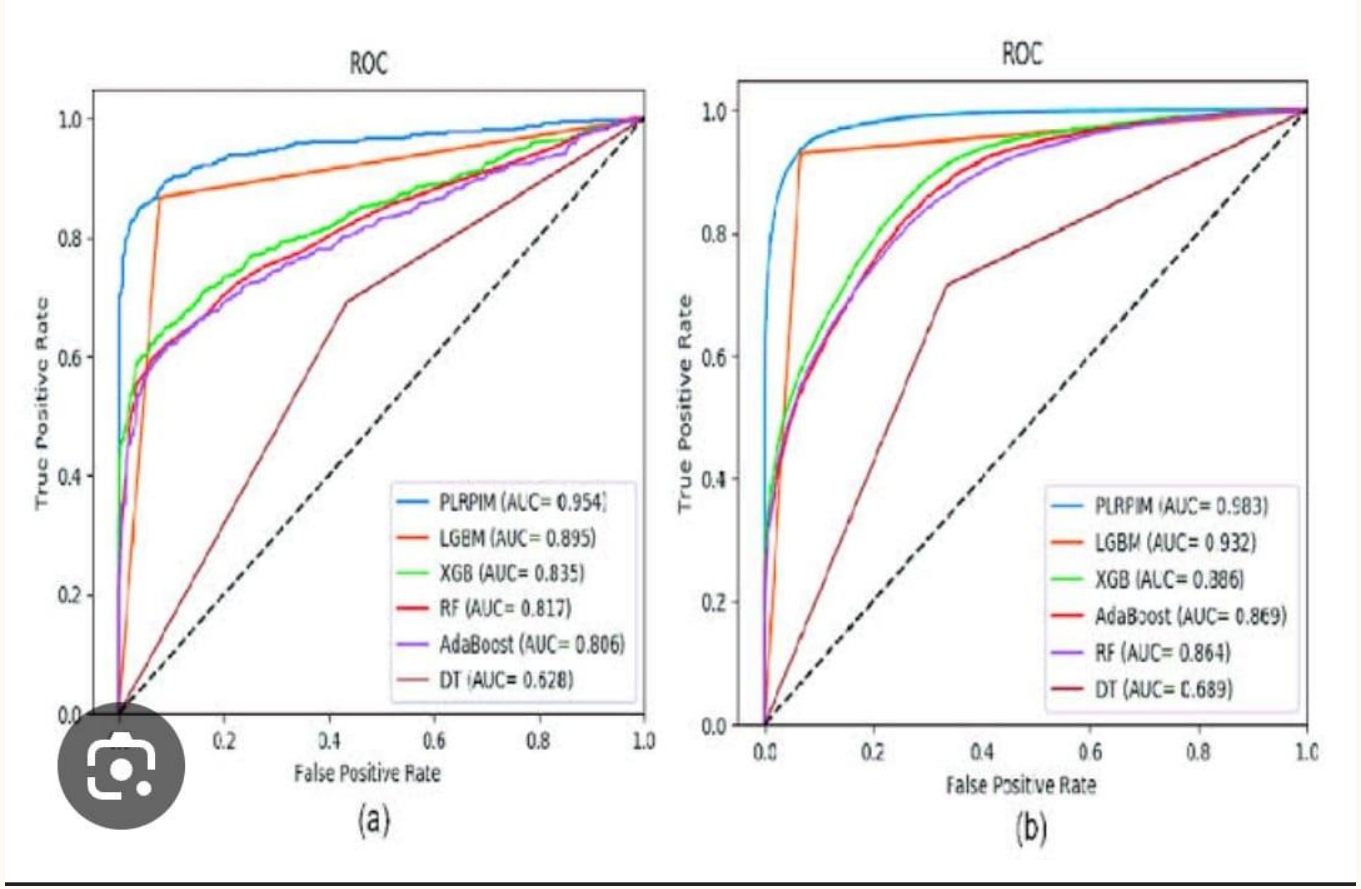
- eXtreme Gradient Boosting Algorithn
- Regularization Approach
- Xgboost outperforms a standard gradient boosting method, it also faster than previous boosting algorithm
- It works better when the dataset contains both numerical and categorical variables

