Workshop on Machine learning – Day2

ML development for classification Problems

Date: 28th Aug 2021

dubrangala@gmail.com

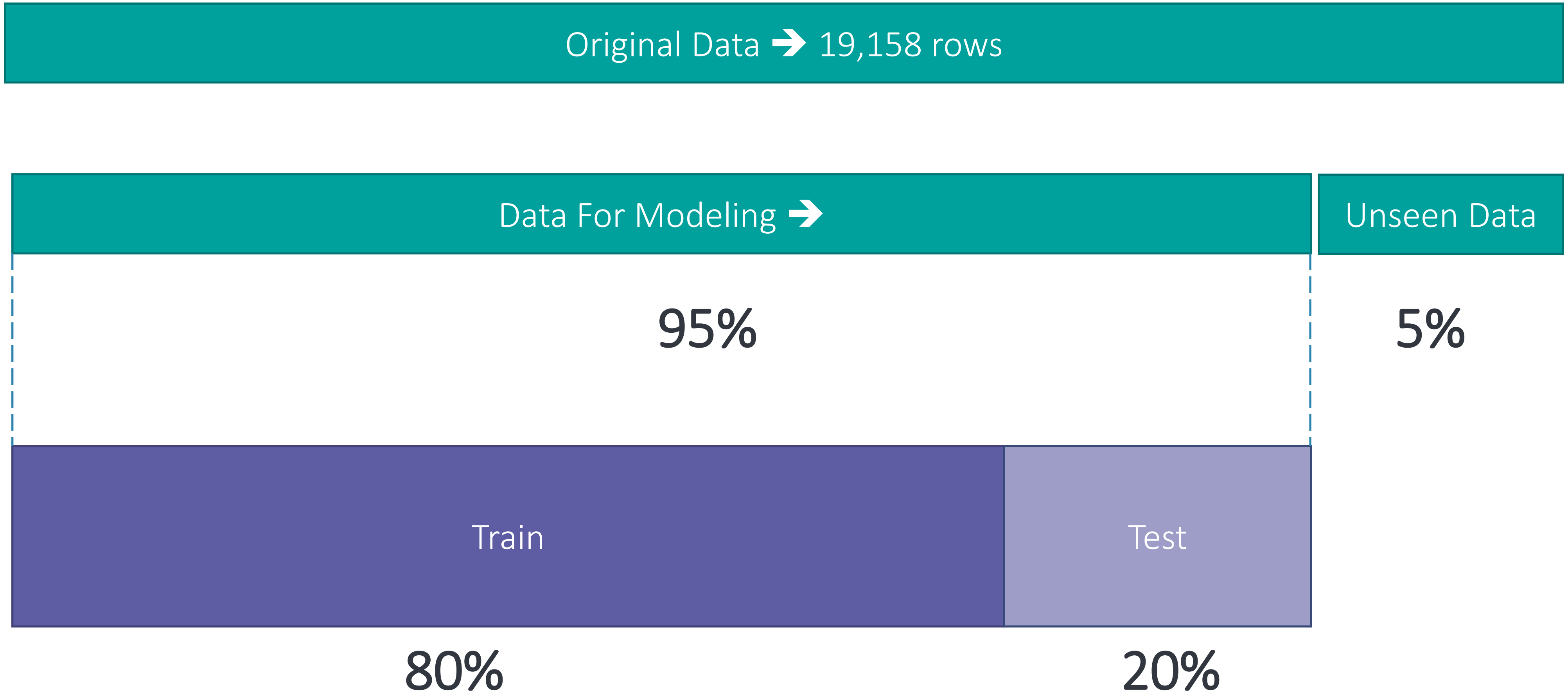
Day 2 Agenda

- General ML model evaluation methods
- Sample Split
- Model Development
  - Random Forest Model
  - XGBoost

