Random Forest Ensemble Method

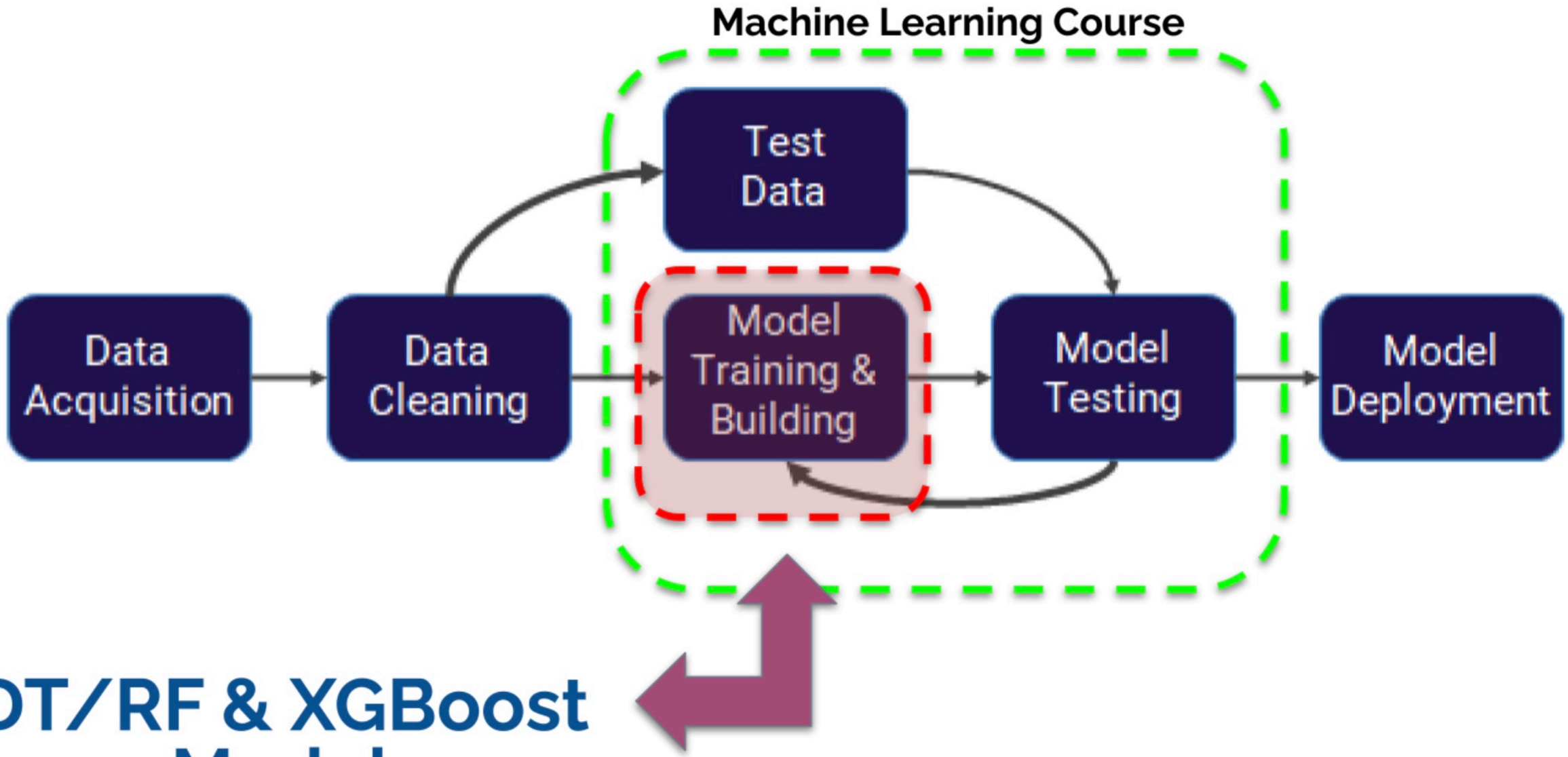
Session-13

SUMMARY of PREVIOUS CLASS

- **Decision Tree Theory:** Root node, Decision node, Leaf Node
- **Gini Index:** minimize the probability of misclassification.
- **Information Gain-Entropy:** obtain the largest IG by minimizing the entropy
- **Regression:** Minimize variance, calculate mean

