Ensemble Learning

- Bias and Variance
- What are Ensemble Learning and their Types
- Bagging Ensemble Learning
- Random Forest
- Boosting Ensemble Learning

Bias and Variance

In general, a machine learning model analyses the data, find patterns in it and make predictions. While training, the model learns these patterns in the dataset and applies them to test data for prediction.

While making predictions, a difference occurs between prediction values made by the model and actual values/expected values, and this difference is known as bias errors or Errors due to bias.

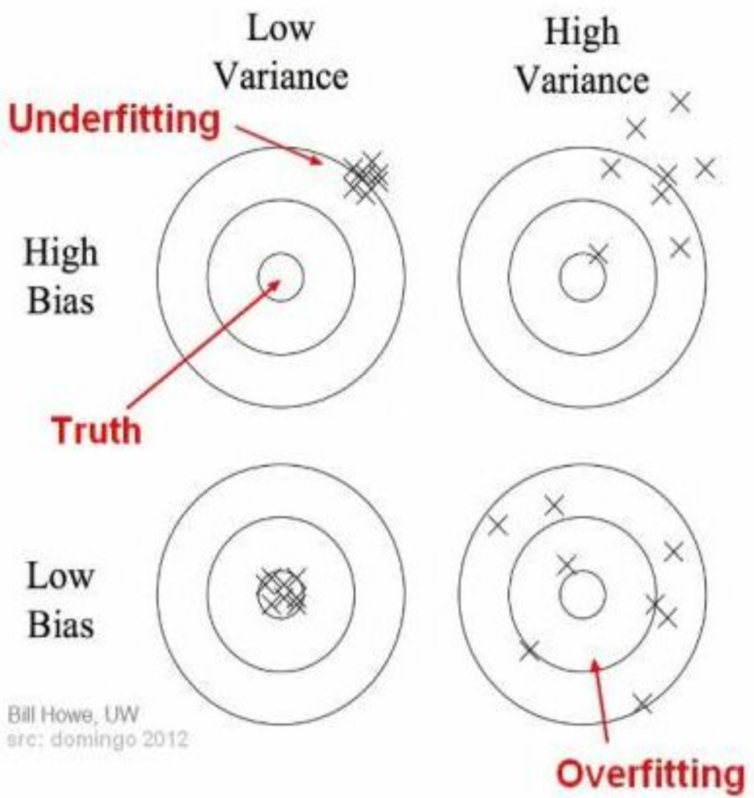
The variance would specify the amount of variation in the prediction if the different training data was used. In simple words, ***Variance tells that how much a random variable is different from its expected value.*** Ideally, a model should not vary too much from one training dataset to another, which means the algorithm should be good in understanding the hidden mapping between inputs and output variables.

- **Low Bias:** A low bias model will make fewer assumptions about the form of the target function.
- **High Bias:** A high bias model will make more assumptions and will not be able to capture the important features of the dataset (Underfitting). A high bias model does not perform well on new data.
- **Low Variance:** There is a small variation in the prediction of the target function with changes in the training data set.
- **High Variance:** There is a large variation in the prediction of the target. A model with high variance performs well with training dataset (Overfitting) but does not generalize well with unseen dataset.

Linear/Logistic Regression models are Low Variance, High Bias whereas

KNN, Decision Tree models are High Variance, Low Bias.

A Good/Ideal Machine Learning model should have Low Bias and Low Variance.