Sample Split

Model Training and Evaluation Sample

Split Data



Original Data → 19,158 rows

Data For Modeling →

Unseen Data

95%

5%

Train

Test

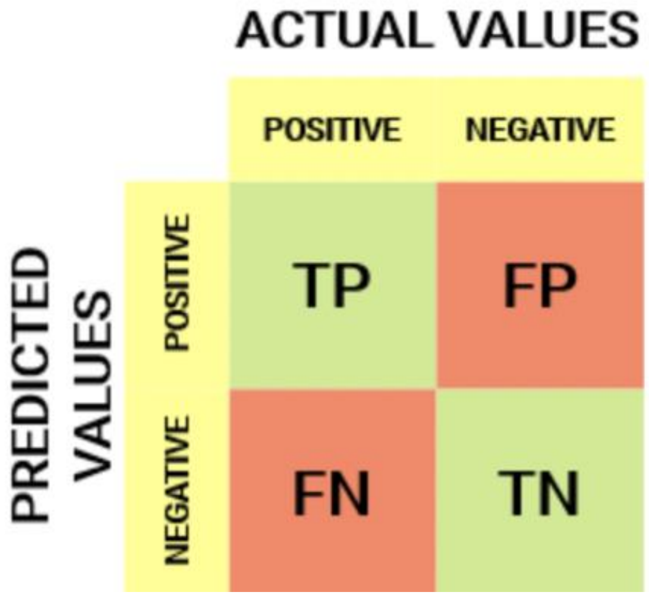
80%

20%

General Classification Model evaluation Methods

Evaluate the model for classification problems

Confusion matrix



		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

$$Precision = \frac{TP}{TP + FP}$$

Precision tells us how many of the correctly predicted cases actually turned out to be positive

$$Recall = \frac{TP}{TP + FN}$$

Recall tells us how many of the actual positive cases we were able to predict correctly with our model.

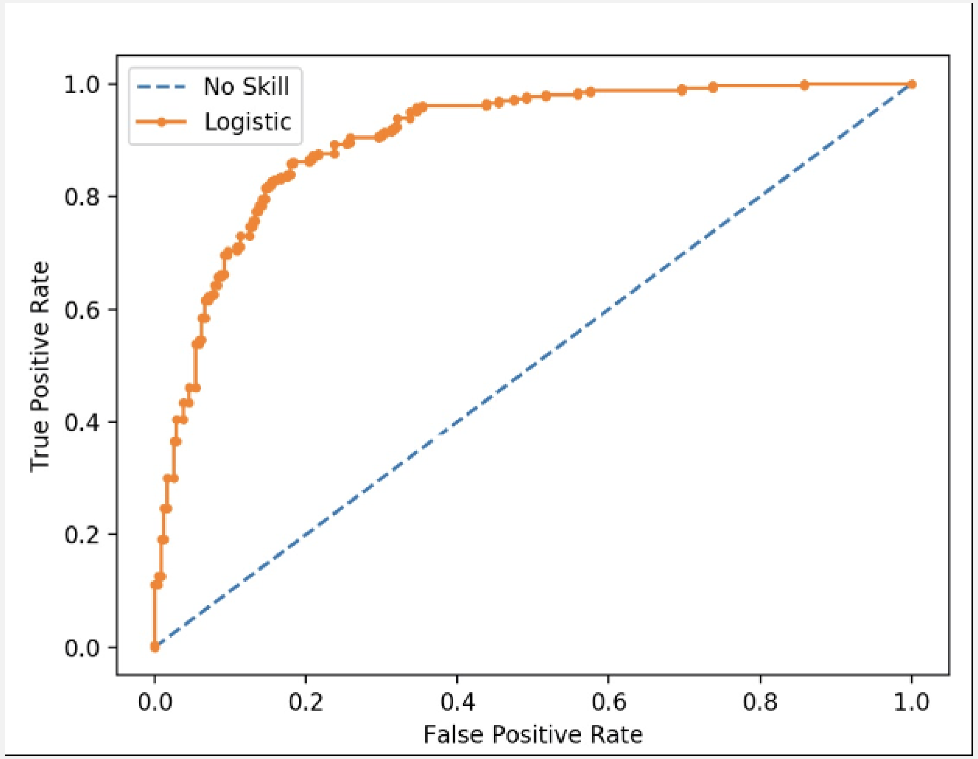
$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

