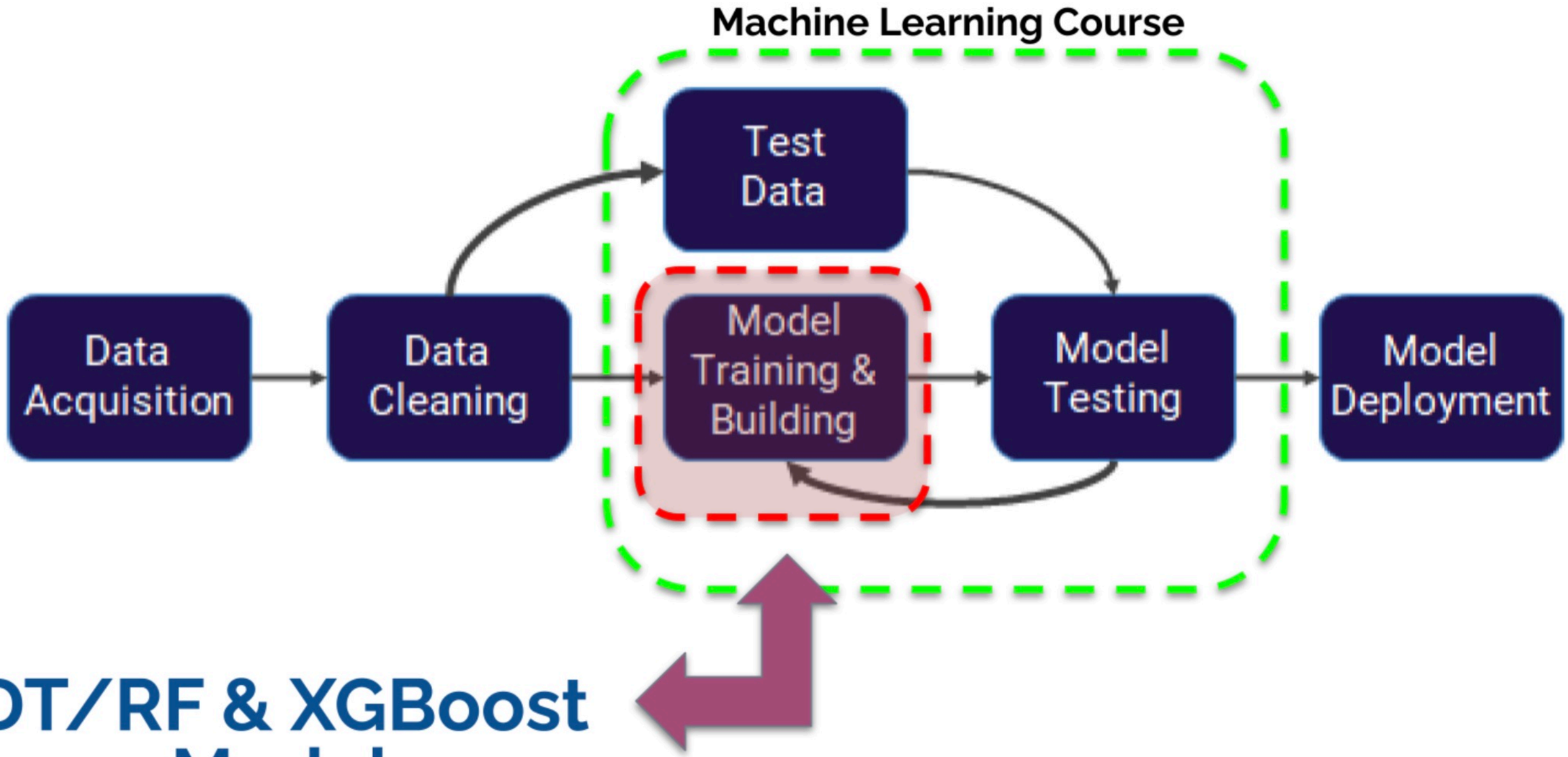
Boosting Theory
Session-14

SUMMARY of PREVIOUS CLASS

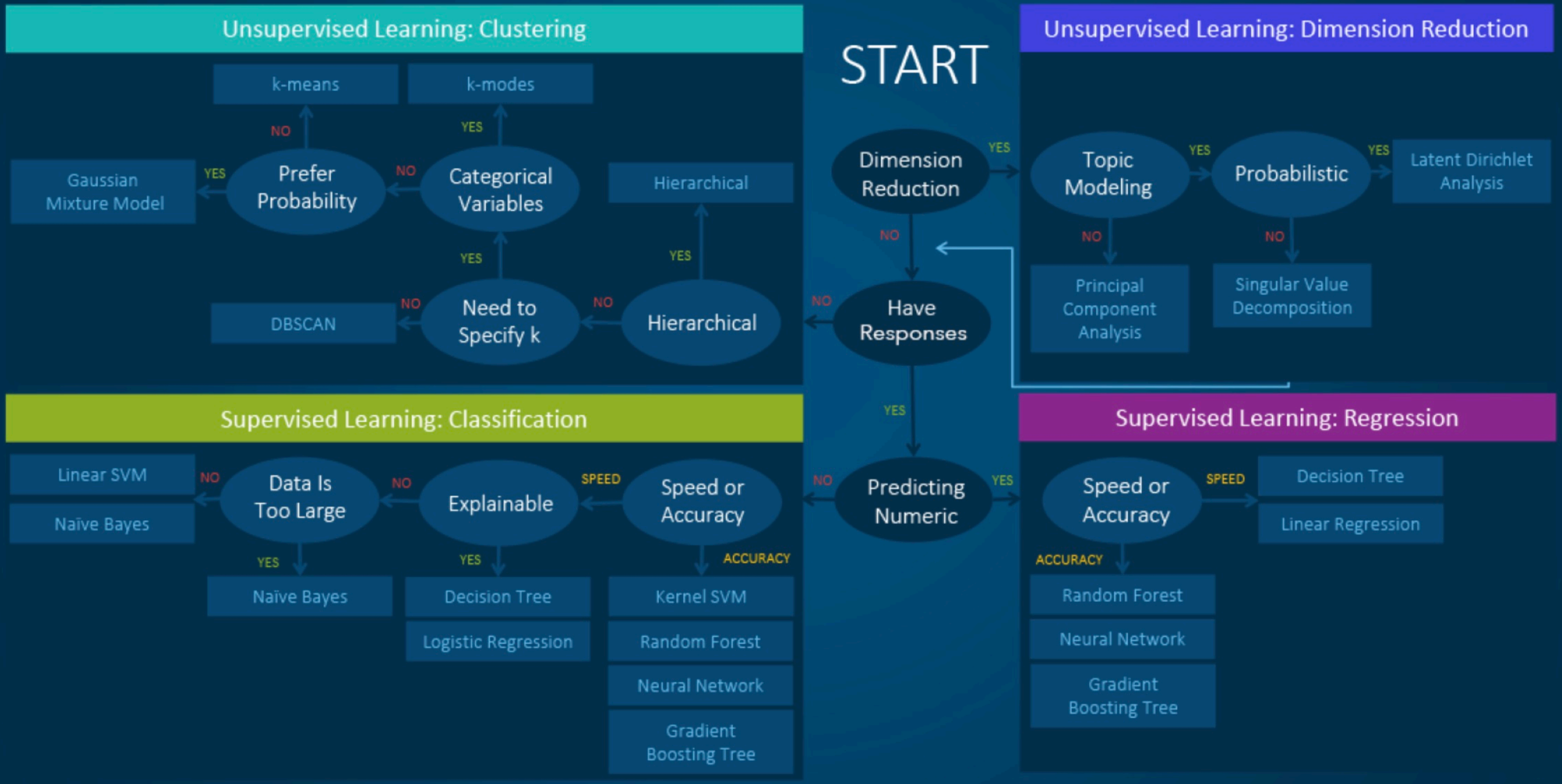
- Ensemble Models
- Ensemble Method: Bagging (Bootstrap Aggregating)
- Random Forest

Where are we?