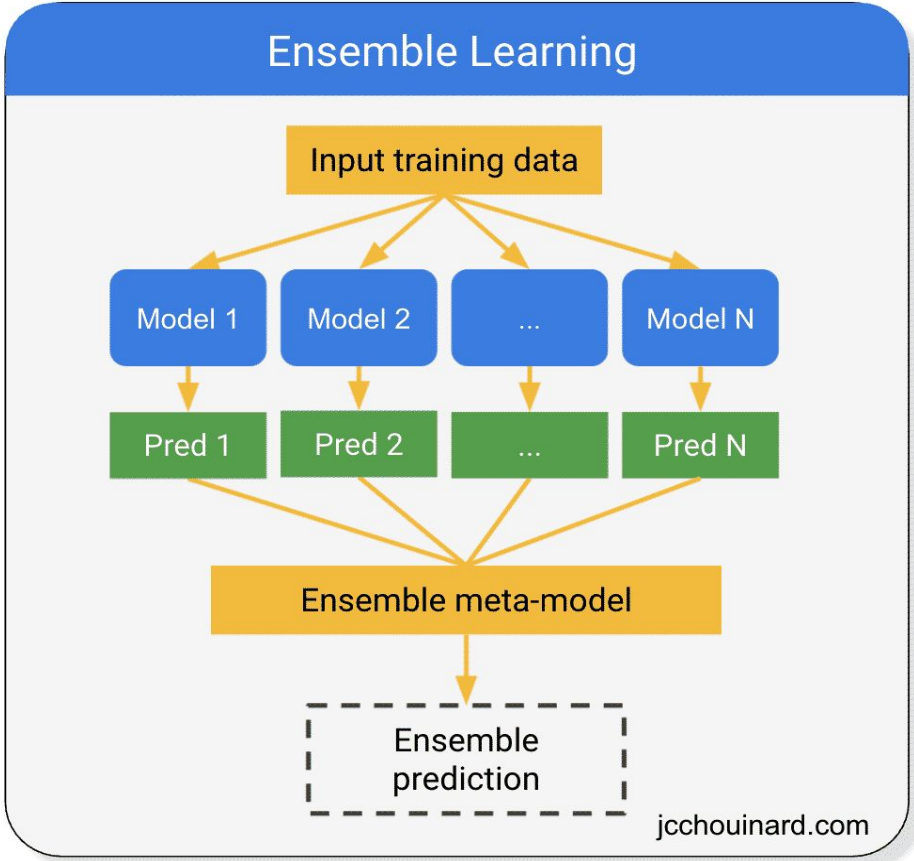
What is Ensemble Learning?

It is based on the principle of **"Wisdom of the Crowd".** It is a technique that combines several base models in order to produce one optimal predictive model.

An ensemble of classifier is a set of classifier whose individual decisions are combined in some way to classify new examples.

Types of Ensemble Learning:

- **Bagging**
- **Boosting**
- **Stacking**
- **Cascading**



- **Base Learner :** An arbitrary learning algorithm which could be used on its own.
- **Ensemble:** A learning algorithm composed of a set of base learners. The base learners may be organized in some structure.

The main purpose of an ensemble is **maximising individual accuracy and diversity.** Since different learner uses different:

- Algorithms
- Hyperparameters
- Representation/Views
- Training Set

Since, no algorithm that is always the most accurate in all situations, we combine them to get better accuracy.

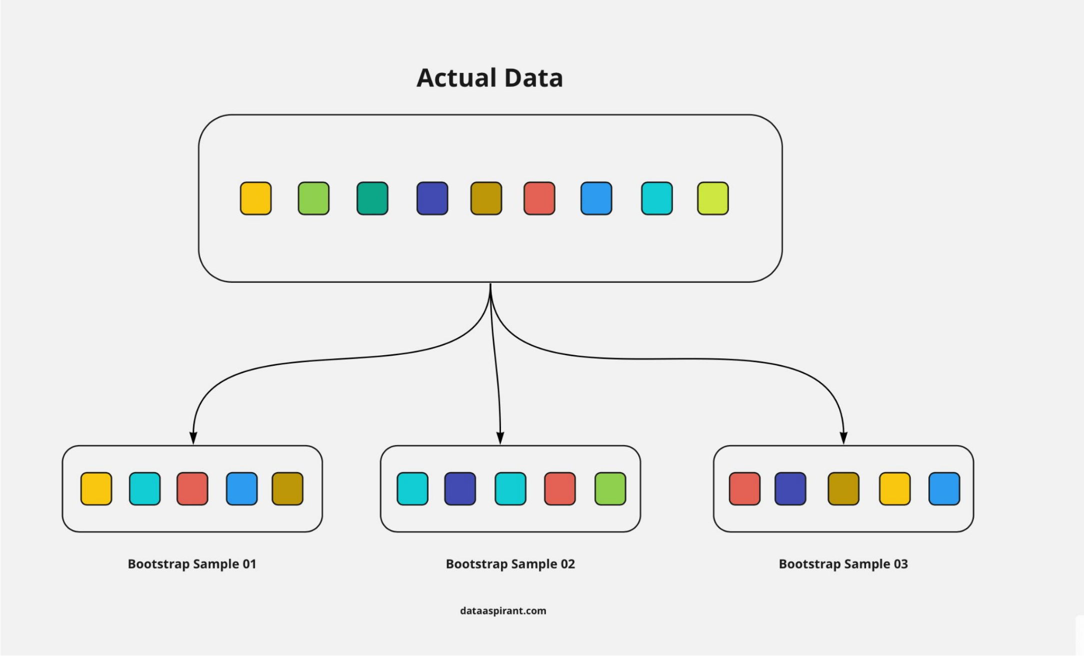
But how do we generate base-learners that complements each other?
How do we combine the outputs of base learners for maximum accuracy?