F1-score is a harmonic mean of Precision and Recall, and so it gives a combined idea about these two metrics. It is maximum when Precision is equal to Recall.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

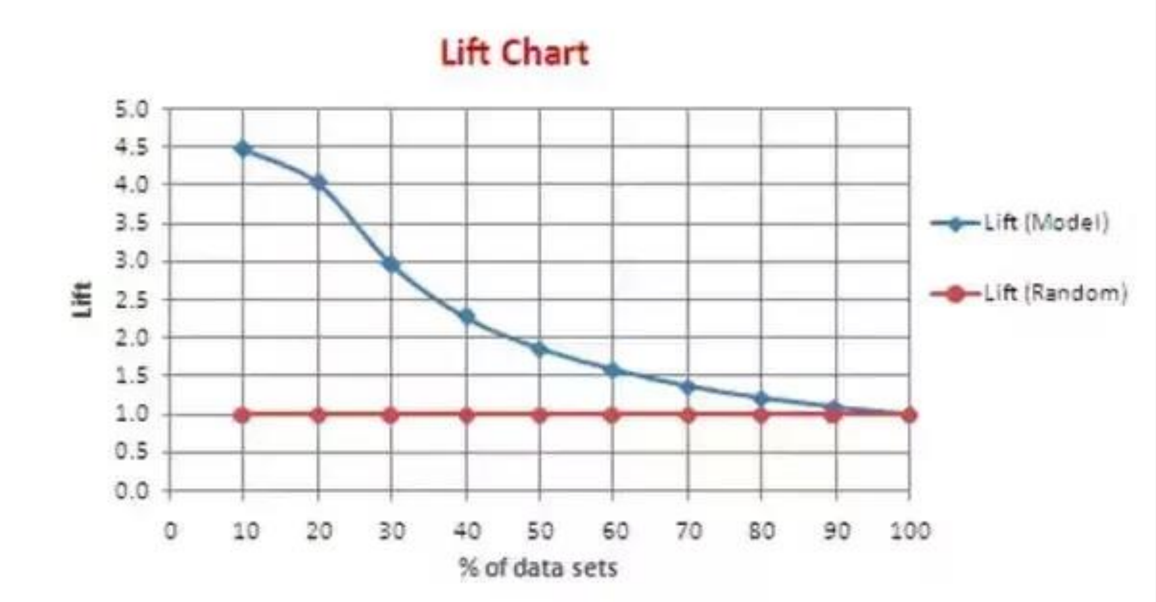
Evaluate the model for classification problems

AUC - ROC



ROC is a probability curve and AUC represents the degree or measure of separability