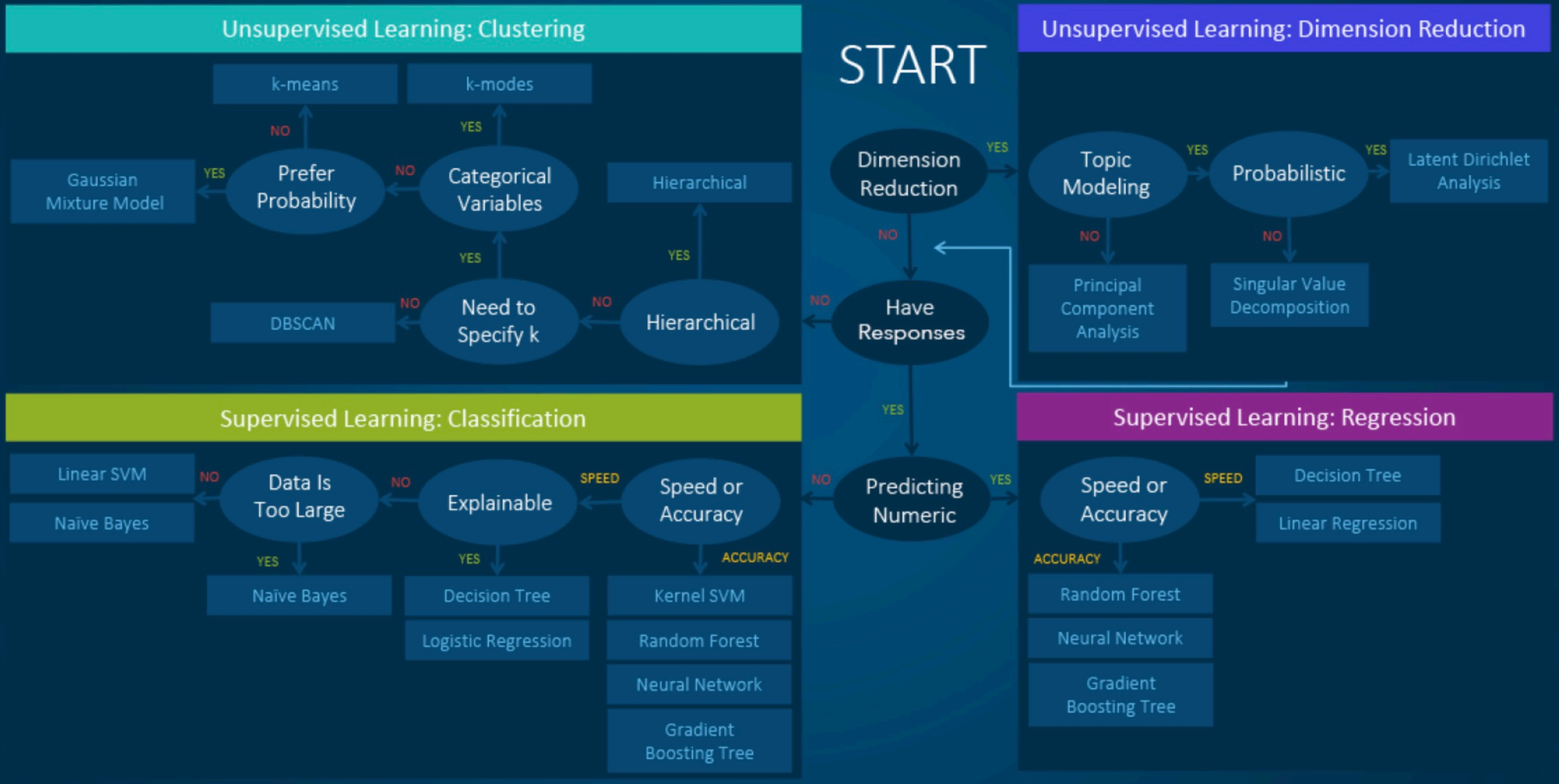
- Ensemble Methods
- Bagging
  (Bootstrap AGGregatING)
- Random Forest Theory
- Random Forest /w Python

Where are we?