Bootstrap Sampling

Bootstrap Sampling (also known as Bootstrapping) is a powerful, non-parametric sampling technique. Here, a large number of samples with the **same size** are drawn repeatedly from an original sample(that is **sampling with replacement**).

It is also called as **row sampling** as we are taking a small subset of rows from the original dataset in each sample.

The resulting samples are called as Bootstrap Samples.



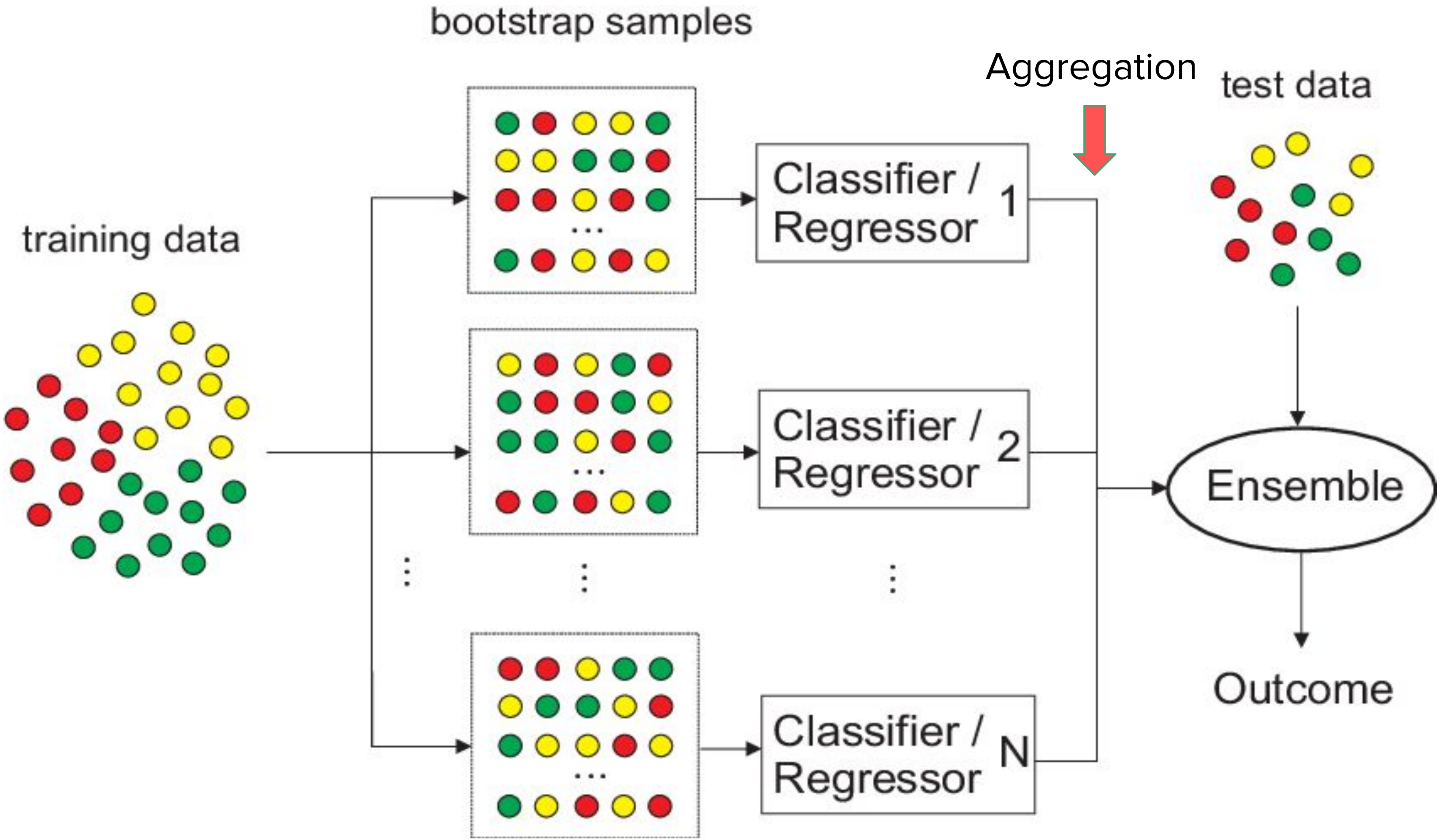
Bagging (Bootstrap **Agg**regation**)**

Bootstrap Aggregation (or Bagging) is a powerful ensemble method to **prevent overfitting.**

The concept behind it is to **combine the prediction of several base learners** to create a more accurate output.

Bagging helps in **increasing accuracy by reducing high variance in base learners.**

- First, we sample the training dataset using Bootstrap Sampling.
- We train a base learner with each sample individually.
