BOOSTING Algorithms

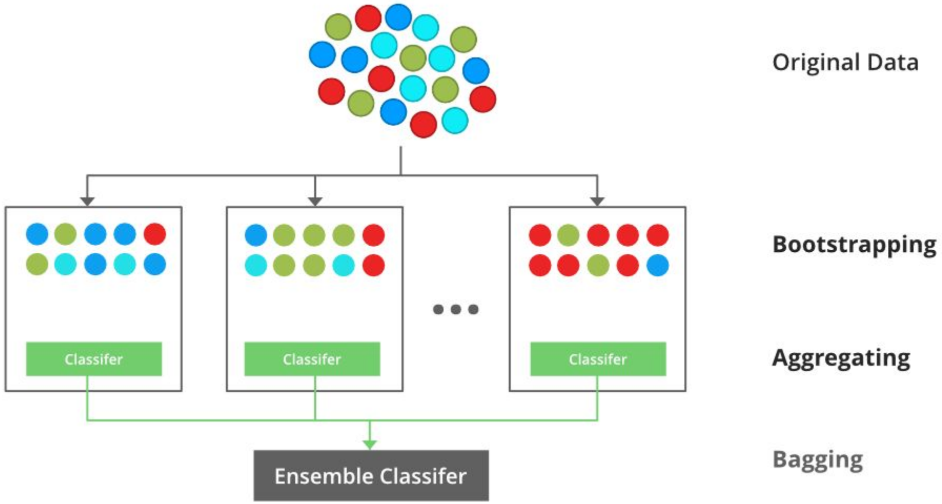
1) AdaBoost
2) XGBoost (Gradient Boosting)
3) LG Boost

Submitted to:
Hope AI
Mentor

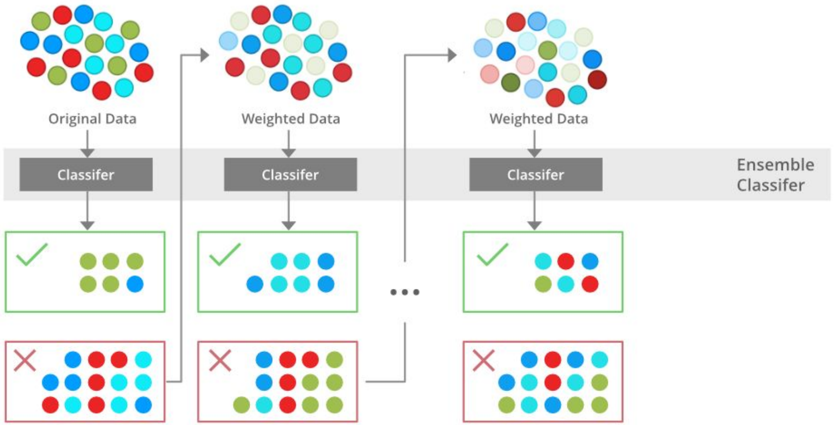
Submitted by:
Jayachandra Prabha K
Student

Difference between **Bagging & Boosting** Algorithm:

BAGGING

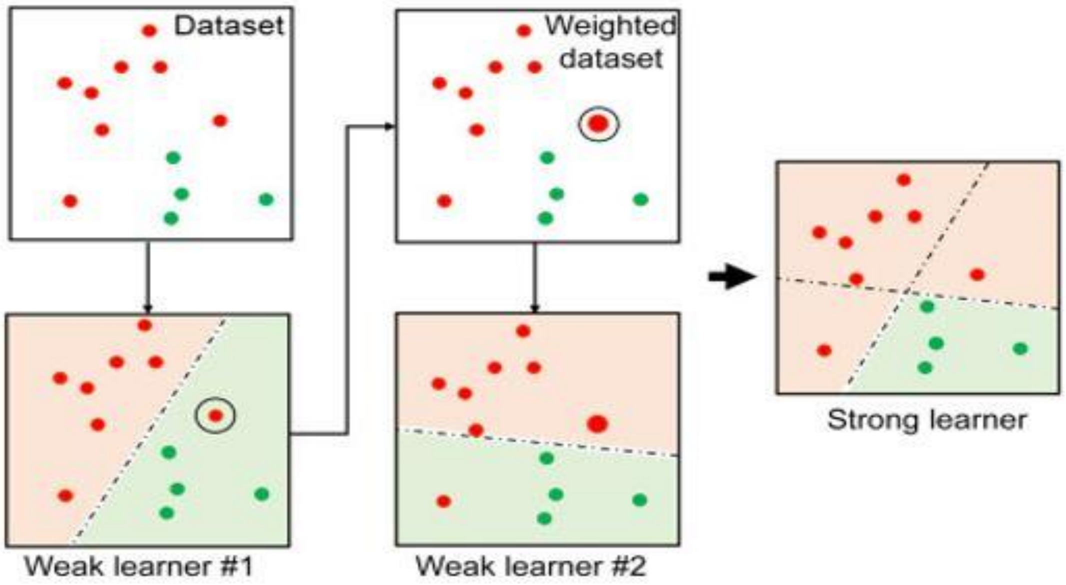


BOOSTING

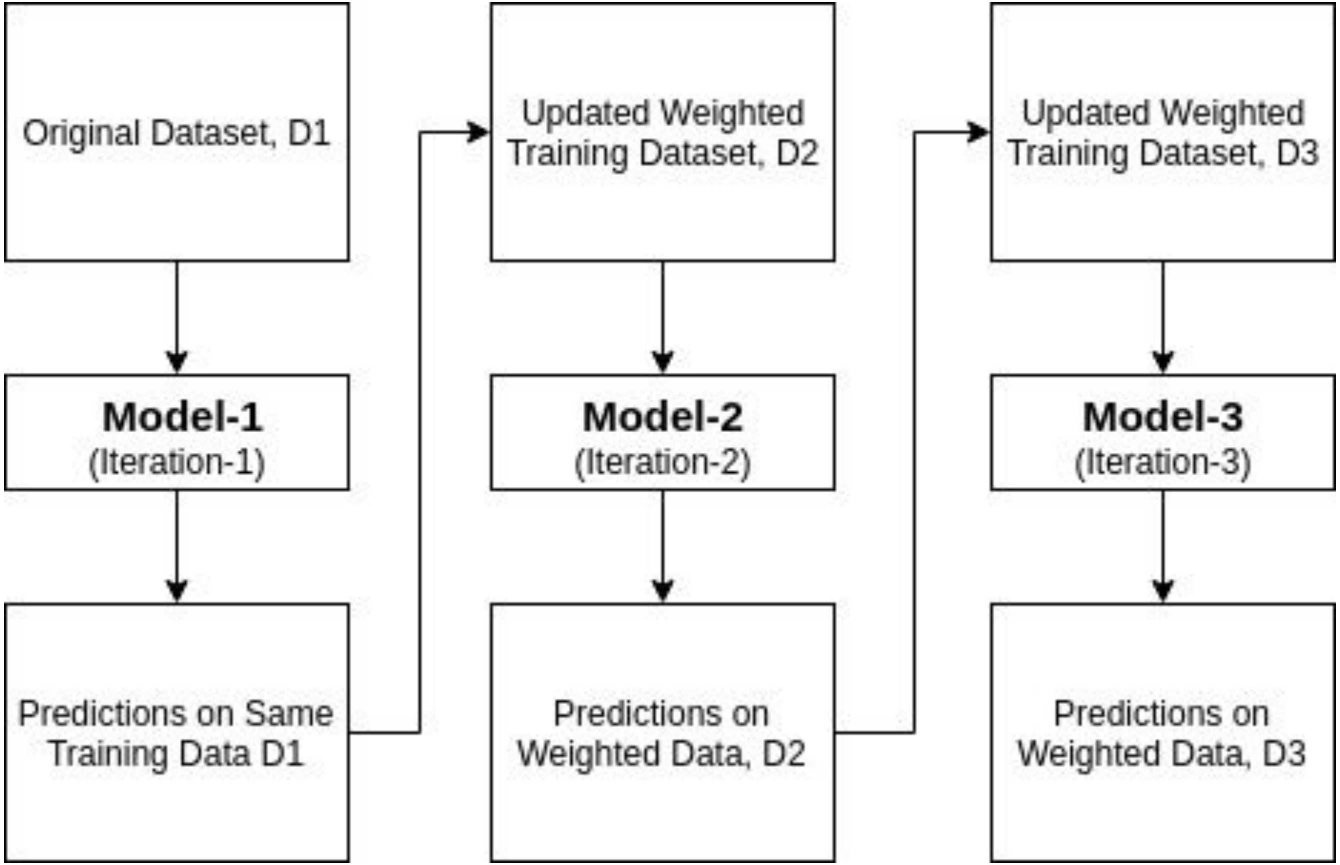


AdaBoost= Adaptive Boosting Algorithm

Group of weak classifiers into a **single strong classifier**



AdaBoost= Adaptive Boosting



Original Dataset, D1

Model-1
(Iteration-1)

Predictions on Same Training Data D1

Updated Weighted Training Dataset, D2