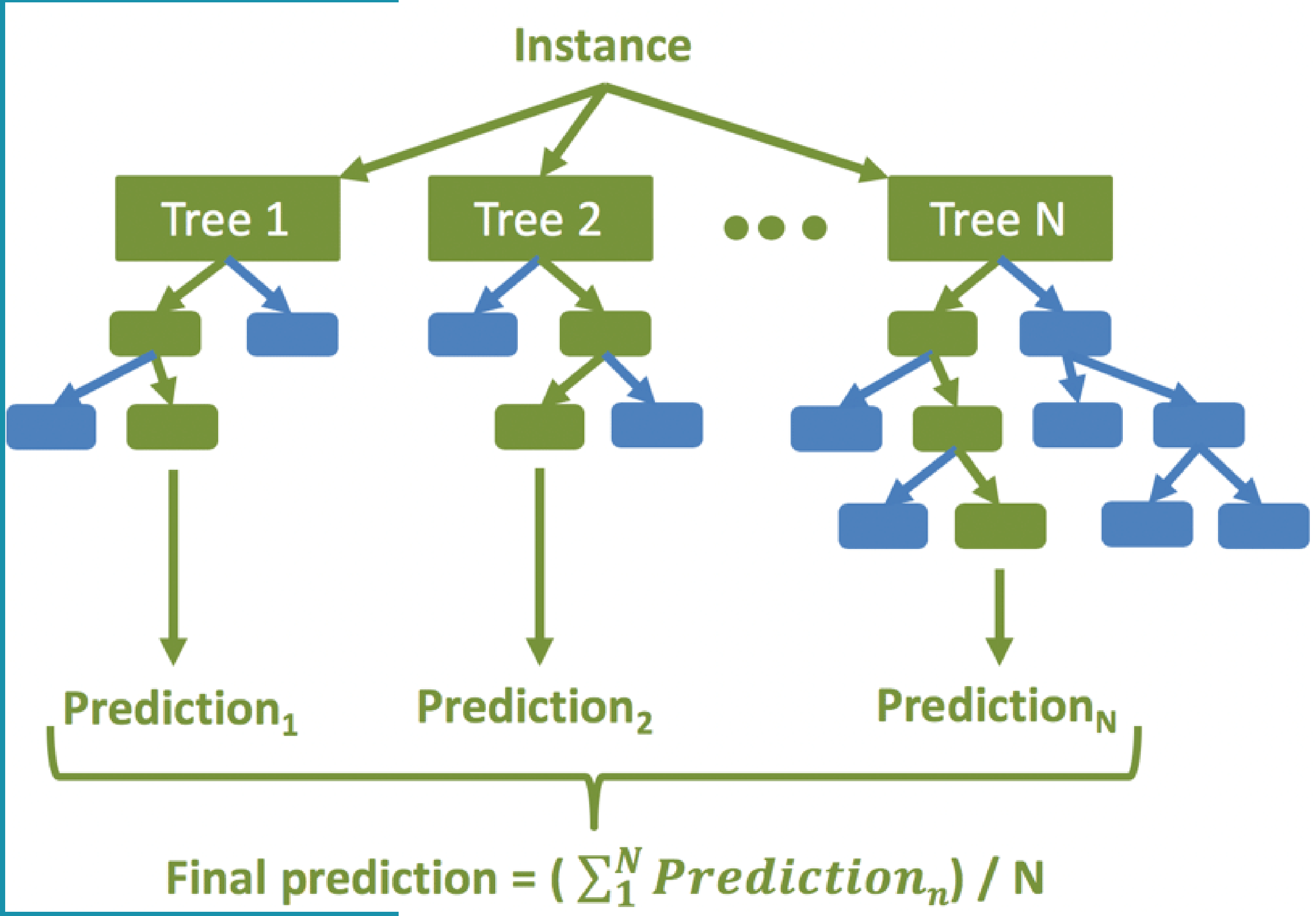
Lift



Lift is a measure of the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model.

Model Development

Random Forest Model



Instance

Tree 1

Tree 2

Tree N

Prediction_1

Prediction_2

Prediction_N

Final prediction = $(\sum_1^N Prediction_n)$ / N

Random Forest Classifier

What is Random Forest

Random Forest is a method that operates by multiple decision trees during training phase.

The decision of the majority of the trees is chosen by the random forest as the final decision.