- We take the prediction of each base learner and aggregate the result.
  - **For Classification:** We use **Majority Voting.**
  - **For Regression:** We use **Mean/Median.**
- The final ensemble model created has a higher accuracy than the base learners.


bootstrap samples
Aggregation
test data
training data
Classifier / Regressor 1
Classifier / Regressor 2
Classifier / Regressor N
Ensemble
Outcome

Random Forest

It is one of the most popular Bagging technique used in ML. It uses **multiple Decision Trees** to make predictions.

It is called **"Random"** because during Bootstrap Sampling, we take random samples of fixed size and **"Forest"** because we are using a number of Decision trees as base learner.

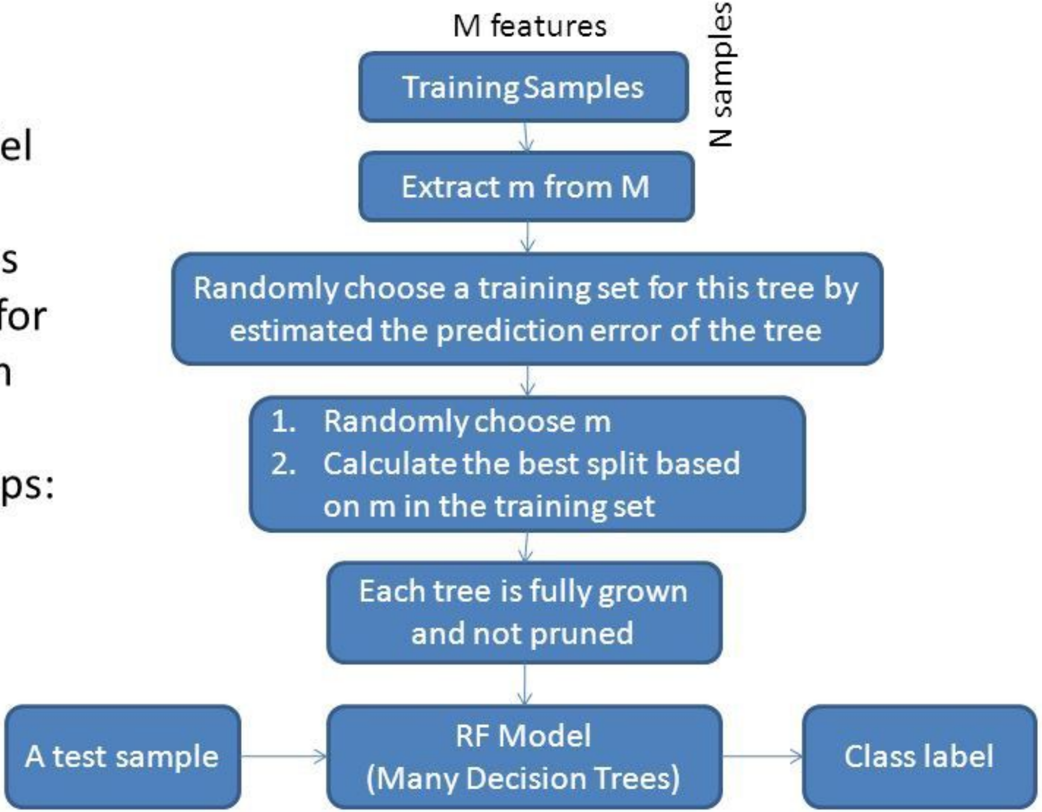
Random Forest = **Decision Trees (High Variance, Low Bias Model)** + **Bagging with Random Bootstrap Sampling** (or row sampling) + **Random Feature Sampling** (or column Sampling)

