Ensemble Methods

Tim Pengajaran

Mata Kuliah Machine Learning

Jurusan Teknologi Informasi

Disclaimer

- This presentation material, including examples, images, references are provided for informational and explanation assistance only

- The names of actual products and companies mentioned here in, if any, may be the trademarks of their respective owners

- Credits shall be given to the images taken from the open-source and cannot be used for promotional activities

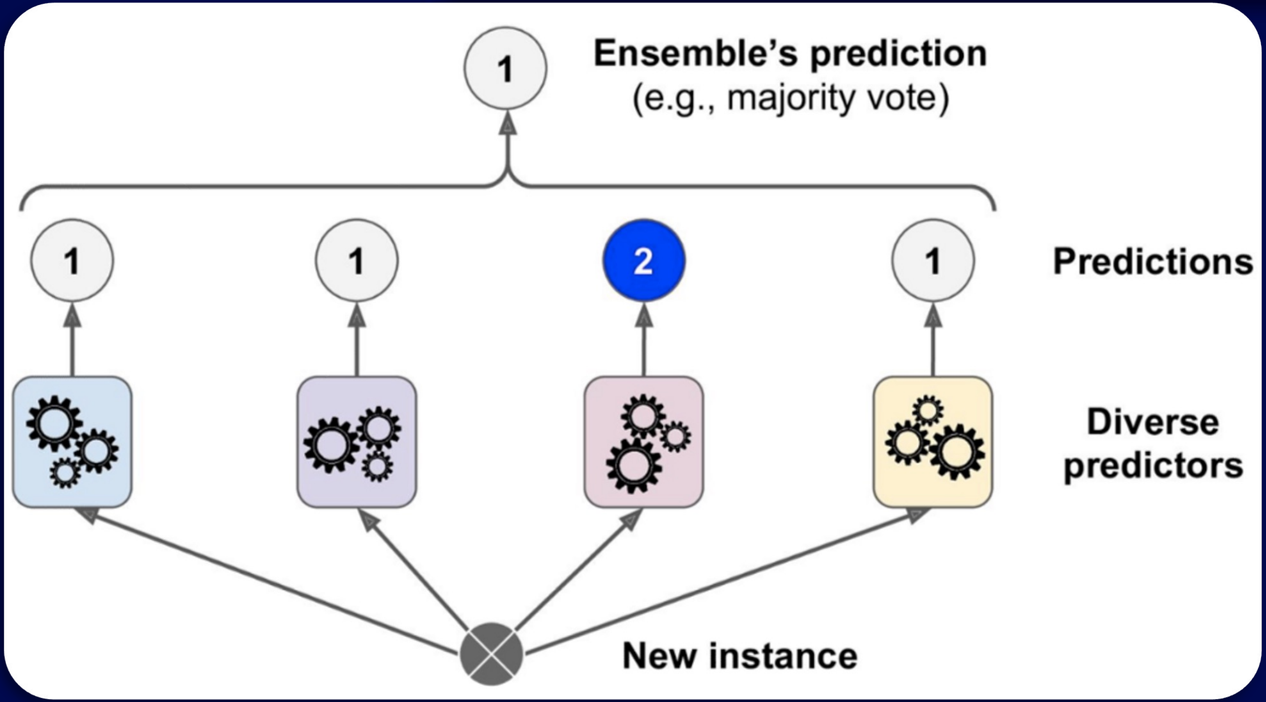
Outline Ensemble Methods

- What is Ensemble Methods?
- Bagging and Random Forest
- Boosting and AdaBoost
- Stacking