DT/RF & XGBoost Model

Machine Learning Algorithms Cheat Sheet



Boosting Algorithms

Boosting Algorithms

General info about Ensemble Methods: Bagging & Boosting

Ensemble methods provide training of **different models on the same data set**. Each model makes its own predict and a **meta-model** is formed from the **combination of predictions**. Thus, a strong model is created that makes the most successful prediction. **Bagging-method** prefers strong **sub models**, in contrast **boosting-method** works on **weak learners.**

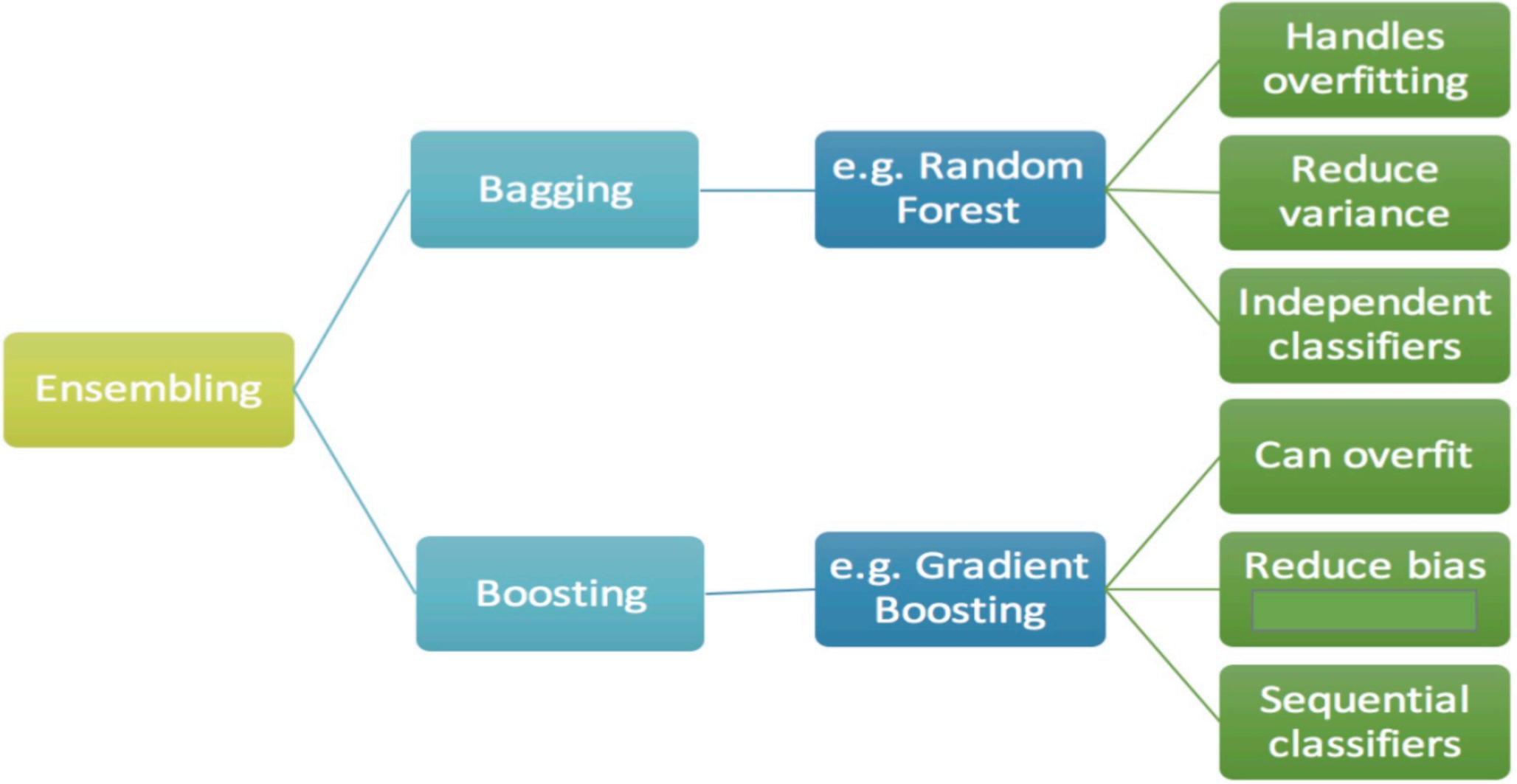
	Bagging	Boosting
Similarities	• Uses voting • Combines models of the same type	
Differences	Individual models are built separately	Each new model is influenced by the performance of those built previously
	Equal weight is given to all models	Weights a model's contribution by its performance

Boosting → AdaBoost

Bagging → Random Forest

An Empirical Study of Bagging and Boosting Ensembles for Identifying Faulty Classes in Object-Oriented Software - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Comparison-between-bagging-and-boosting_tbl1_224453264 [accessed 30 Jun, 2021]

