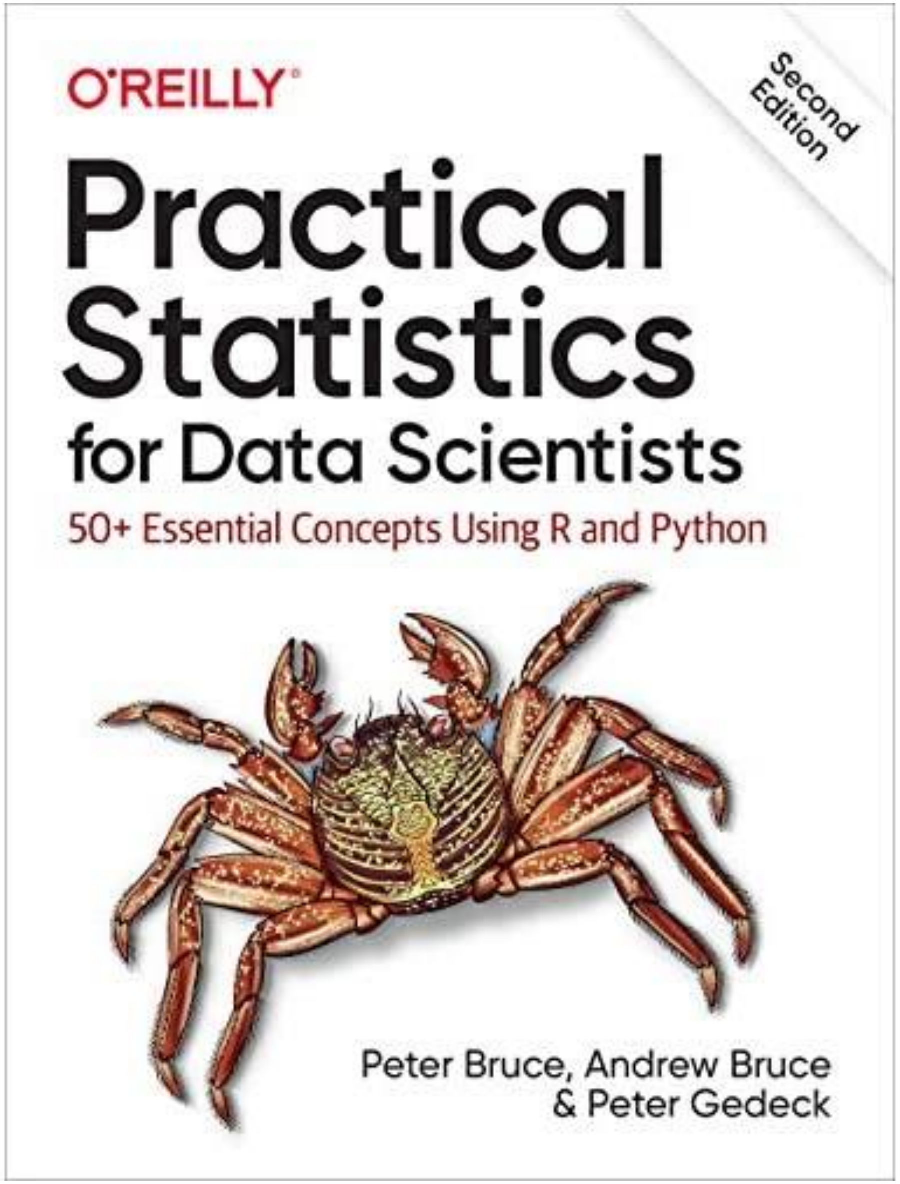
CHAPTER 6
Statistical Machine Learning

Part 2

“Unity is strength”



04/2nd/2022

Presented by: Samah EL KAH

Presentation Outline

- Bagging
- Pasting
- Random Forest
- Variable importance
- Hyperparameters

Ensemble learning theory

Key Terms

Ensemble

Forming a prediction by using a collection of models.