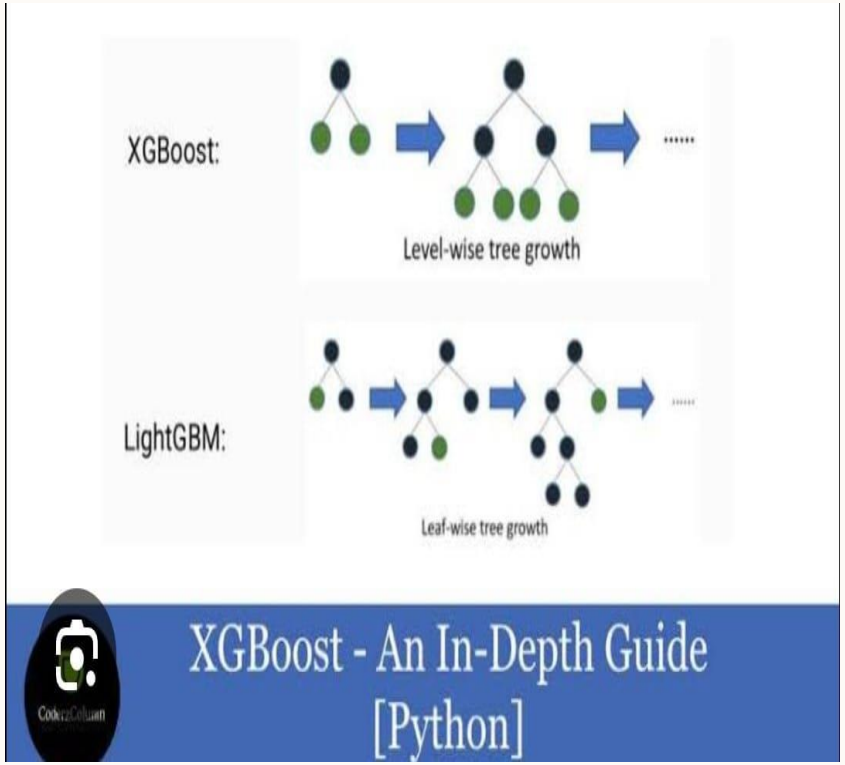
XG BOOSTING ALGORITHM





XG BOOSTING ALGORITHM





LG BOOSTING ALGORITHM

- Light gradient-boosting machine.
- It is a gradient boosting ensemble method
- Using AutoML and is based on decision trees.
- Histogram-based split finding and leaf-wise tree gro
- Faster and more efficient than traditional gradient boosting algorithm

LG BOOSTING ALGORITHM

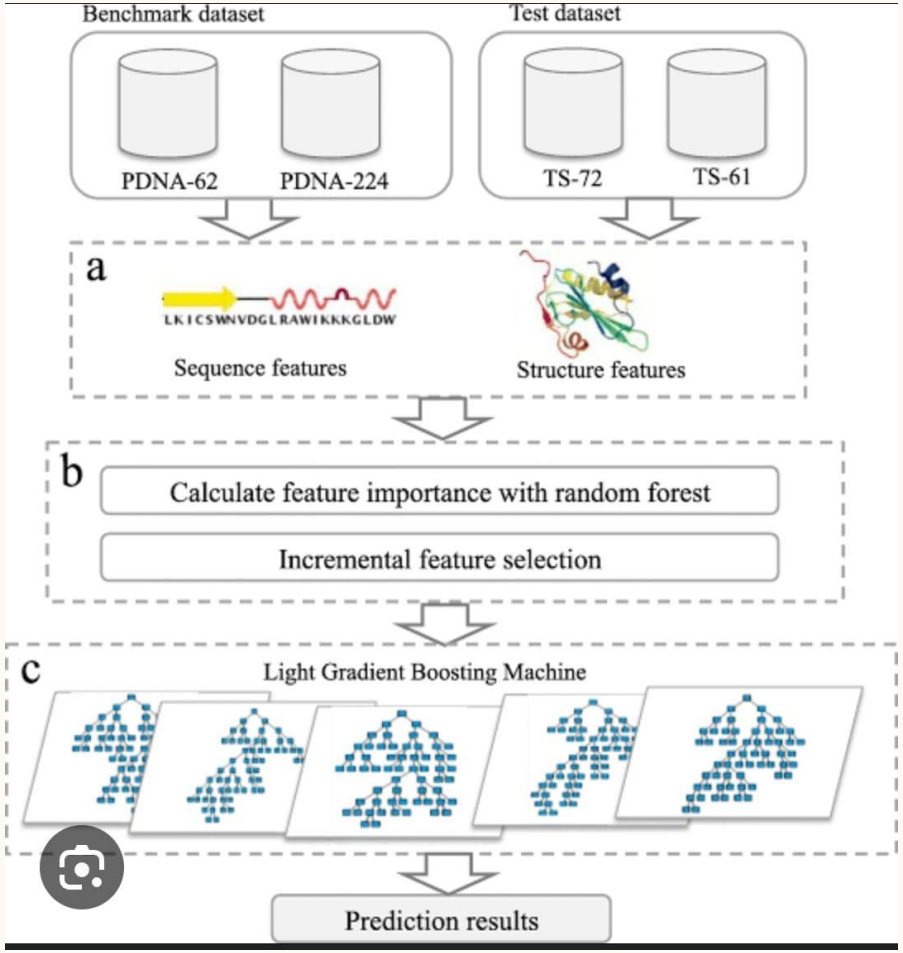
The LightGBM algorithm
Input:
Training data: $T = \{(\chi_1, y_1), (\chi_2, y_2)..., (\chi_N, y_N)\}$, $\chi_i \in \chi$, $\chi \subseteq R$,
$y_i \in \{-1,+1\}$; loss function: $L(y, \theta(\chi))$;
Iterations: M;
Ratio of sampling the Big gradient data: f; slight gradient data sampling ratio: z;
1: Combine features that are mutually exclusive (i.e., features not simultaneously accept nonzero numbers) of χ_i, i = {1...,N} by the exclusive feature bundling (EFB) technique;
2: Set $\theta_0(\chi) = arg\ min_c \sum_i^N L(y_i, c)$;
3: For m = 1 to M do
4: Calculate gradient absolute values:

$$r_i = \left|\frac{\partial L(y_i, \theta(\chi_i))}{\partial \theta(\chi_i)}\right|_{\theta(x)=\theta_{m-1}(\chi)}, i = \{1, \ldots, N\}$$

5: Resample dataset using gradient-based one-side sampling process:
$highN = f \times len(T)$; $randN = z \times len(T)$;
$sorted = GetSortedIndices(abs(r))$;
$F = sorted[\ 1: highN]$; $Z = RandomPick(sorted[\ highN: len(T)], randN)$;
$\acute{T} = F + Z$;
6: Calculate IG:

$$V_j(d) = \frac{1}{n}\left(\frac{\left(\sum_{\chi_i \in F_l} r_i + \frac{1-a}{b}\sum_{\chi_i \in Z_l} r_i\right)^2}{n_l^j(d)} + \frac{\left(\sum_{\chi_i \in F_r} r_i + \frac{1-a}{b}\sum_{\chi_i \in Z_r} r_i\right)^2}{n_r^j(d)}\right)$$

7: Develop a novel decision tree $\theta_m(x)'$ on set T'
8: Update $\theta_m(\chi) = \theta_{m-1}(\chi) + \theta_m(\chi)$
9: End of for
10: Return $\tilde{\theta}(\chi) = \theta_M(\chi)$



HOW TO INSTALL LG BOOST ALGORITHM

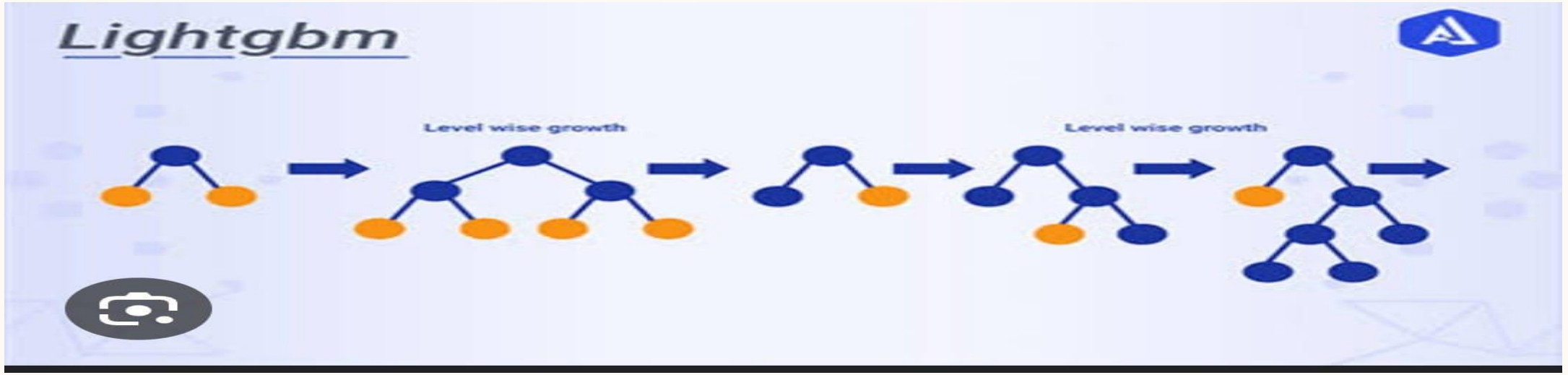
- Run this command to install the lightgbm to perform the lg boost algorithm

pip install --trusted-host pypi.org --trusted-host pypi.python.org --trusted-host files.pythonhosted.org lightgbm

- In python, Algorithm for regressor

from lightgbm import LGBMRegressor

regressor=LGBMRegressor(max_bin=4900, n_estimators=30,learning_rate=0.1, lambda_l1=9.9, max_depth=5)



THANK YOU

Saranya S

https://github.com/Saranya285216/Boosting_Algorithm