Boosting Algorithms

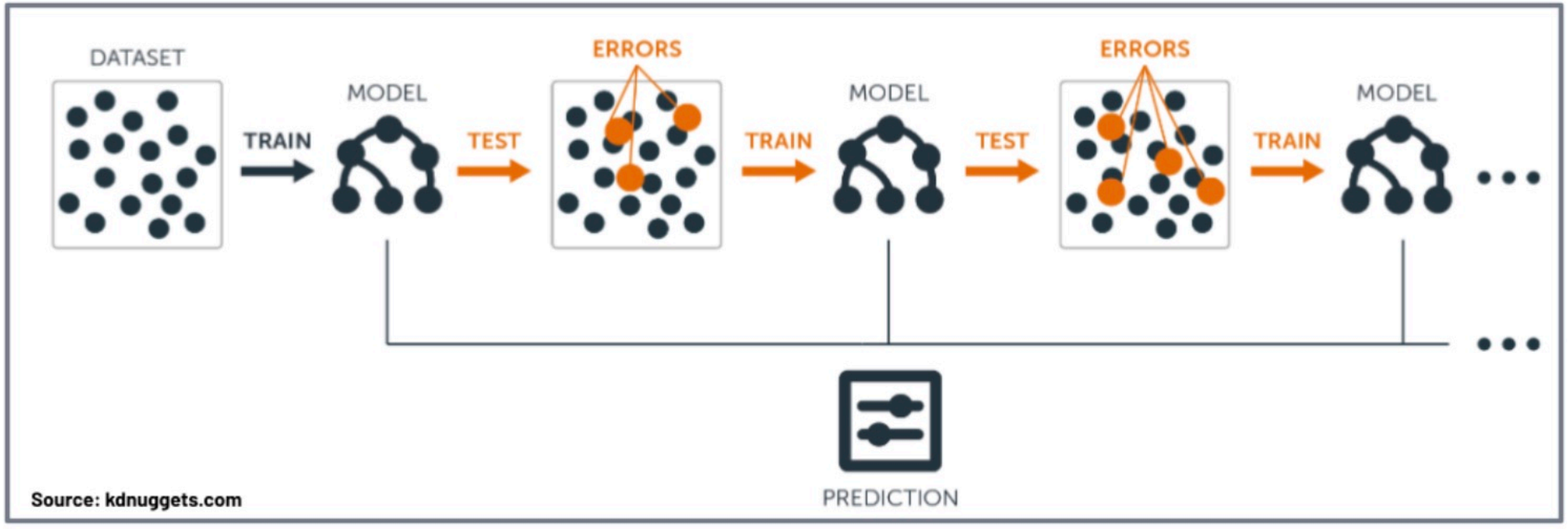


Boosting Algorithms

Boosting :

It is a strong model building method with iterations, **taking into account the errors of weak algorithms** created in the same data set.

It combines the outputs from weak learner and creates a strong learner which eventually improves the prediction power of the model.



AdaBoost,
Gradient Boosting,
XGBoost,
Light GBM,
CAT Boost
etc.