Synonym

Model averaging

Weak learners

Models that can be used as building blocks for designing more complex models by combining several of them.

Synonym

Base models

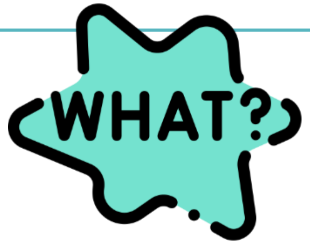
Bagging

A general technique to form a collection of models by bootstrapping the data.

Synonym

Bootstrap aggregation

Introductory example



Online reviews

checking for their features, specifications

make a decision considering the opinions of other people as well.

Ask friends & colleagues opinion

Compare prices

Ensemble learning theory

Ensemble methods are meta-algorithms that **associate** a variety of machine learning methods into a single predictive model to obtain better results.

decrease variance by means of **bagging**, decrease bias by means of **boosting**, or improve predictions by means of **stacking**.

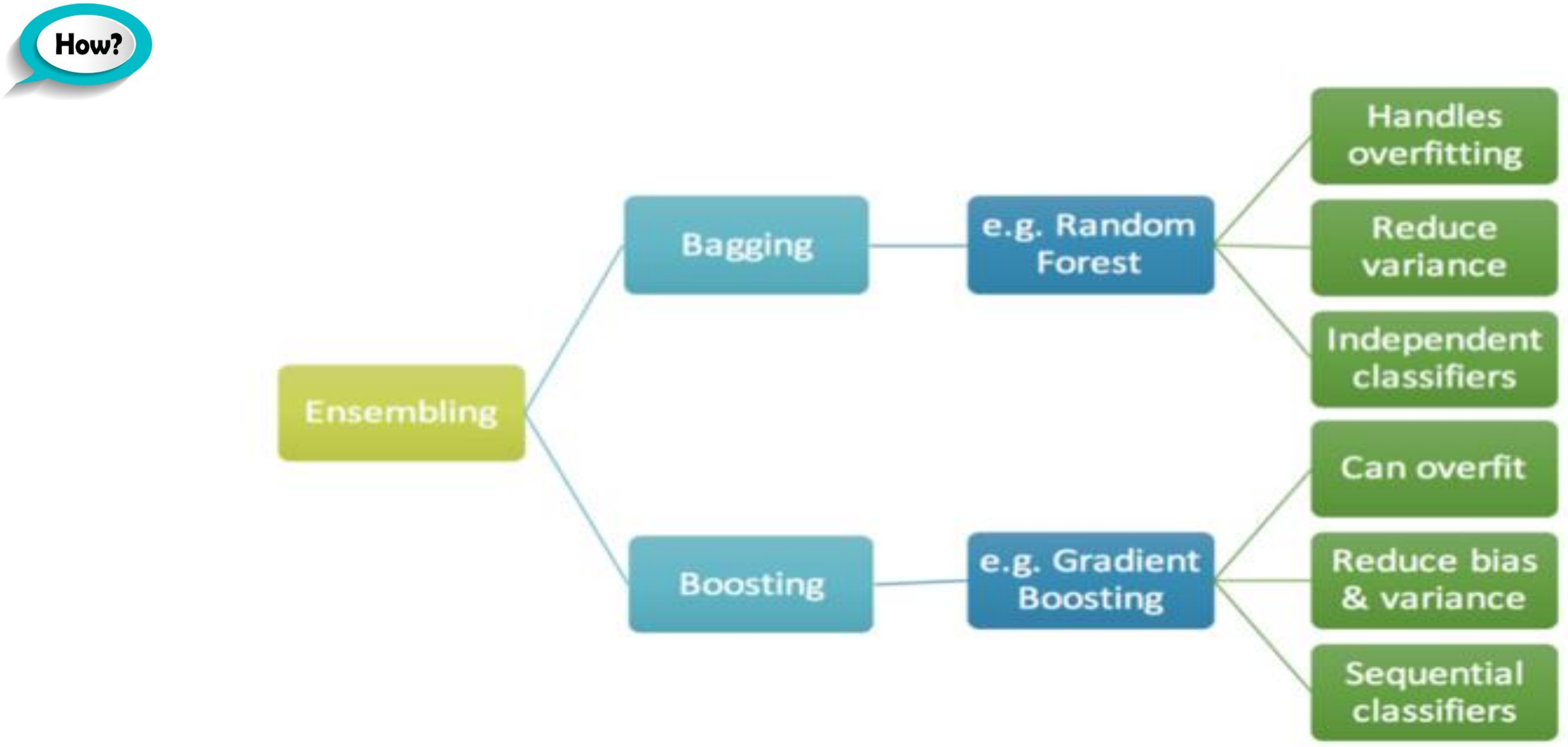