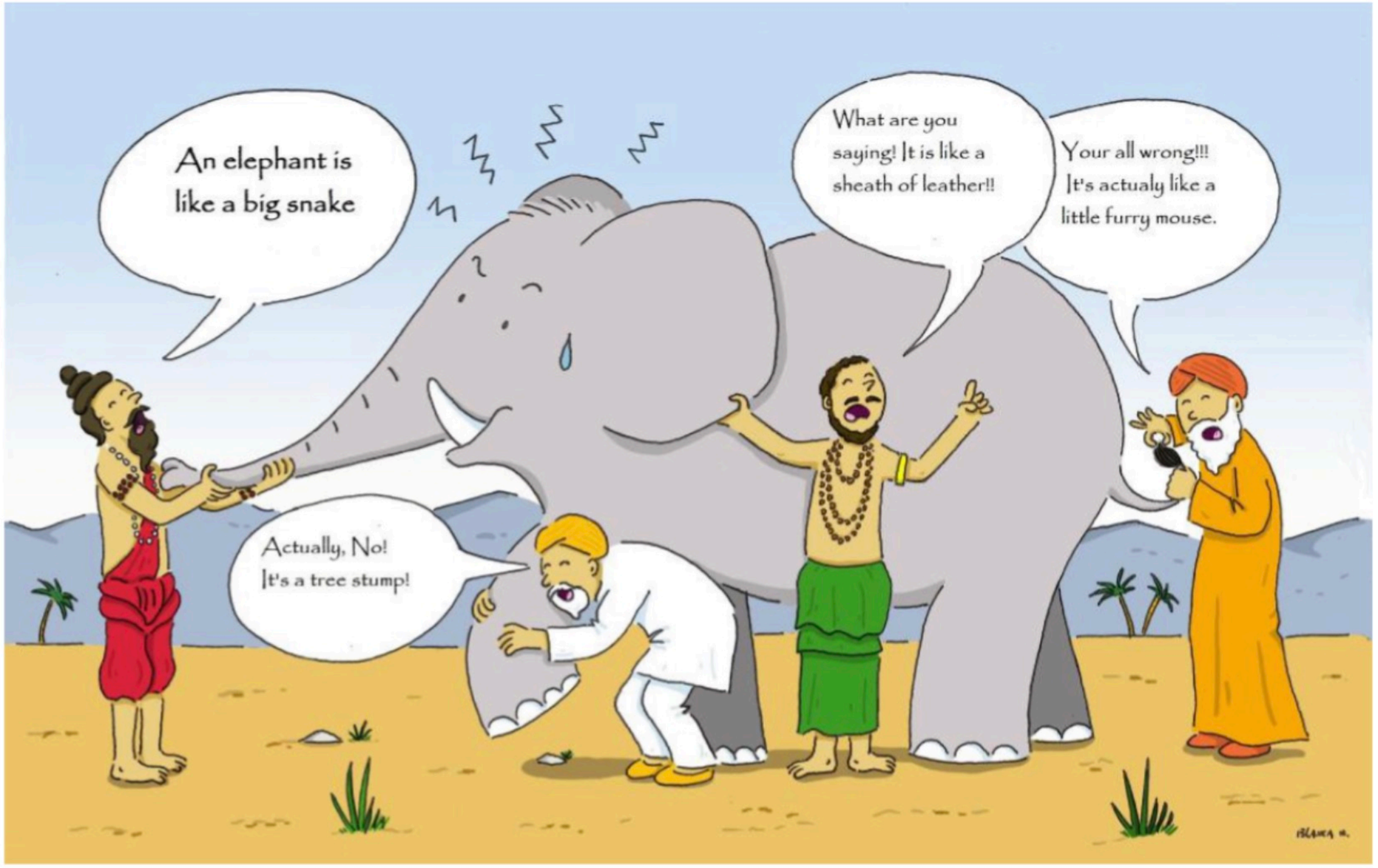


DT/RF & XGBoost Model

Machine Learning Algorithms Cheat Sheet



Ensemble Methods



Ensemble Methods

General info about Ensemble Methods: Bagging & Boosting

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

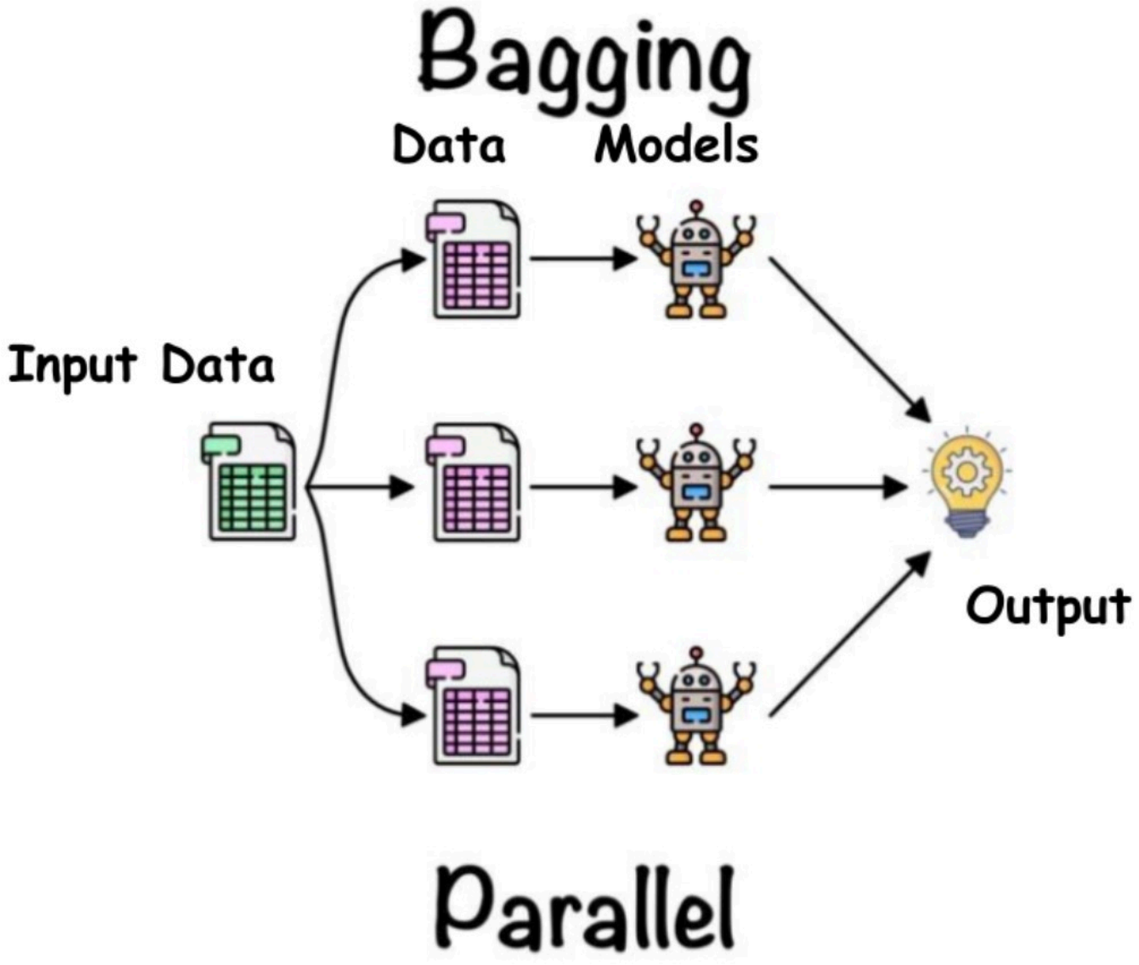
	Bagging	Boosting
Similarities	• Uses voting • Combines models of the same type	
Differences	Individual models are built separately	Each new model is influenced by the performance of those built previously
	Equal weight is given to all models	Weights a model's contribution by its performance