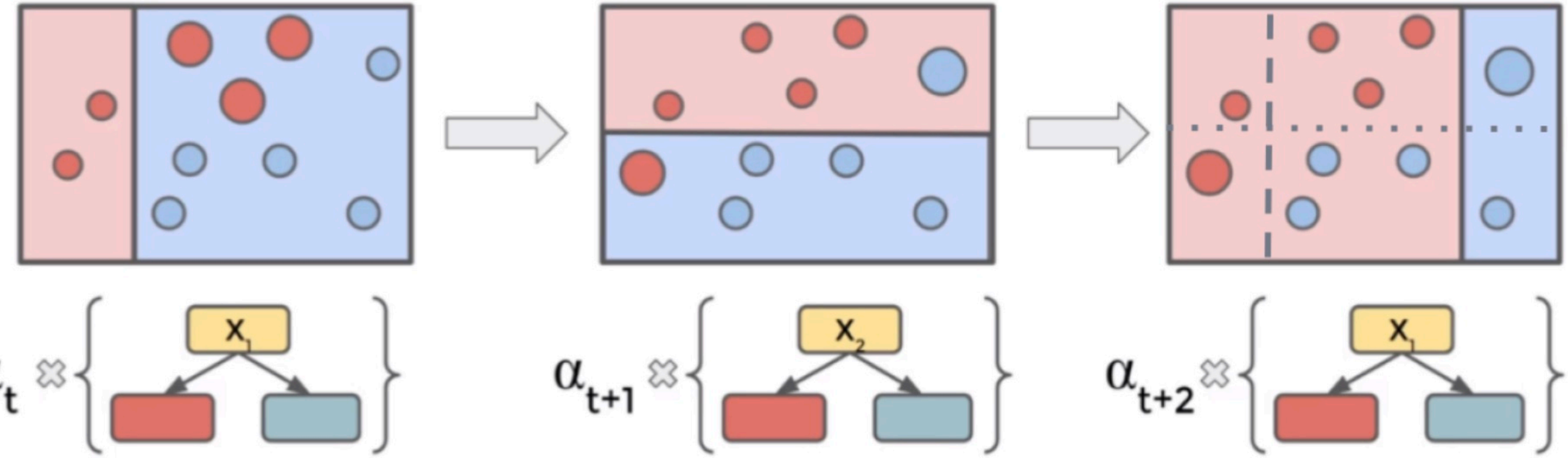
scikit learn **sklearn.ensemble.class**

Boosting Algorithms

Sample of The Boosting Algorithms' Learning:

(AdaBoost - **Ada**ptive **Boost**ing)

$$F_T(x) = \sum_{t=1}^{T} f_t(x) \qquad f_t(x) = \alpha_t h(x)$$



$$\alpha_m = \frac{1}{2}\ln\left(\frac{1-\epsilon_m}{\epsilon_m}\right)$$

Amount of Say = $\frac{1}{2}\log\left(\frac{1 - \text{Total Error}}{\text{Total Error}}\right)$

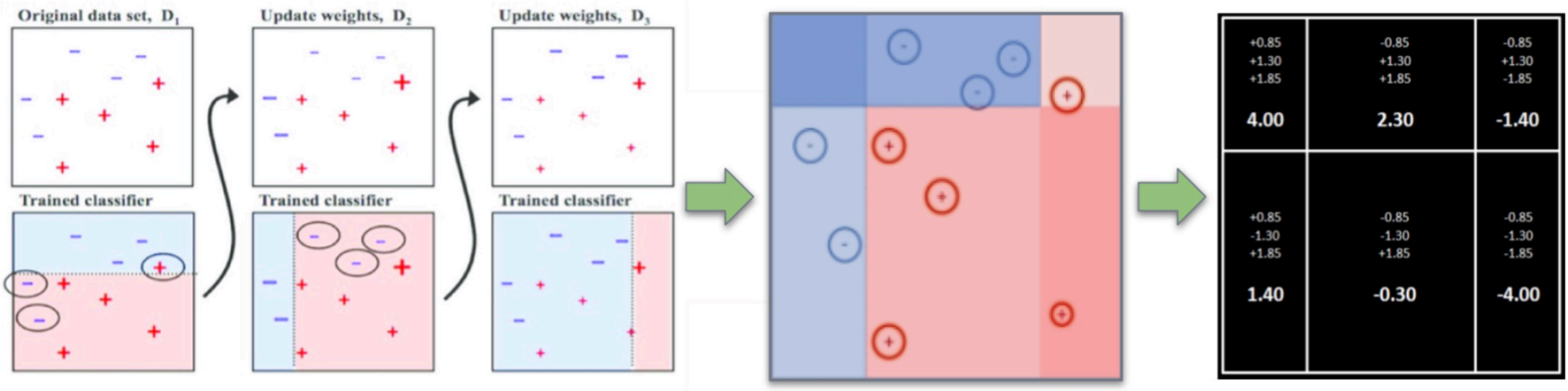
Stump

Boosting Algorithms

Boosting :

Sample of The Boosting Algorithms' Learning:

(AdaBoost - **Ada**ptive **Boost**ing)



Multivariate Analysis of the Vector Boson Fusion Higgs Boson - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Training-of-an-AdaBoost-classifier-The-first-classifier-trains-on-unweighted-data-then_fig3_306054843 [accessed 30 Jun, 2021]

Boosting Algorithms

AdaBoost Algorithm:

Hyperparameters

“base_estimator” *parameter:* (default=None)

The base estimator from which the boosted ensemble is built.

If None, then the base estimator is DecisionTreeClassifier initialized with max_depth=1 (aka stump).