Feature Sampling involves taking a small subset of features/columns from the training dataset instead of taking all the features at once.

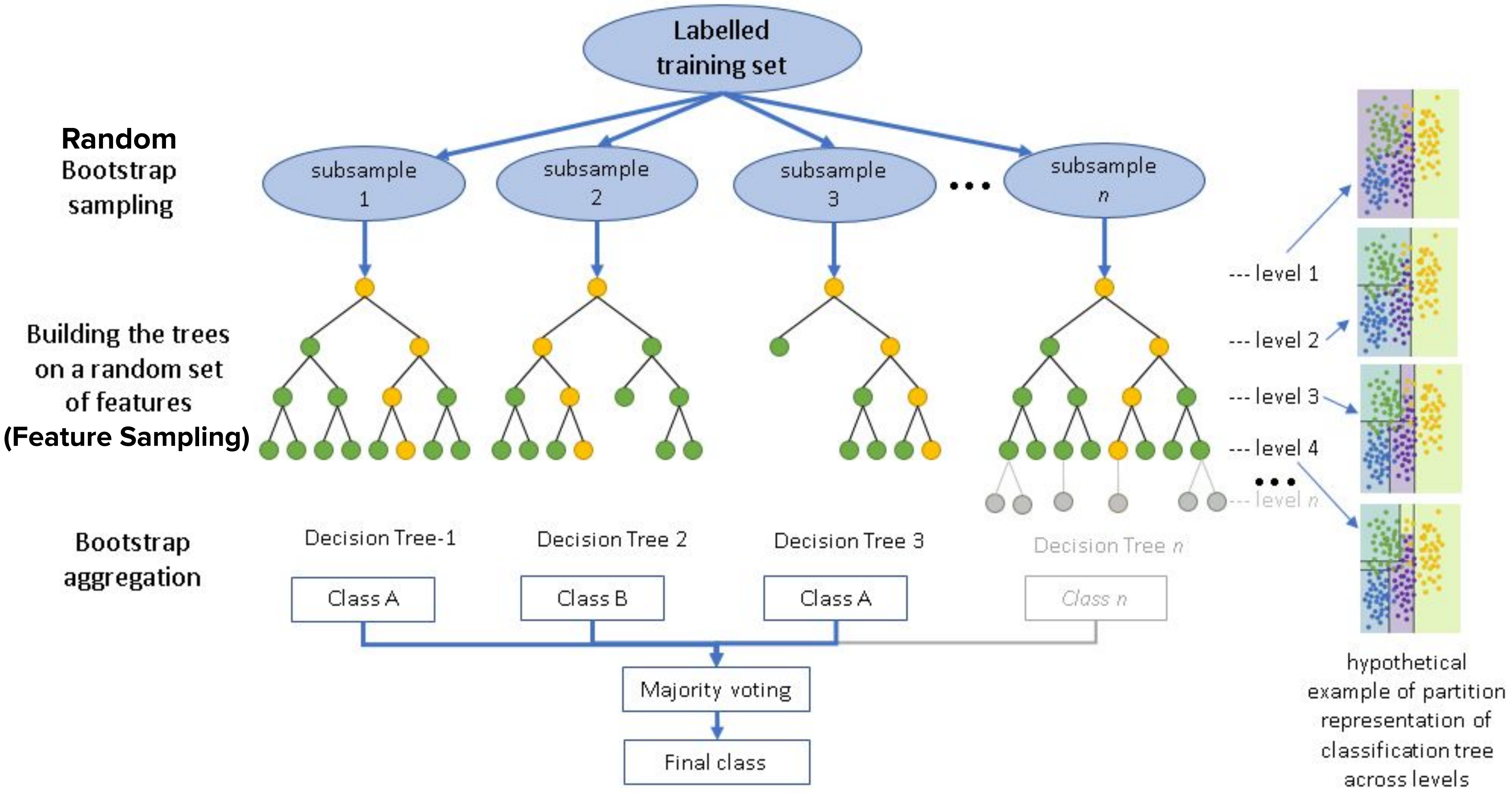
This helps in determining feature importance during training.