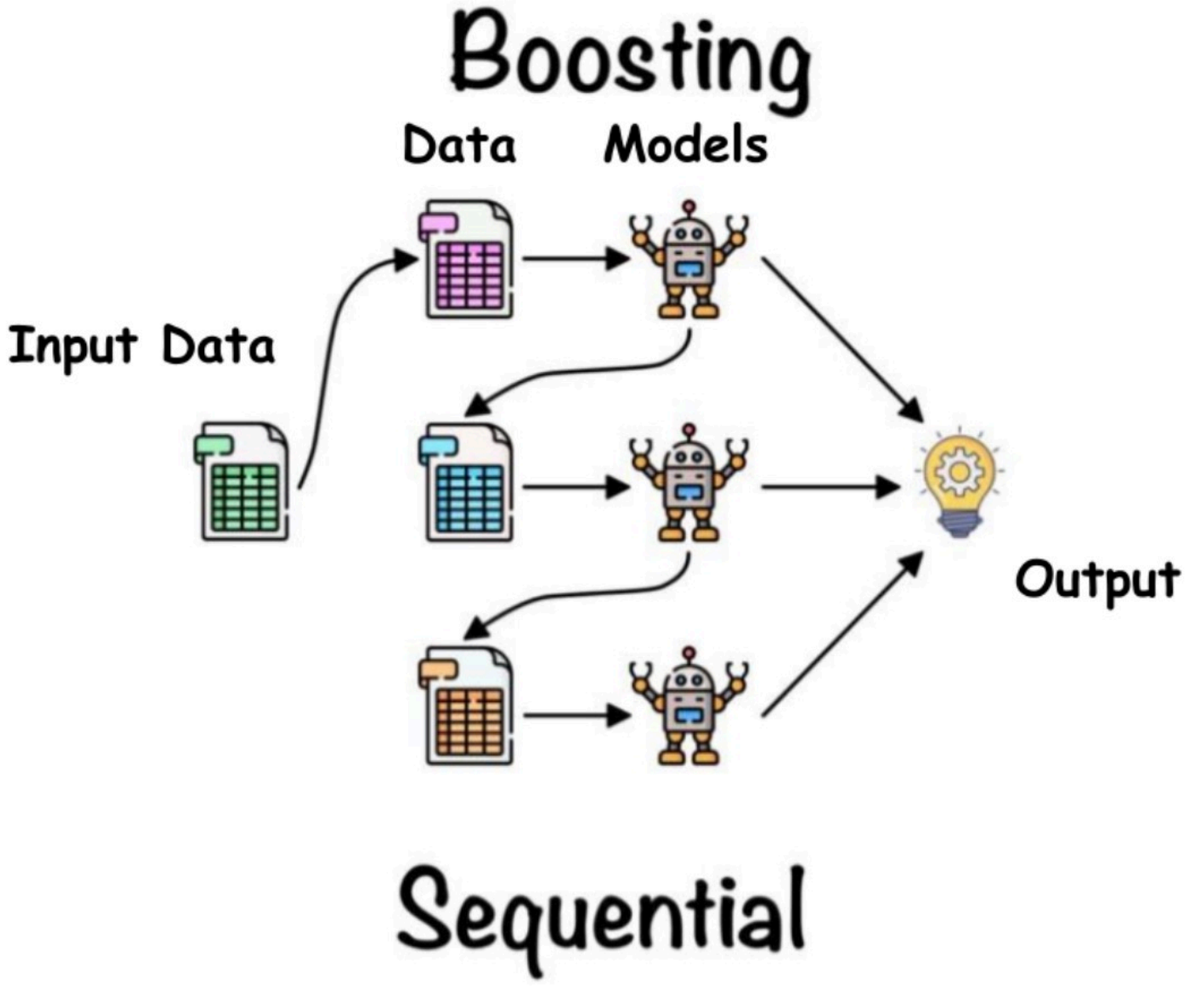
Bagging → Random Forest

Boosting → AdaBoost

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]