“n_estimators” *parameter:* (default=50)

The maximum number of estimators at which boosting is terminated. In case of perfect fit, the learning procedure is stopped early.

Boosting Algorithms

Gradient Boosting (GBM):

Like Adaboost, the Gradient Boosting algorithm is based on correcting the errors of the previous model. However, **this correction is realized by adding residual estimation errors to the model**, not by changing the weights like AdaBoost.

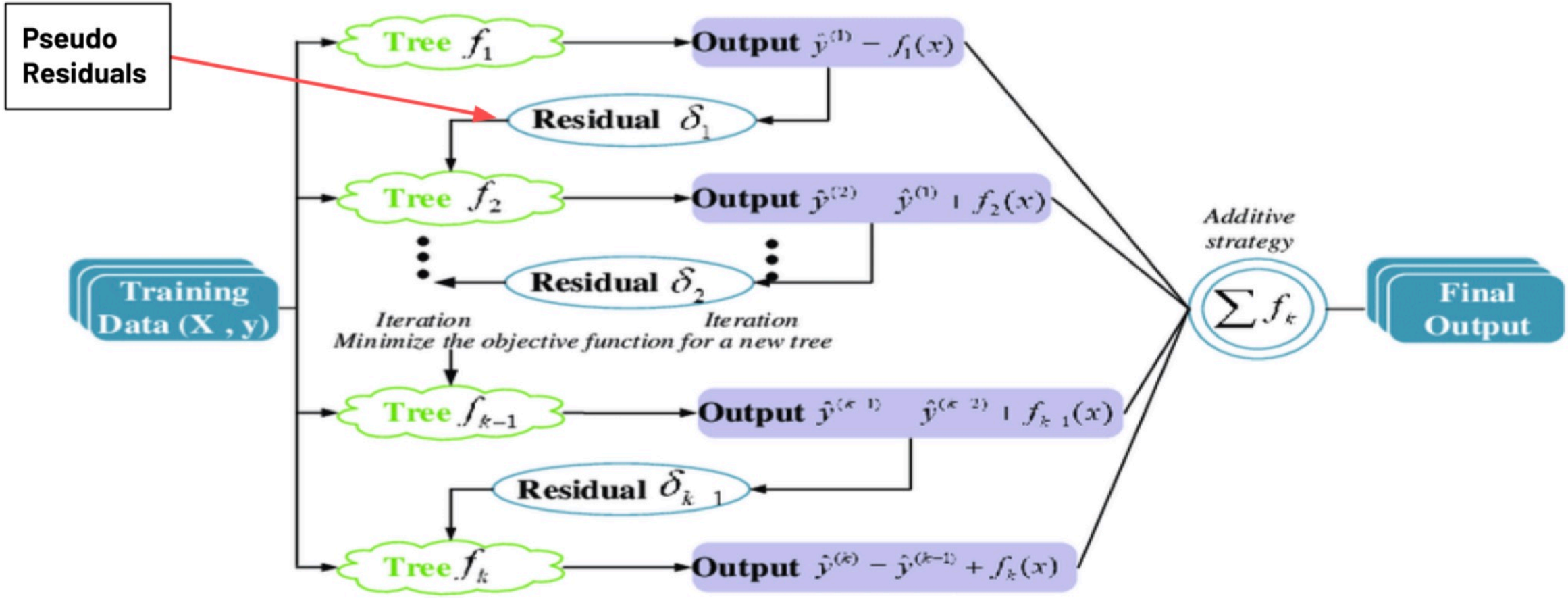
In the **Bagging method**, trees obtain their results **independently** from each other. In **Boosting**, each tree uses the results of the previous one. So **trees are dependent** on each other.

Height	Weight	Male?	Likes Cake	Label
1.02	35	True	False	12
1.82	90	True	False	30
1.67	50	False	True	66

[Friedman, 2001]

Boosting Algorithms

Gradient Boosting (GBM):



Simultaneous Determination of Metal Ions in Zinc Sulfate Solution Using UV-Vis Spectrometry and SPSE-XGBoost Method - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-structure-of-extreme-gradient-boosting_fig3_344011237 [accessed 30 Jun, 2021]