Ensemble learning theory

There are two main reasons to use an ensemble over a single model, and they are related; they are:

Performance: An ensemble can make better predictions and achieve better performance than any single contributing model.

Robustness: An ensemble reduces the spread or dispersion of the predictions and model performance.

Ensemble learning theory



Bagging & Pasting

- Bagging = bootstrap+aggregating and it is a ensemble method in which we first bootstrap our data and for each **bootstrap** sample we **train** one model. After that, we **aggregate** them with equal weights.
- When it's not used replacement, the method is called ***pasting***.

- Example:

We have a set of observations: [2, 4, 32, 8, 16].

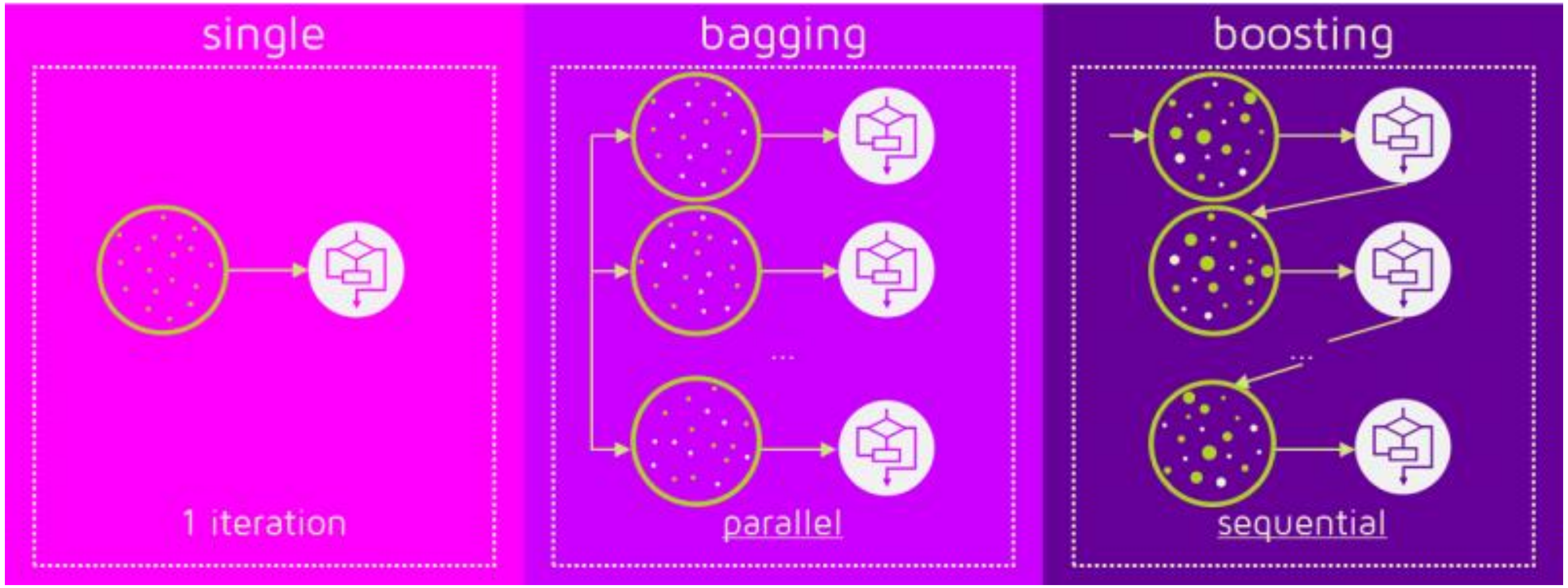
If we want each bootstrap sample containing *n* observations, the following are valid samples:

n=3: [32, 4, 4], [8, 16, 2], [2, 2, 2]…

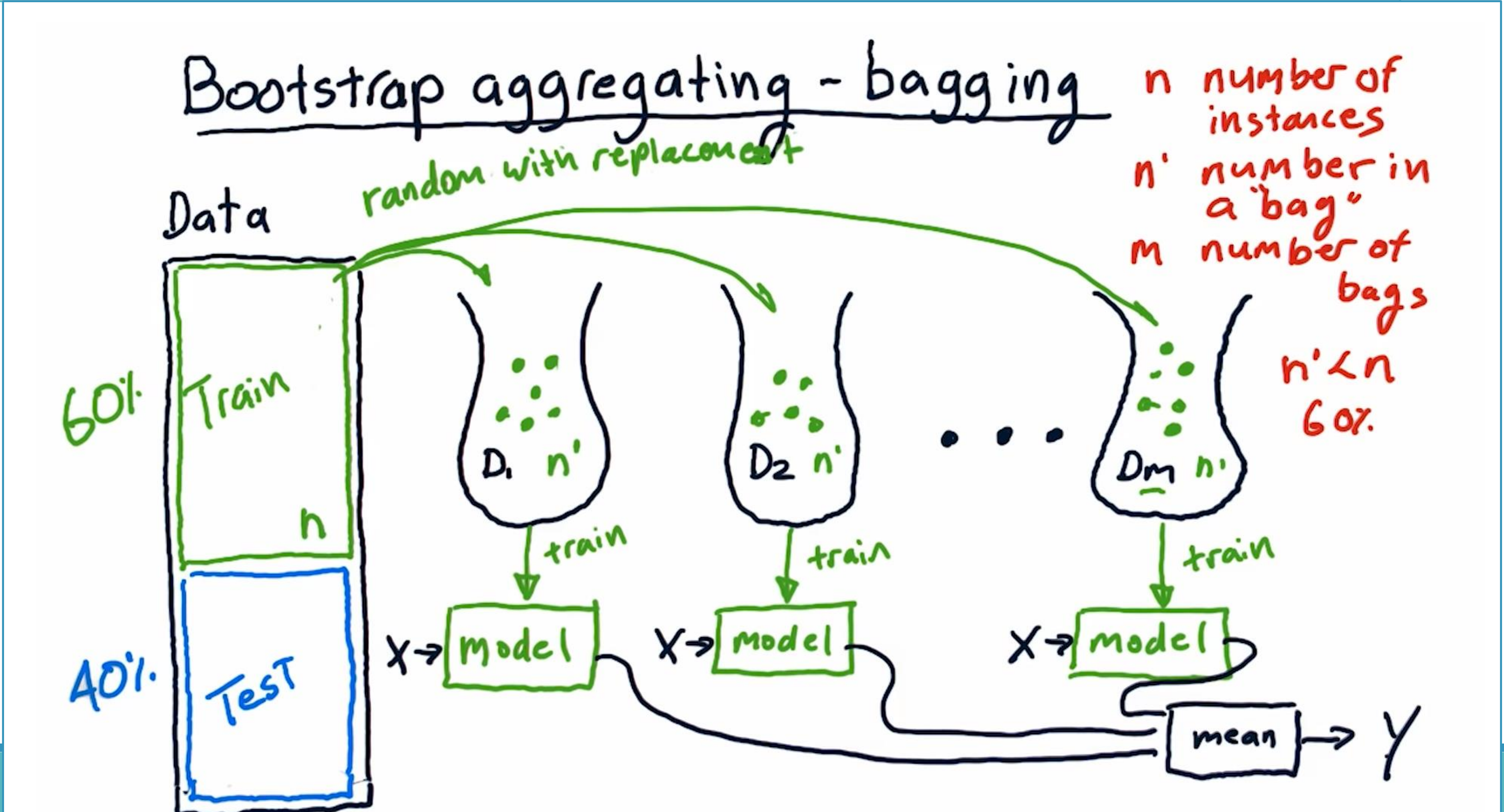
n=4: [2, 32, 4, 16], [2, 4, 2, 8], [8, 32, 4, 2]…