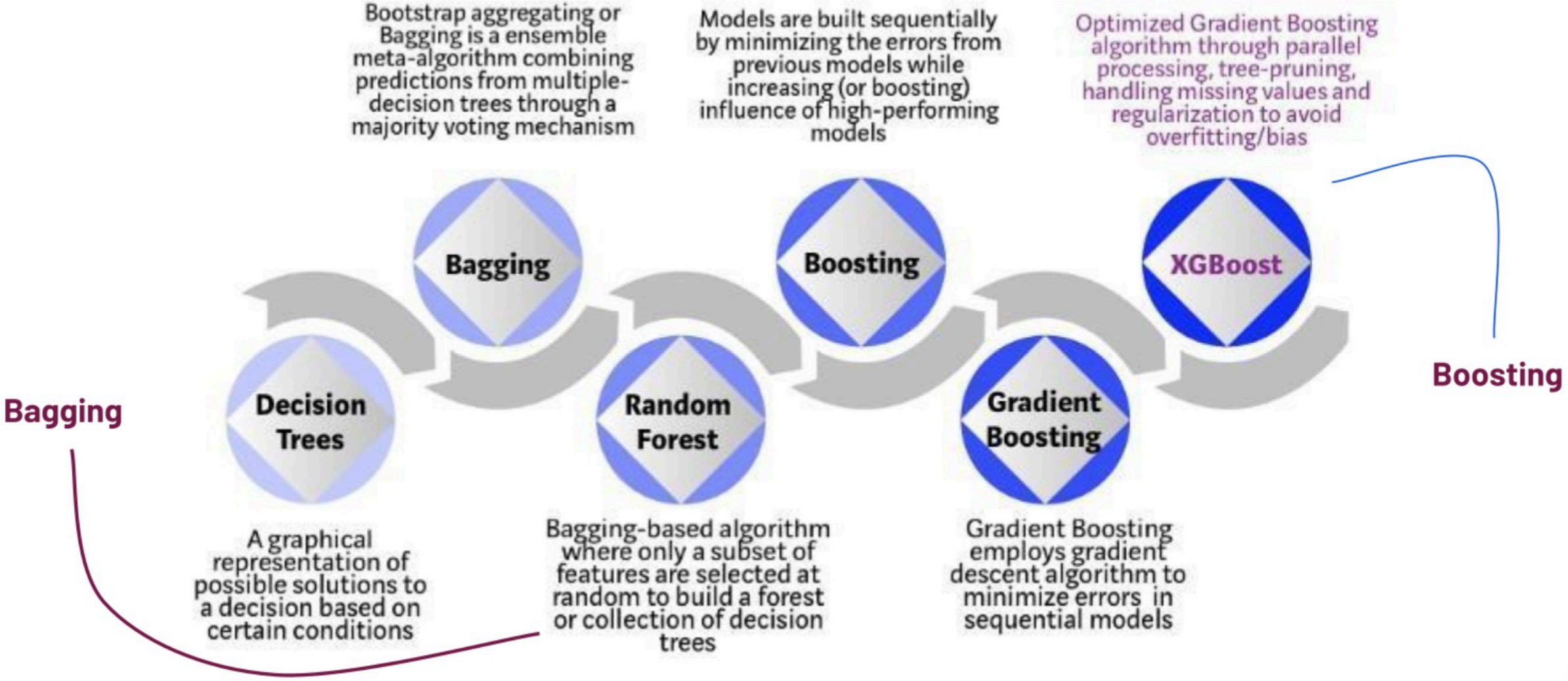
Ensemble Methods





Ensemble Methods

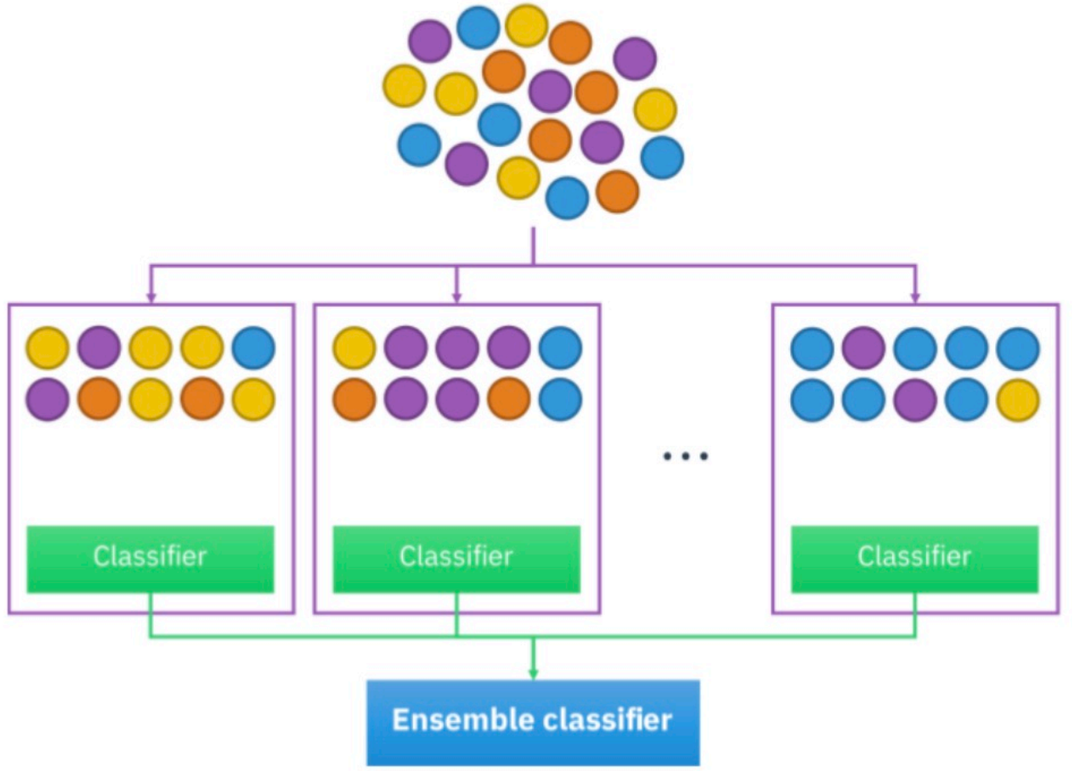
General info about Ensemble Methods: Boosting & Bagging



Ensemble Methods

Bagging (Bootstrap AGGregatING) :