Why Random Forest

- ❑ No Overfitting
  - ▪ Use of multiple trees reduce the risk of overfitting
  - ▪ Training time is less
- ❑ High Accuracy
  - ▪ Runs efficiently on large database
- ❑ Estimates missing data
  - ▪ Random forest maintain accuracy when a large proportion of data is missing

Random Forest Parameter

•n_estimators = number of trees in the forest

•max_features = max number of features considered for splitting a node

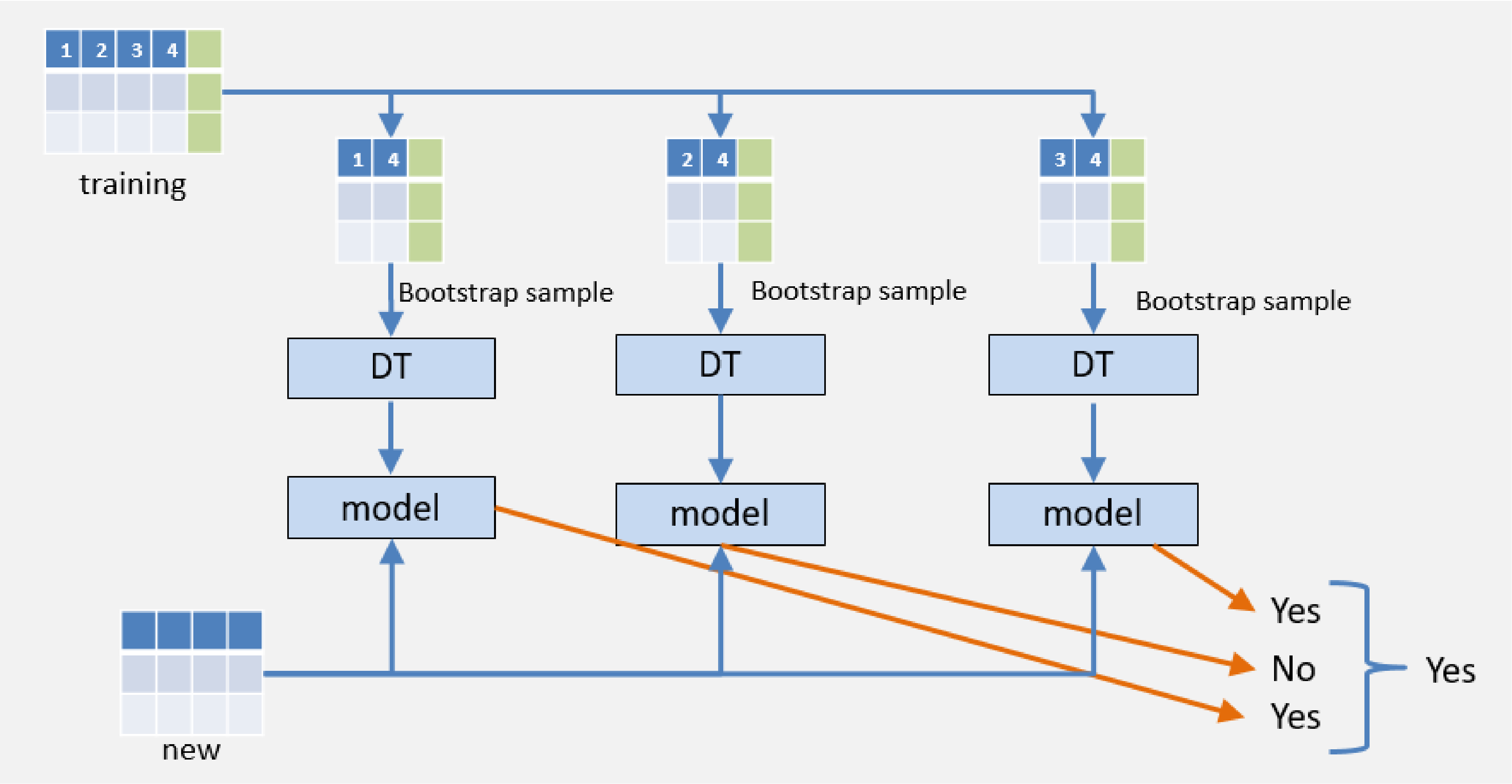
•max_depth = max number of levels in each decision tree

•min_samples_split = min number of data points placed in a node before the node is split