Bagging & Pasting

Bagging, that often considers homogeneous weak learners, learns them independently from each other in **parallel** and combines them following some kind of deterministic averaging process.

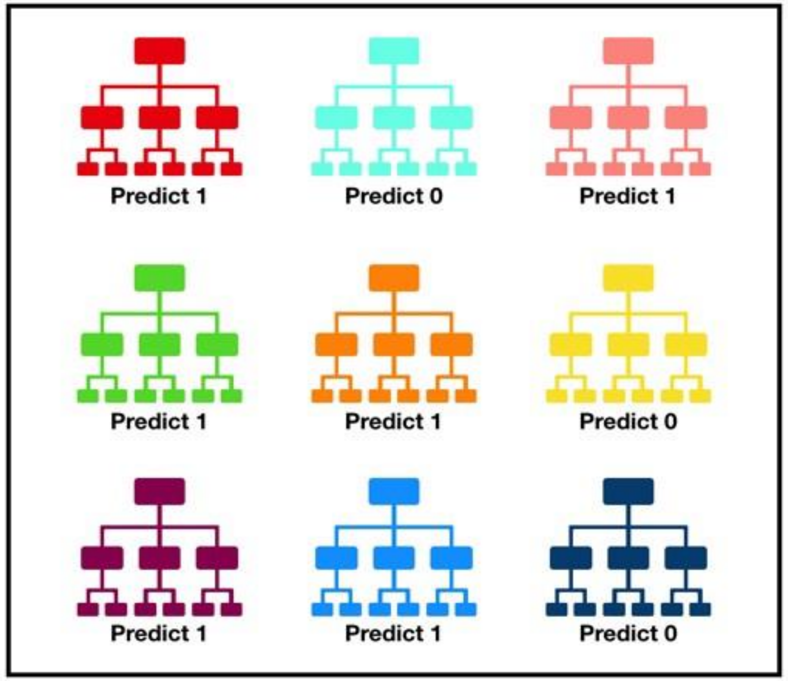


Bagging


Bootstrap aggregating - bagging
random with replacement
Data
60%
Train
n
40%
Test
D1 n'
D2 n'
Dm n'
train
train
train
X → model
X → model
X → model
mean → Y
n number of instances
n' number in a "bag"
m number of bags
n' < n
60%