Random Forest: Algorithm Steps

To create training model (RF), many decision trees are needed, for each decision tree, we do following steps:



Decision trees


Labelled training set
Random Bootstrap sampling
subsample 1
subsample 2
subsample 3
subsample n
Building the trees on a random set of features (Feature Sampling)
--- level 1
--- level 2
--- level 3
--- level 4
--- level n
Bootstrap aggregation
Decision Tree-1
Decision Tree 2
Decision Tree 3
Decision Tree n
Class A
Class B
Class A
Class n
Majority voting
Final class
hypothetical example of partition representation of classification tree across levels

Boosting

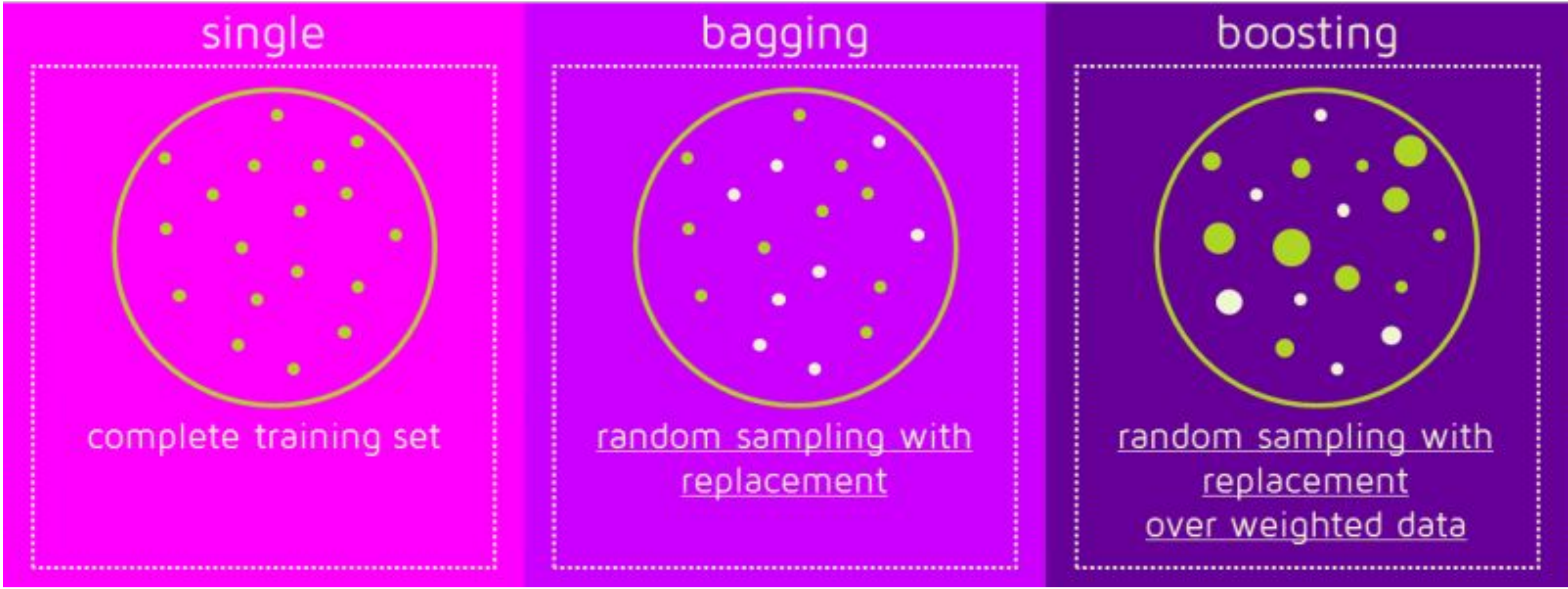
It is another ensemble technique used to **increase accuracy by reducing the bias.** Rather than just combining isolated classifiers, boosting uses the mechanism of increasing the weights of misclassified data in preceding classifiers.