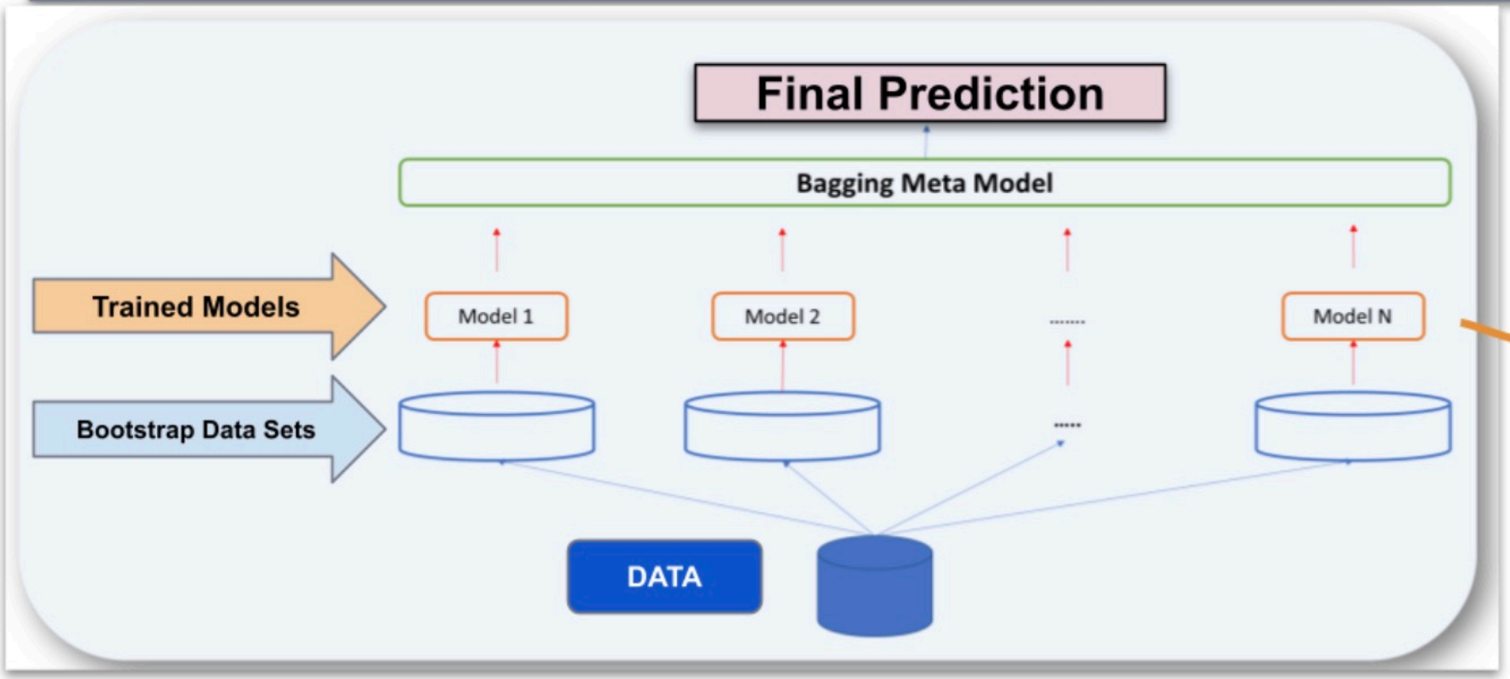
Bagging is used instead of "***bootstrap aggregation***". We can define **Bootstraps** as creating **a new sample data set** using values from the original data set. The clue is that each sub-sample has the same size but different parts of the observations of the original dataset.



Ensemble Methods

Bagging (Bootstrap AGGregatING) :

In bagging methods, **more than one sub dataset created from the data set is trained with the same algorithm**. With these different data sets trained, a final model is created. While creating this model, **voting (mode)** is made for **classification** problems, and **averages (mean)** are taken for **regression** problems.



We can **choose any model as the main algorithm.**

Decision tree*,
Logistic regression,
etc.