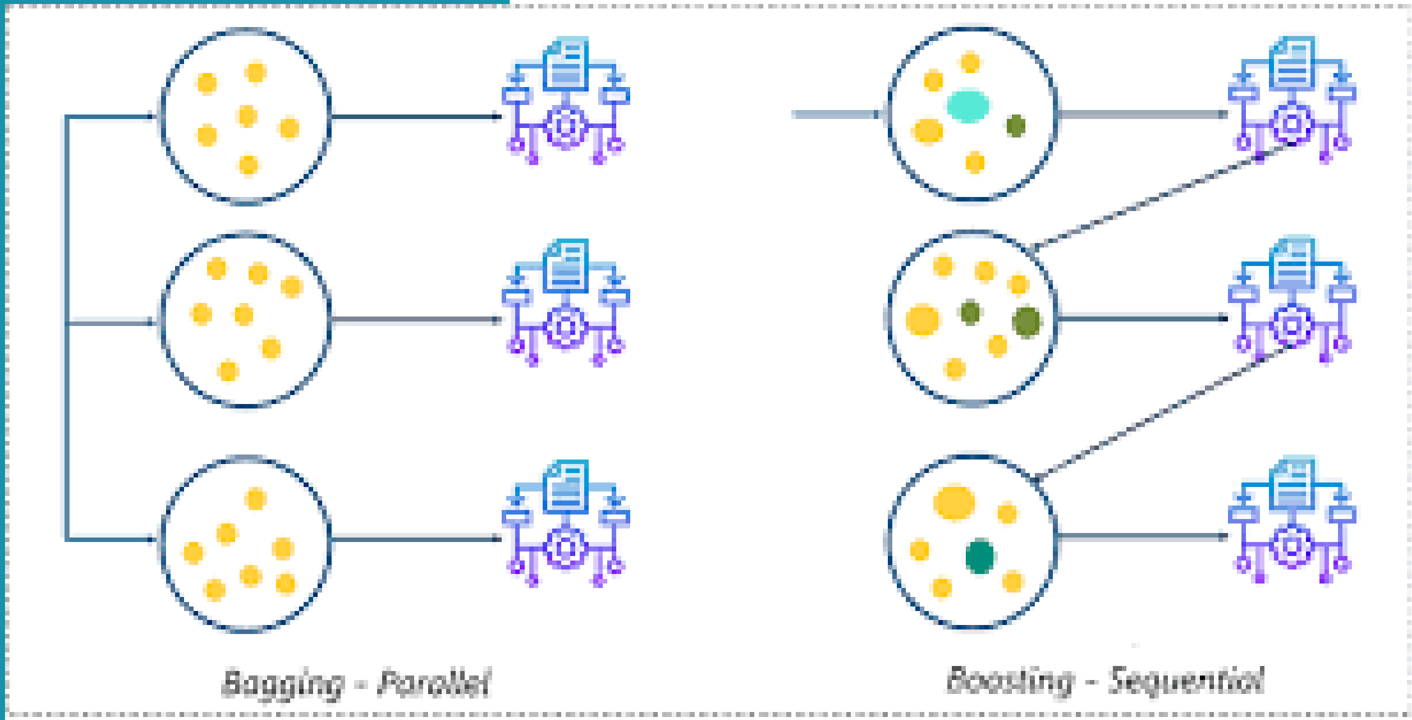
•min_samples_leaf = min number of data points allowed in a leaf node

•bootstrap = method for sampling data points (with or without replacement)