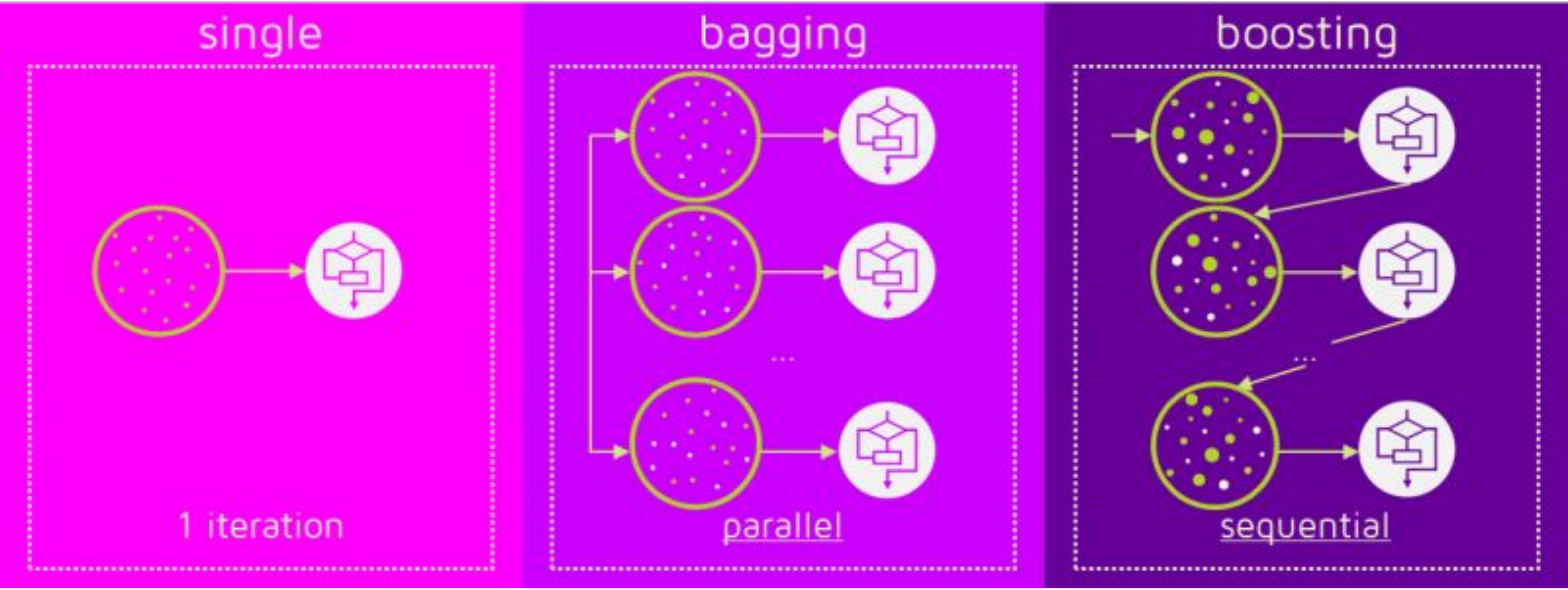
Broadly speaking, It is a 2-step process:

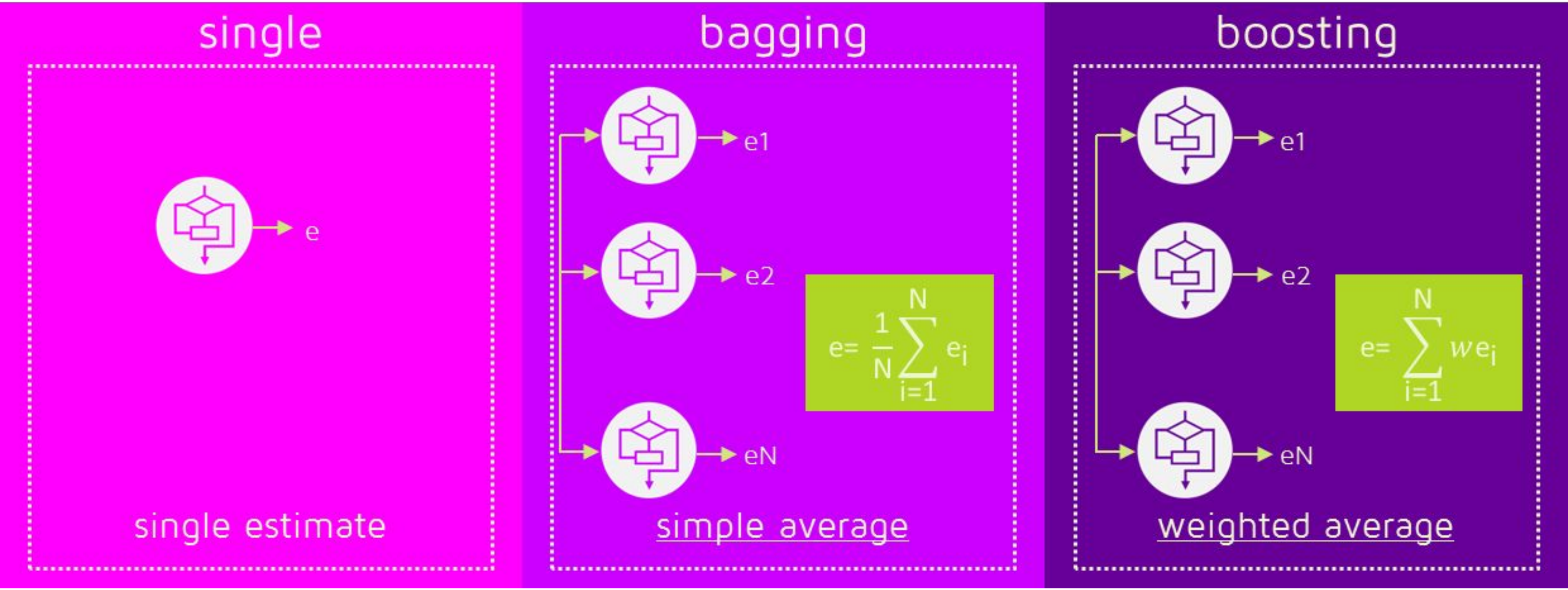
1. Develop averagely performing models (Weak learners) over subset of original data which are weighted (rather than random).
2. Reduce error mainly by developing new learners taking into account the errors of previous learner (Sequential Learning).

Examples:

- **Gradient Boosted Decision Trees (GBDT)**
- **AdaBoost**
- **CatBoost**
- **XGBoost**


single
complete training set
bagging
random sampling with replacement
boosting
random sampling with replacement over weighted data


single
1 iteration
bagging
...
parallel
boosting
...
sequential


single
bagging
boosting
e
e1
e2
eN
e1
e2
eN
$e = \frac{1}{N}\sum_{i=1}^{N} e_i$
$e = \sum_{i=1}^{N} w e_i$
single estimate
simple average
weighted average


single
train
keep
train & keep
bagging
train
keep
train & keep
boosting
train
evaluate
Y
keep?
N
set
weight for
learner i
update
samples'
weights
repeat i
train & evaluate

Comparison

Similarities

Both are ensemble methods to get N learners from 1 learner…

Both generate several training data sets by random sampling…

Both make the final decision by averaging the N learners (or taking the majority of them)…

Both are good at reducing variance and provide higher stability…

Differences

… but, while they are built independently for Bagging, Boosting tries to **add new models** that do well where previous models fail.

… but only Boosting **determines weights** for the data to tip the scales in favor of the most difficult cases.

… but it is an **equally weighted average** for Bagging and a **weighted average** for Boosting, more weight to those with better performance on training data.

… but only Boosting tries to reduce bias. On the other hand, Bagging may solve the **overfitting problem**, while Boosting can increase it.