Boosting Algorithms

Adaboost vs Gradient Boosting

AdaBoost	GradientBoost
Both AdaBoost and Gradient Boost use a base weak learner and they try to boost the performance of a weak learner by iteratively shifting the focus towards problematic observations that were difficult to predict. At the end, a strong learner is formed by addition (or weighted addition) of the weak learners.	
In AdaBoost, shift is done by up-weighting observations that were misclassified before.	Gradient boost identifies difficult observations by large residuals computed in the previous iterations.
In AdaBoost "shortcomings" are identified by high-weight data points.	In Gradientboost "shortcomings" are identified by gradients.
Exponential loss of AdaBoost gives more weights for those samples fitted worse.	Gradient boost further dissect error components to bring in more explanation.
AdaBoost is considered as a special case of Gradient boost in terms of loss function, in which exponential losses.	Concepts of gradients are more general in nature.

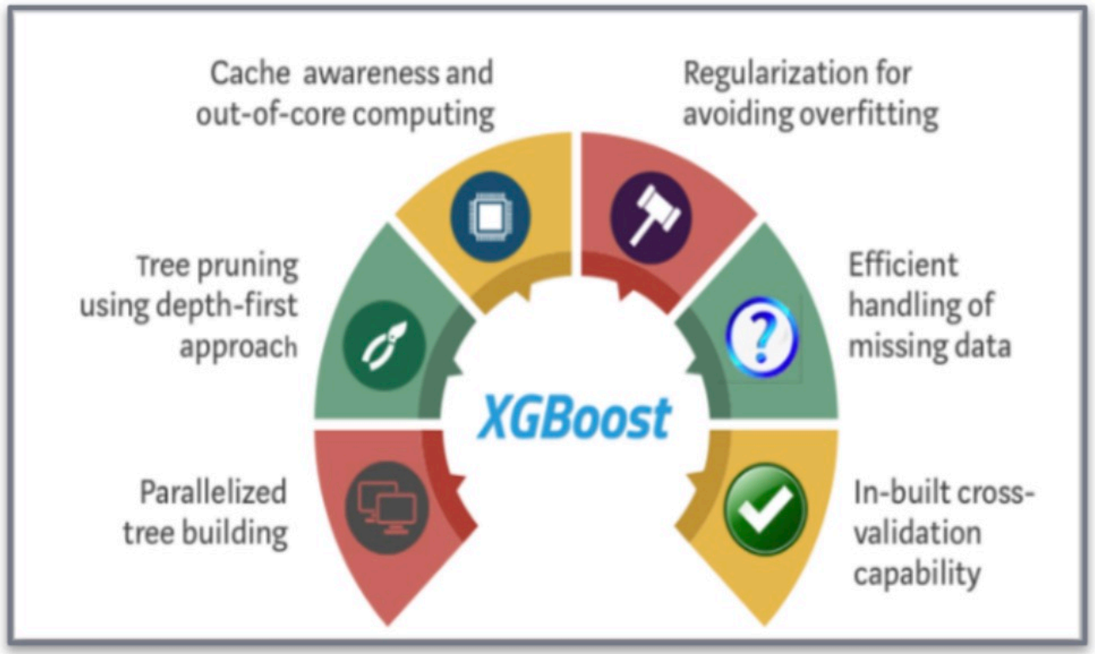
https://www.packtpub.com/mapt/book/big_data_and_business_intelligence/9781788295758/4/ch04lvl1sec34/comparison-between-adaboosting-versus-gradient-boosting

Boosting Algorithms

XGBoost (EXtreme Gradient Boosting):