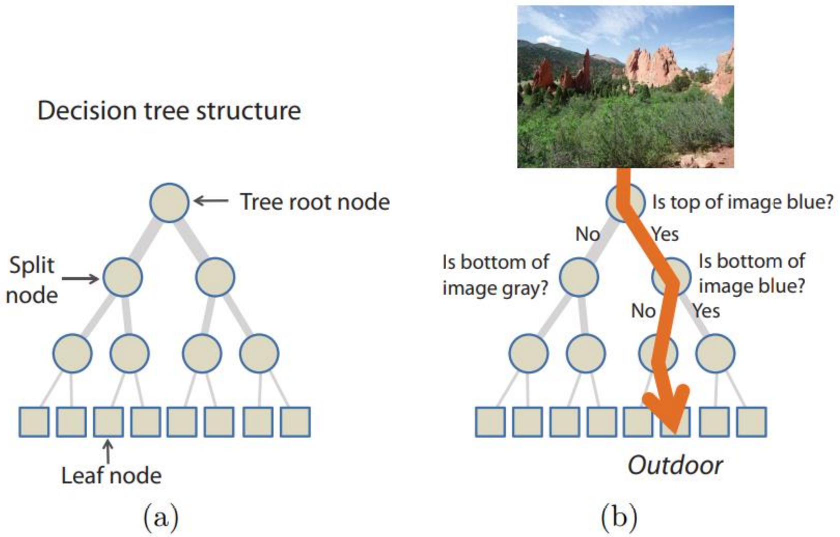
Random Forests

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.



Tally: Six 1s and Three 0s
Prediction: 1

Random Forests



Decision tree. (a) Decision trees contain one root node, internal or split nodes (circles), and terminal or leaf nodes (squares). (b) A pattern arrives at the root and is sequentially passed to one of two children of each split node according to the node-based split function until it reaches a leaf node. Each leaf node is associated with a probability of a specific decision, for example associating a pattern with a class label.

Random Forests

	Instance	Red	Green	Blue	Size (cm)	Fruit (Label)
1	Instance	Red	Green	Blue	Size (cm)	Fruit (Label)
2	0	1.0	0.0	0.0	7.0	Apple
3	1	0.0	1.0	0.0	20	Water Melon
4	2	1.0	0.0	0.0	1.0	Cherry
5	3	0.0	1.0	0.0	7.5	Apple
6	4	1.0	0.0	0.0	1.0	Strawberry
7	5	1.0	0.0	0.0	0.8	Cherry

Figure 1. Example random forest with three decision trees.

Random Forests Algorithm

1. Take the original dataset and create *N* bagged samples of size *n*, with *n* smaller than the original dataset.
2. Train a Decision Tree with each of the *N* bagged datasets as input. But, when doing a node split, don't explore all features in the dataset. Randomly select a smaller number, *M* features, from all the features in training set. Then pick the best split using impurity measures, like Gini impurity or Entropy.
3. Aggregate the results of the individual decision trees into a single output.
4. Average the values for each observation, produced by each tree, if you're working on a Regression task.
5. Do a majority vote across all trees, for each observation, if you're working on a Classification task.