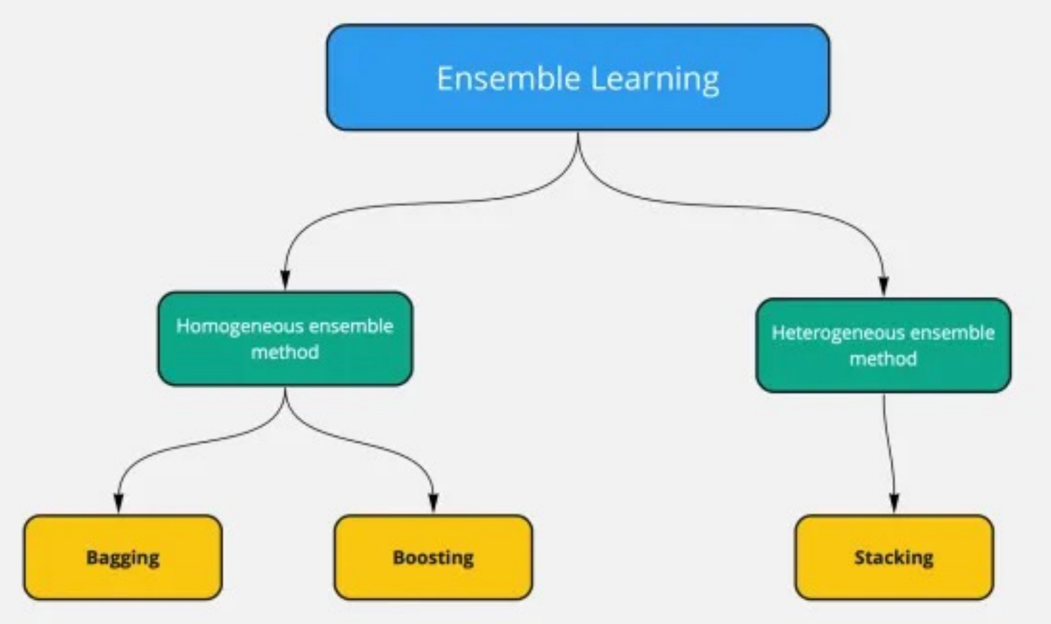
What is Ensemble Methods?

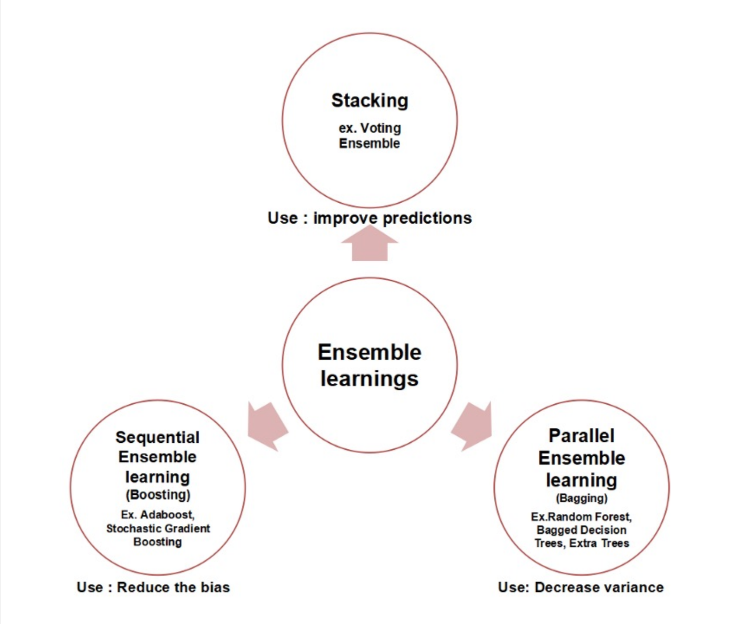
- **A combination of estimators** that performs better than each of its components.
- A machine learning technique that **combines several base models** in order to produce one optimal predictive model.
- Techniques that **create multiple models and then combine them** to produce improved results, usually produces more accurate solutions than a single model would.
- A technique **to combine the predictions of several base estimators** built with a given learning algorithm in order to improve generalizability / robustness over a single estimator.
- The art of **combining diverse set of learners (individual models) together** to improvise on the stability and predictive power of the model.

What is Ensemble Methods?