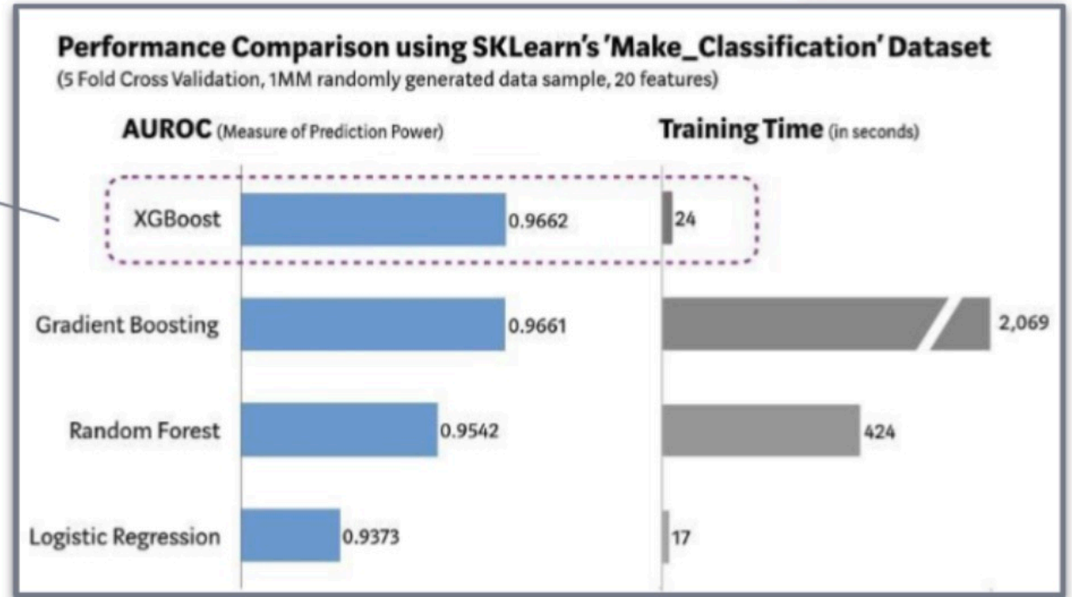
As the name suggests, XGBoost is from the same family as Gradient Boosting. However, there are some additional **EXTREME** capabilities of XGBoost that make it very prominent.

EXTREME Capabilities of XGBoost



Source: towardsdatascience.com/Vishal Morde

kaggle



Boosting Algorithms

XGBoost (EXtreme Gradient Boosting):

- XGBoost builds upon the idea of Gradient Boosting with some modifications to have high performance and faster implementation than gradient boosting and Adaboost algorithm.
- It has a variety of in-built features like handling missing values and different hyperparameters to reduce overfitting and improve overall performance.
- Steps involved in XGBoost-
    1. XGBoost sets the initial prediction for all observations as the mean of the target values for regression and 0.5 for binary classification problems.
    2. Then it calculates the residual for each observation.
    3. It builds a tree to predict the residuals and
    4. The main difference in terms of building a tree between xgboost and gradient boosting is how they decide the best split at each level. While gradient boosting traditional slitting criterion like Gini or entropy, xgboost uses gain calculated from similarity score to find the best split which is directly
    5. After a tree a built, it calculates the output value for each leaf node.
    6. The same process from steps 2-4 is repeated until there is no reduction in the residuals, or the number of estimators is reached.

Boosting Algorithms

XGBoost (EXtreme Gradient Boosting):

How a tree is built in XGBoost?

1. At each level, XGBoost calculates a **similarity score(ss)** for the node and the possible two leaves.

SS for Regression

$$\frac{(\sum Residuals)^2}{Number\ of\ Samples + \lambda}$$

SS for Classification

$$\frac{(\sum Residuals)^2}{[Probability * (1 - Probability)] + \lambda}$$

2. The formula for similarity score is derived directly from the loss function and helps to find the optimal point which minimizes the loss function, Where λ is a regularization parameter in loss function of XGBoost

3.Gain is calculated for each possible split, where the gain is calculated as **Left leave ss + Right leave ss - Root node ss**

4. The split which maximizes the gain is considered as the best split and is chosen to grow a tree.

XGBoost Algorithm

XGBoost Algorithm:

Hyperparameters

“n_estimators” *parameter:* (default=100)

The number of boosting stages to perform.

“subsample” *parameter:* (default=1.0)

The fraction of samples to be used for fitting the individual base learners. Choosing subsample < 1.0 leads to a reduction of variance and an increase in bias.

XGBoost Algorithm

XGBoost Algorithm:

Hyperparameters

“max_*depth*” *parameter:* (default=3)

Max. depth of the individual estimators. This parameter ***limits the number of nodes*** *in the tree.*

“learning_rate” *parameter:* (default=0.1)

Learning rate shrinks the contribution of each tree by learning_rate. There is a trade-off between learning_rate and n_estimator

XGBoost Algorithm

Pros & Cons

Pros:

- **Handles large sized datasets well**
- Perfect for quick predictions
- Ranking for feature importance
- Good model performance
  (wins most of the Kaggle competitions)

Cons:

- Difficulty interpretation due to visualization difficulties
- Harder to tune as there are too many hyperparameters

DT & RF & XGBoost Algorithm

Be ready for

Boosting Python

Session