Random Forest

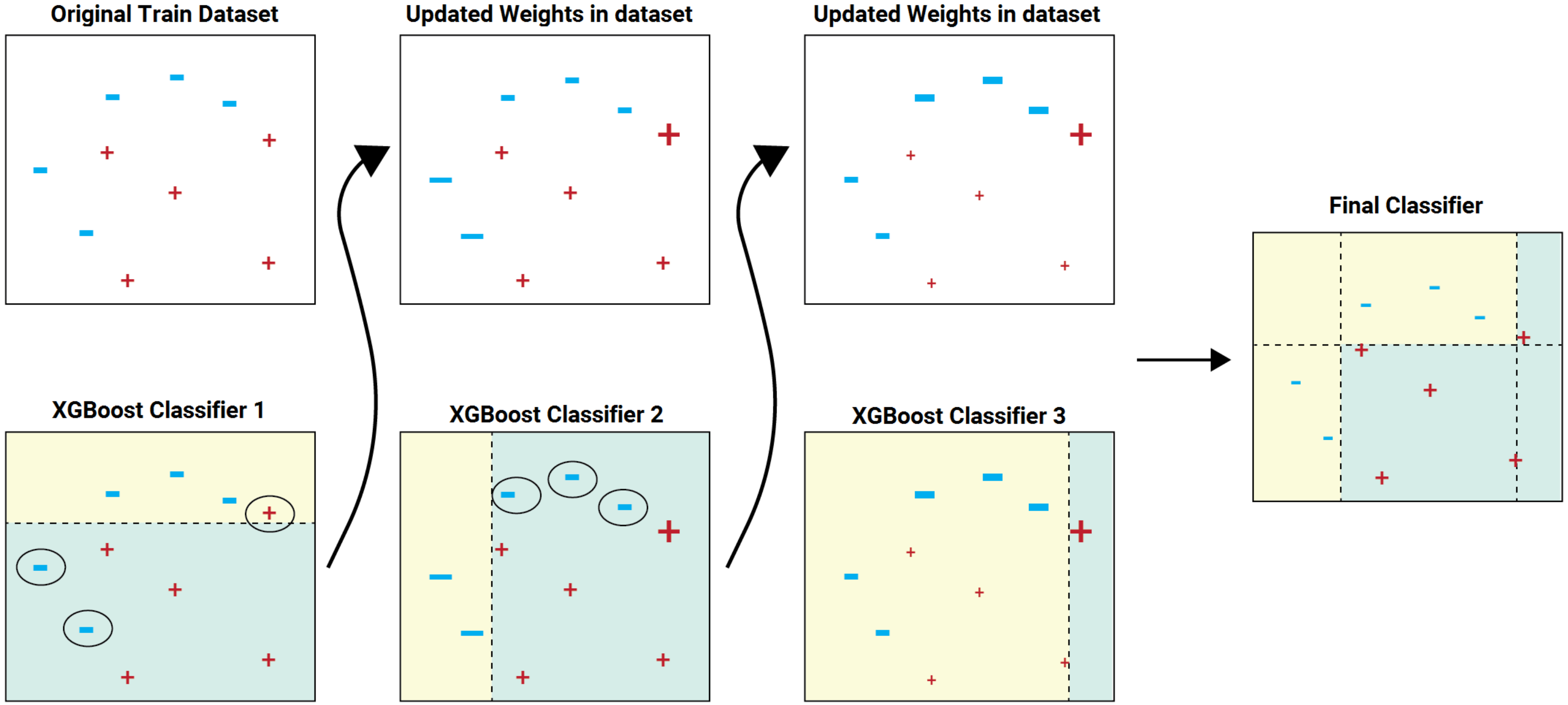


XGBoost Model



XGBoost

- **What is XGBoost?**
- **What is boosting?**
- **What is gradient boosting?**
- **Why is XGBoost so good?**



XGBoost hyperparameters

Generally, the XGBoost hyperparameters have been divided into 4 categories

General parameters	**booster** **nthread** **verbosity**	Booster parameters	**eta** ; **gamma**; **max_depth**; **min_child_weight** **max_delta_step** **Subsample**; **tree_method** **scale_pos_weight** etc..
Learning task parameters	**objective** **eval_metric** **seed**	Command line parameters	They are only used in the console version of XGBoost

Q&A