Types of Ensemble Methods

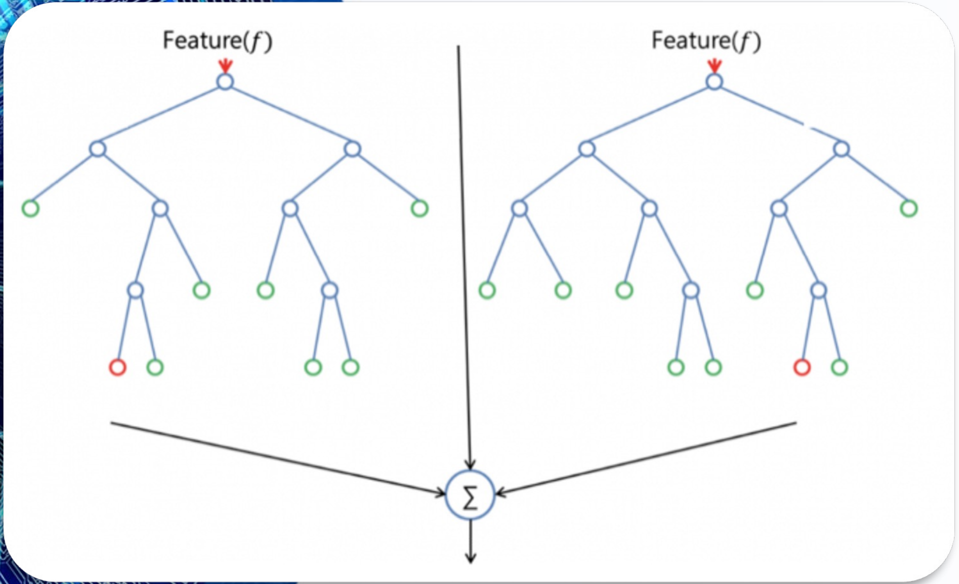




Bagging – Boostrap Aggregating

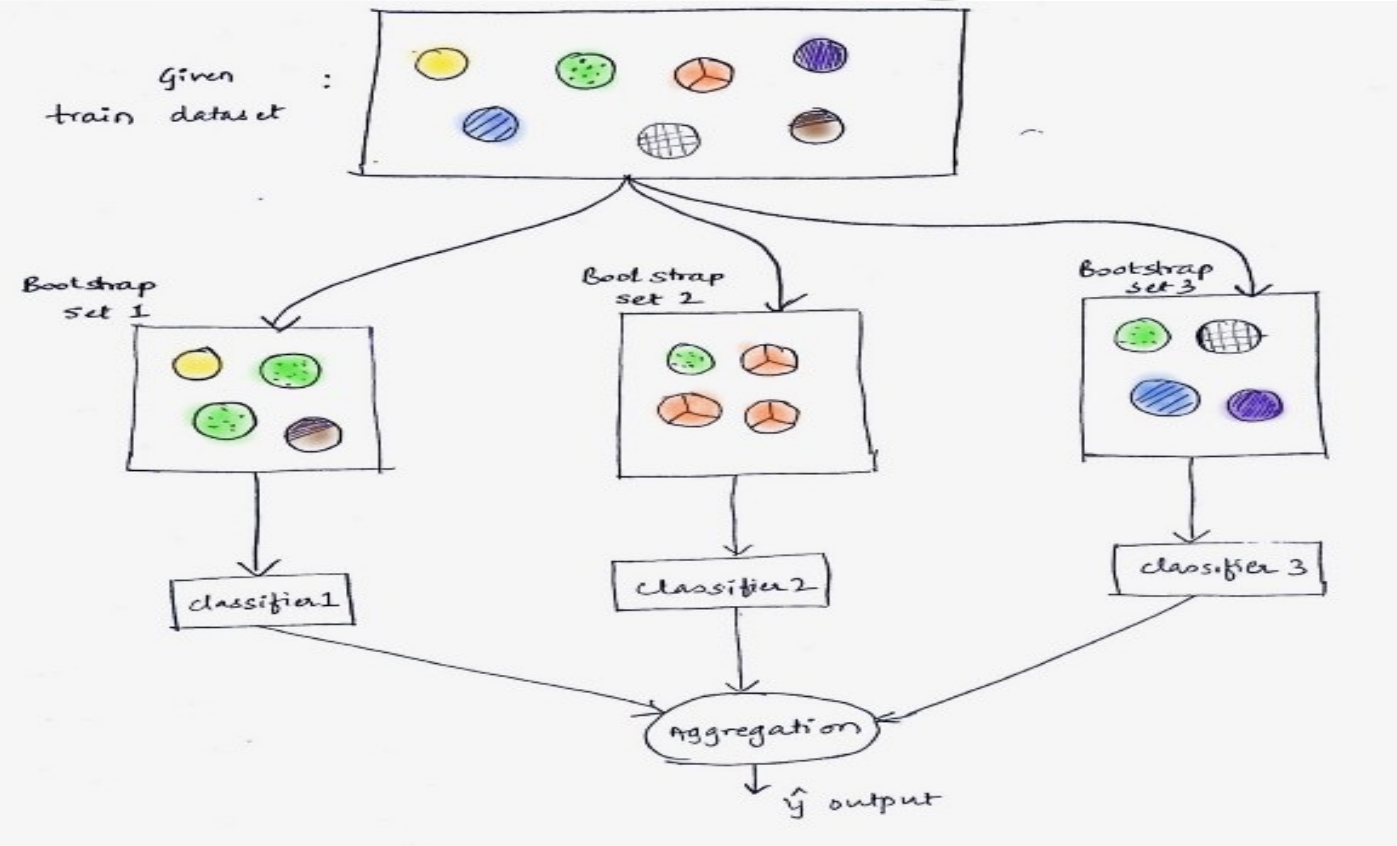
- An ensemble meta-algorithm that can reduce the variance in an estimator.
- Can be used in classification and regression tasks.
  - When the component estimators are **regressors**, the ensemble **averages their predictions**.
  - When the component estimators are classifiers, the ensemble **returns the mode class**.
- A useful meta-algorithm for estimators that **have high variance and low bias**.
- The general idea is that a combination of learning models **increases the overall result**.

