

- ⇒ XGBoost (Advanced form of GB)
- ⇒ Introduced by Tiangi Chen (2014)
↳ PHD research project
- ⇒ Regression and Classification

General.

Already discussed.

Mathematically

(Same as Gradient Boosting)

• Differences

① Speed & Performance

GB \Rightarrow Stored in Numpy arrays, pandas dataframe

XGBoost \Rightarrow Stored in DMatrix

* Compress large data efficiently

\hookrightarrow Size small \rightarrow Model fast

* Sparse data \rightarrow handle with CSR format
(Compressed Sparse Row)

② Regularization (Regression)

GB \Rightarrow Don't have any regularization technique.

XGBoost \Rightarrow Provide L1 and L2 regularization
(Lasso) (Ridge)

\rightarrow Prevent overfitting.

③ Parallel Processing

CPU \rightarrow 8 core

GB \Rightarrow Using \rightarrow 1 core

All trees are built one by one

XGBoost \rightarrow Multi-Threading

CPU \rightarrow 8 core
Using \rightarrow 8 cores

✓
faster

At one time \rightarrow 8 DT
trained

④ Tree Construction

GB \Rightarrow standard approach

XGBoost \Rightarrow "Exact Greedy Algorithm"

It focuses on present split
instead of future outcomes.

⑤ Learning Rate

GB \Rightarrow Constant LR. for throughout model training.

XGBoost \Rightarrow Learning Rate decay

Assign diff LR to diff weak learners
acc to their importance.

lower LR \rightarrow lower importance.

When to Use

- ① Large datasets
- ② Complex / Non-linear data
- ③ Handles missing data
- ④ Less computational than GB.
- ⑤ Structured data

When Not to Use

- ① Highly imbalanced dataset
- ② Non-structured data (images, texts)
- ③ Small & simple dataset
↳ overkill
Overfitting.