Model-2
(Iteration-2)

Predictions on Weighted Data, D2

Updated Weighted Training Dataset, D3

Model-3
(Iteration-3)

Predictions on Weighted Data, D3

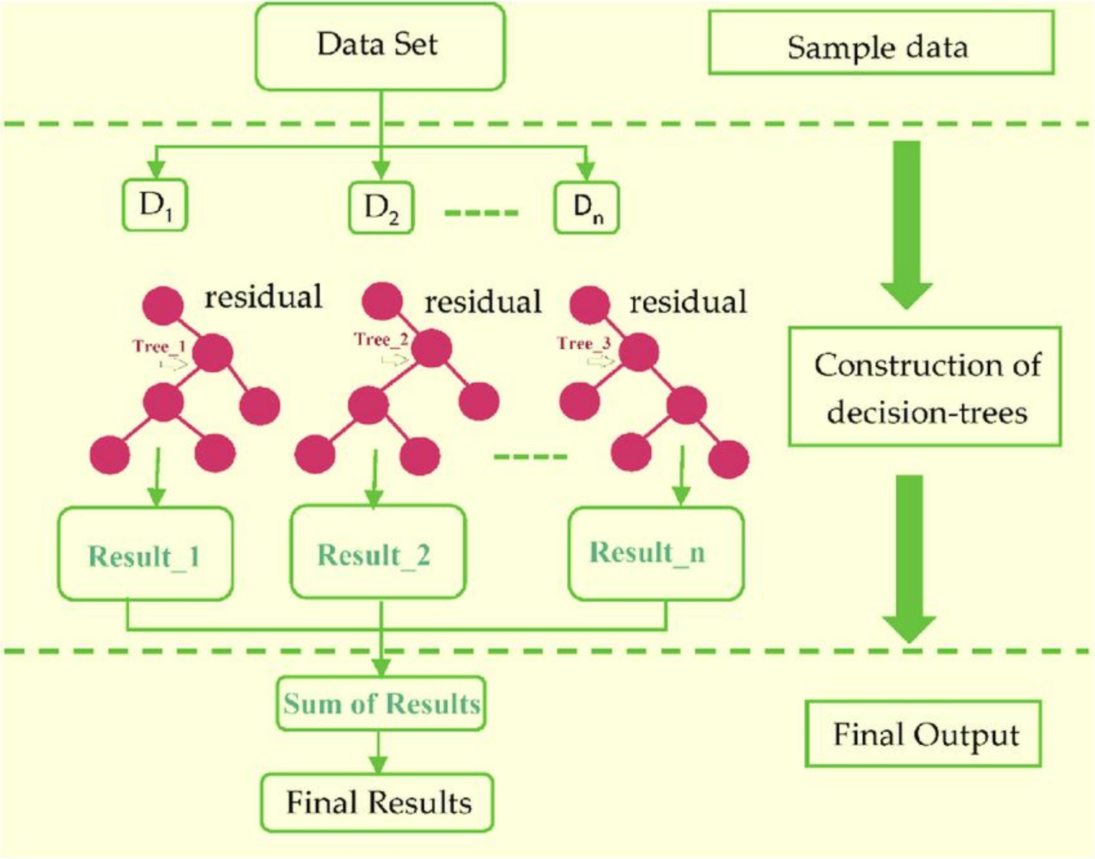
AdaBoost= Adaptive Boosting

Syntax:

```
    from sklearn.ensemble import AdaBoostRegressor

    regressor=AdaBoostRegressor(n_estimators=10, loss="linear")
```