Random Forests

Random forest decision making

The behavior of the random forest decision process depends on

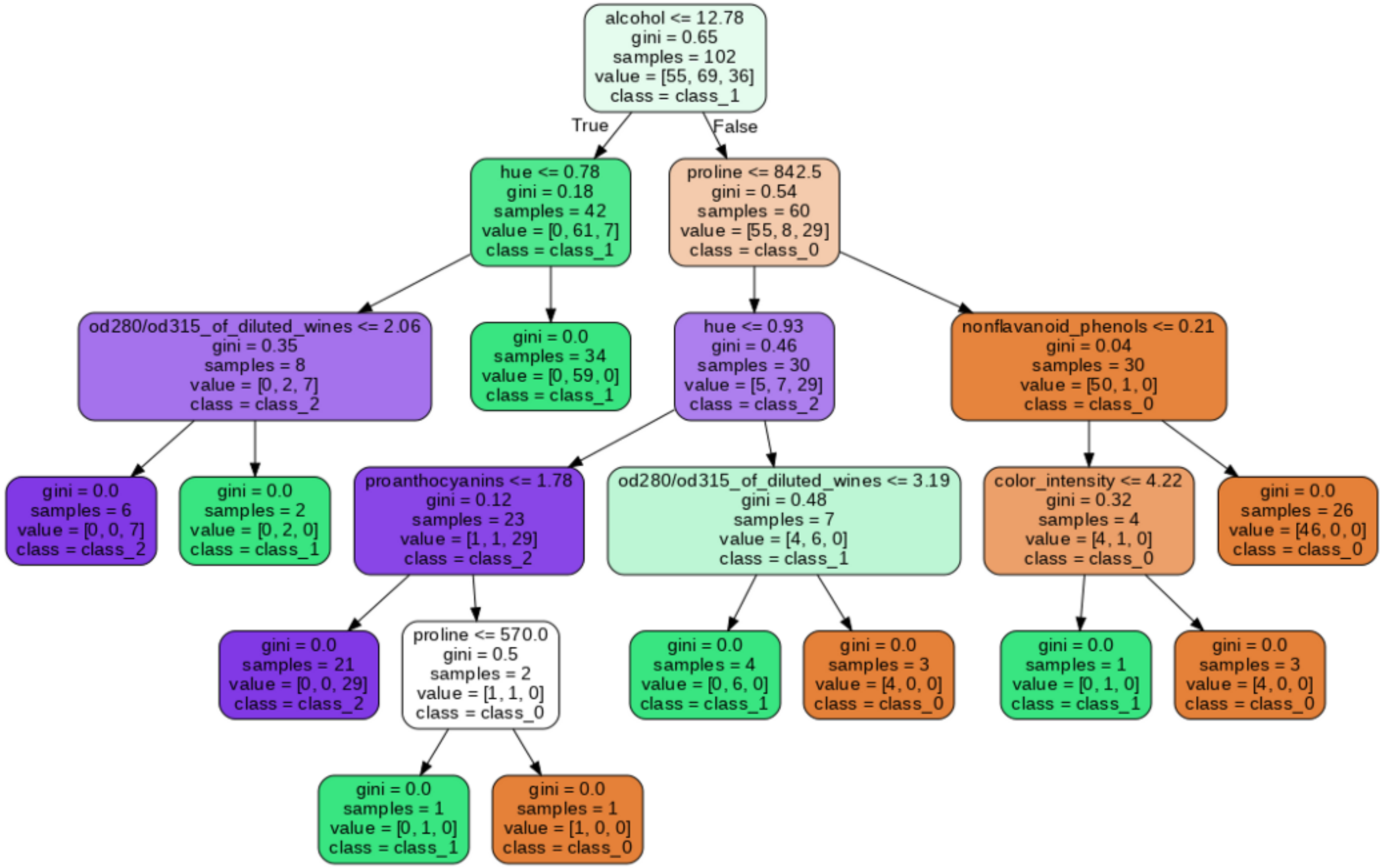
- number T of trees forming the forest
- maximum allowed tree depth D
- parameters of the randomness
- choice of the weak learner model
- objective function used for tree training
- choice of features representing the underlying data

Feature Importance

Feature Importance - ranks variables by their ability to minimize error when split upon, averaged across all trees.

feature importance allows to decide which features to possibly drop because they don’t contribute enough (or sometimes nothing at all) to the prediction process. This is important because a general rule in machine learning is that the more features you have the more likely your model will suffer from overfitting and vice versa.

Feature Importance



Hyperparameters

Hyperparameters is like the settings of an algorithm that can be adjusted to optimize performance, must be set by the data scientist before training.

In the case of a random forest, hyperparameters include the number of decision trees in the forest and the number of features considered by each tree when splitting a node.

nodesize/min_samples_leaf
The minimum size for terminal nodes (leaves in the tree). The default is 1 for classification and 5 for regression in *R*.
maxnodes/max_leaf_nodes
The maximum number of nodes in each decision tree. By default, there is no limit and the largest tree will be fit subject to the constraints of nodesize.

References

1. Book: Hands on Machine Learning
2. https://www.youtube.com/watch?v=2Mg8QD0F1dQ (UDACITY)
3. https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

Randmo Forest

1. https://developer.nvidia.com/blog/accelerating-random-forests-up-to-45x-using-cuml/ (basic example of RF)
2. https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6
3. https://www.youtube.com/watch?v=nxFG5xdpDto (Krish Naik YT Chanel)