scikit learn

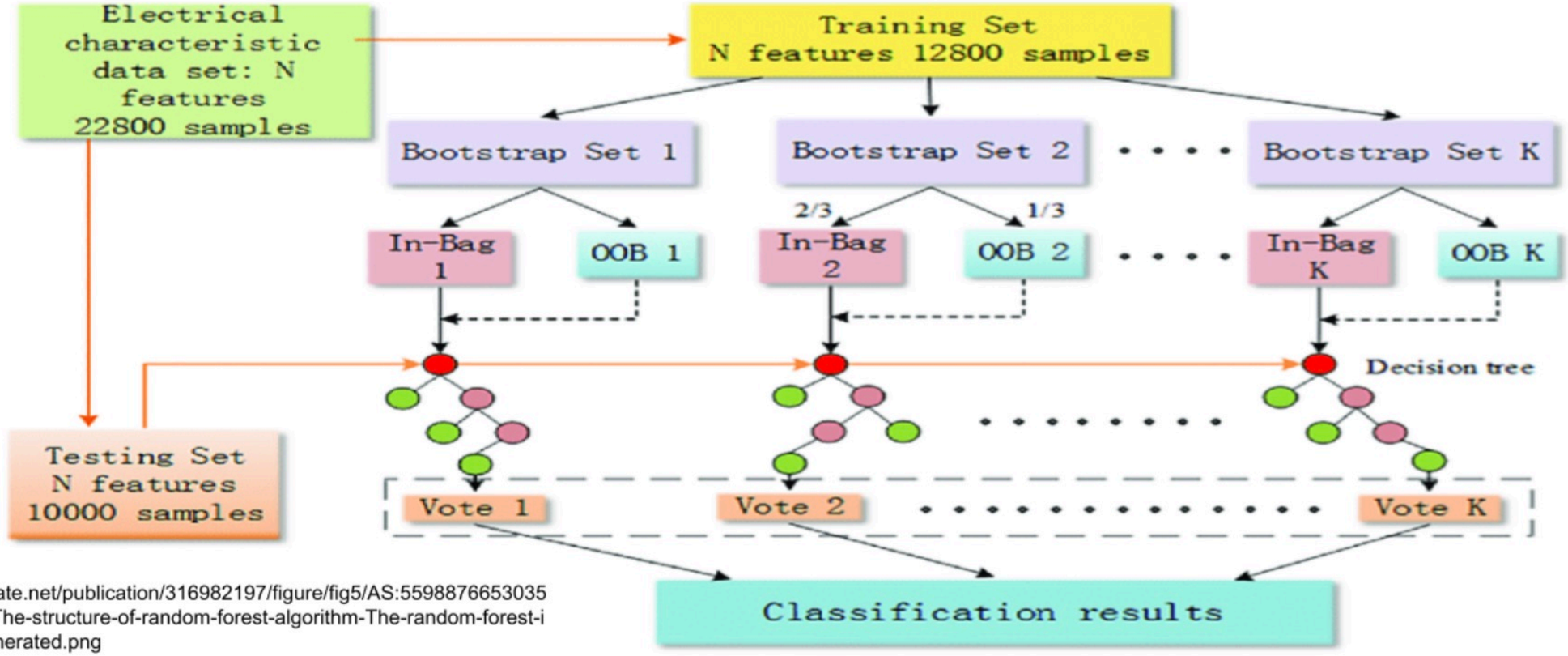
**BaggingClassifier &
BaggingRegressor**

***Random Forest** is a bagging method using Decision Tree as the main algorithm.

Random Forest Algorithm

Random Forest Algorithm:

- Random Forests (RF) is an ensemble bagging algorithm composed of many individual trees (Breiman, 2001)
- RF reduces the variance of individual trees by randomly selecting (and thus de-correlating) many features from the dataset, and averaging them



https://www.researchgate.net/publication/316982197/figure/fig5/AS:559887665303554@1510499029585/The-structure-of-random-forest-algorithm-The-random-forest-is-composed-of-the-generated.png

Random Forest Algorithm

Random Forest Algorithm:

Prevent Overfitting

In addition to taking the random subset of data, as in bagging, ***Random Forest uses a random subset of features rather than using all features*** *to grow trees.*

Hyperparameters:

"n_estimators" *parameter:* (default=100)

The number of trees in the forest.

The more # of trees, the better accuracy. But CPU intensive.

Random Forest Algorithm

Hyperparameters:

"max_depth" *parameter:* (default=None)

The maximum depth of the tree.

If None, then nodes are expanded until all leaves are pure.

"max_features" *parameter:* (default="sqrt")

Number of features to consider when looking for the best split.

Increase will improve the performance, but results in a correlation between the trees.

Random Forest Algorithm

Hyperparameters:

“bootstrapbool” *parameter:* (default=True)

Whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.

“oob_scorebool” *parameter:* (default=False)

This score can be used as validation score, in case of you have small data set

Whether to use out-of-bag samples to estimate the generalization score. Only available if bootstrap=True.

Random Forest Algorithm

Pros & Cons

Pros:

– Good for multi-class problems

– Excellent Predictive Powers

– Easy Data Preparation

– Ranking for feature importance

– Suitable for large data

– Can handle missing values (but not in Scikit-learn)

– Compare with DT robust to overfitting

Cons:

– **Not good on smaller data sets**

– Training could be computationally expensive for numerous trees

– For Regression, not as good option as Classification

– Lack of interpretability

– Biased towards high cardinality categorical features

RF Algorithm

Be ready for

RF

Python 1 & 2

Session