XG Boost= Extreme Gradient Boosting



XG Boost= Extreme Gradient Boosting

Level wise Decision Tree Growth

Steps:

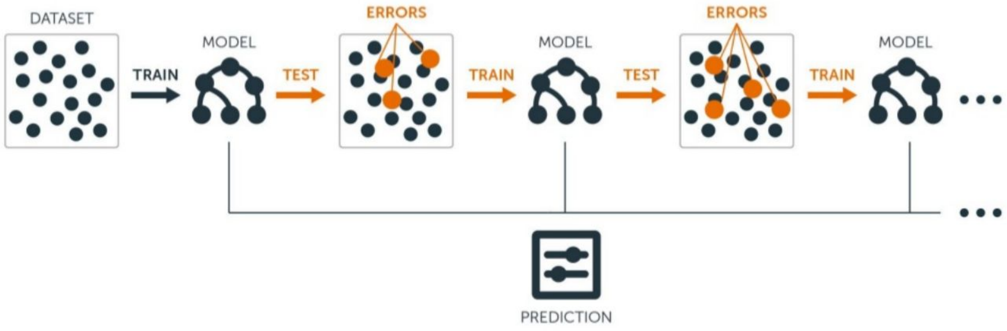
1) **Make an Initial Prediction and Calculate Residuals=(actual-predicted)**
2) **Build an XGBoost Tree**
3) **Prune the Tree (to avoid overfitting)**
4) **Calculate the Output Values of Leaves =(sum of residual/(no.of residuals+lambda))**
   lambda=regularization parameter=1
5) **Make New Predictions**
6) **Calculate Residuals Using the New Predictions**
7) **Repeat Steps 2–6 (**until the *Residuals* are small or reached the maximum number of iterations we set for algorithm)

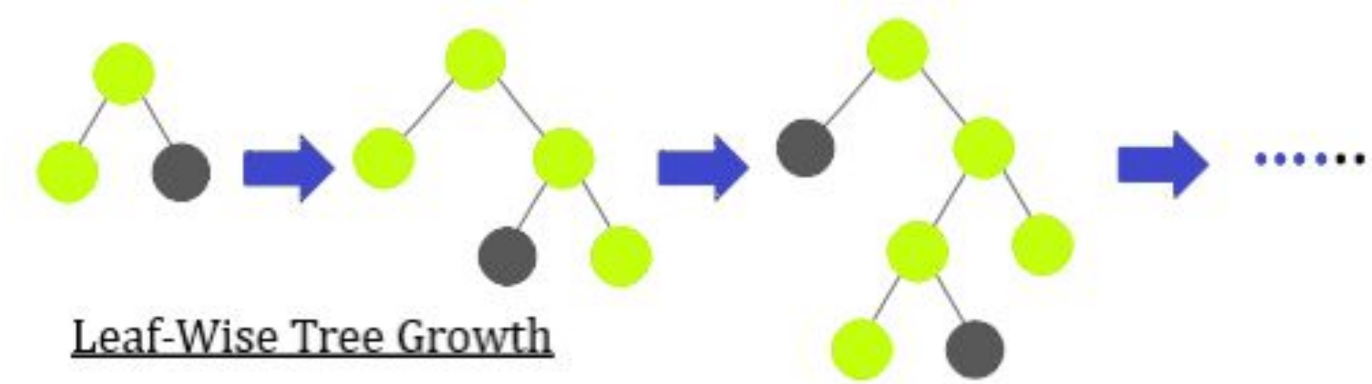
XG Boost= Extreme Gradient Boosting

Syntax:

```
import xgboost as xg
regressor = xg.XGBRegressor(objective ='reg:linear',n_estimators = 10, seed = 123)
regressor.fit(x_train, y_train)
```

LG Boost= Light Gradient Boosting





LG Boost= Light Gradient Boosting

1) Gradient boosting framework
2) Based on Leaf wise Decision Tree Growth

Uses 2 techniques:

i) Gradient-based One Side Sampling(GOSS) &
ii) Exclusive Feature Bundling (EFB)

Syntax:

```
Import lightgbm as lgb
from lightgbm import LGBMRegressor
regressor=LGBMRegressor()
regressor.fit(x_train,y_train)
```