Random Forest



- Developed by Leo Breiman and Adele Cutler.
- An ensemble method that **groups multiple Decision Tree predictors**.
- A **supervised learning** algorithm that the "forest" it builds, is an ensemble of decision trees, usually **trained with the "bagging" method**.
- **Parallel** processing.

Random Forest Algorithm



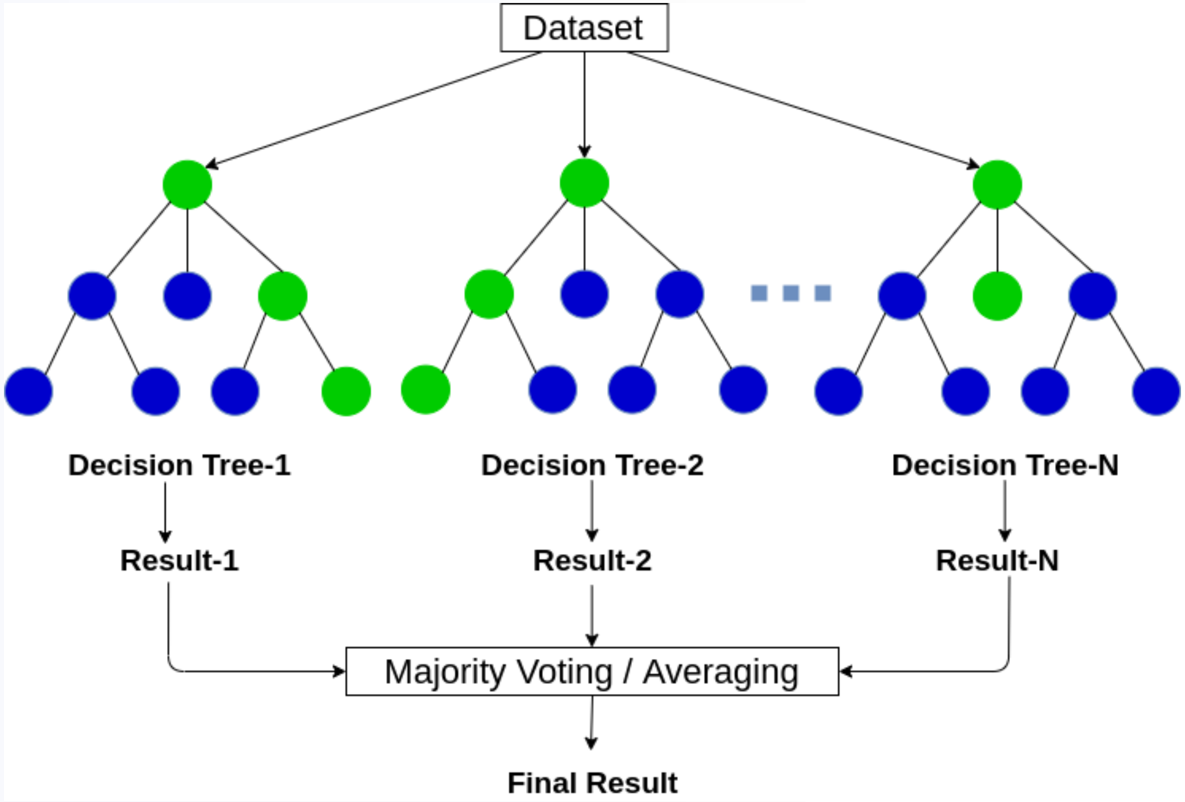
Random Forest Algorithm

Algorithm 1: Pseudo code for the random forest algorithm

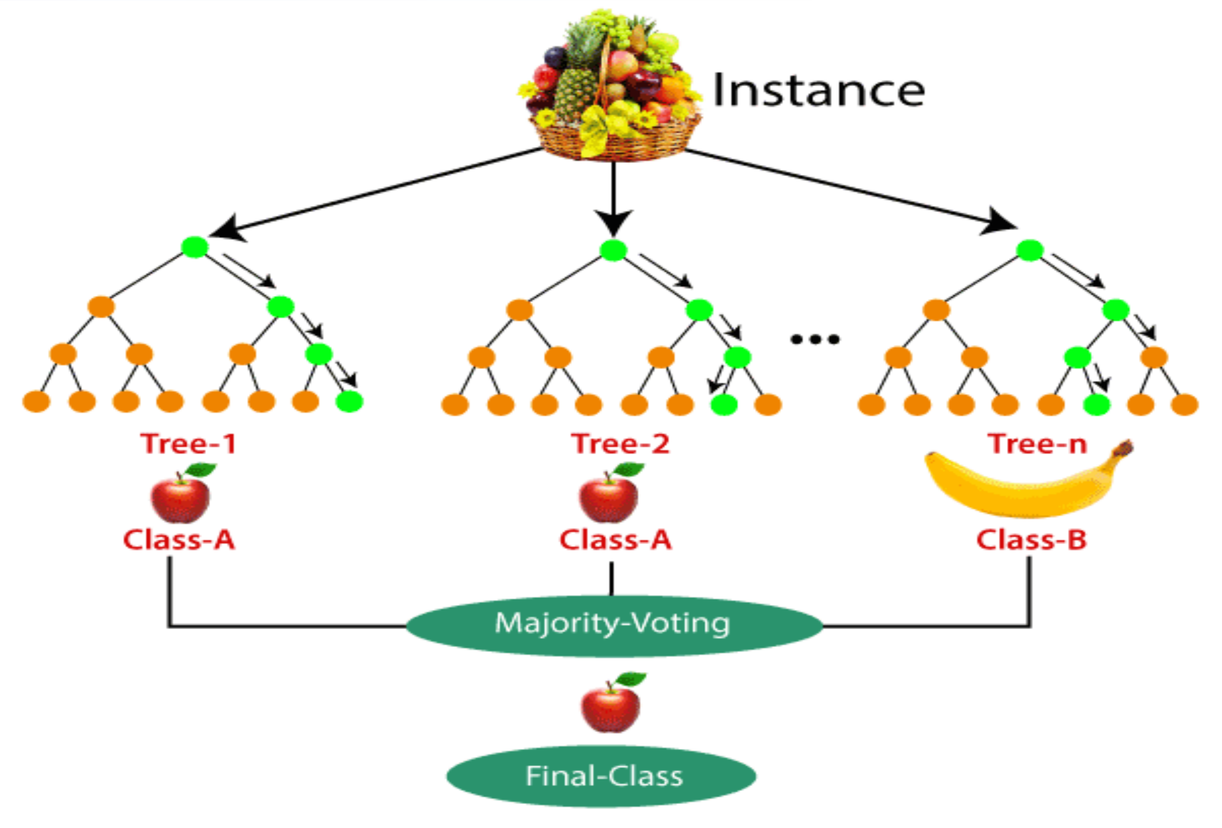
To generate c classifiers:
for $i = 1$ to c **do**
  Randomly sample the training data D with replacement to produce D_i
  Create a root node, N_i containing D_i
  Call BuildTree(N_i)
end for

BuildTree(N):
if N contains instances of only one class **then**
  return
else
  Randomly select x% of the possible splitting features in N
  Select the feature F with the highest information gain to split on
  Create f child nodes of N , $N_1, \ldots, N_f$, where F has f possible values ($F_1, \ldots, F_f$)
  for $i = 1$ to f **do**
    Set the contents of N_i to D_i , where D_i is all instances in N that match
    F_i
    Call BuildTree(N_i)
  end for
end if

Random Forest Algorithm



Random Forest Algorithm



Random Forest Computation – Create Sub-Dataset

Dataset (D)

Insta-nce	Plays fetch	Is grumpy	Favorite food	Animal
1	Yes	No	Bacon	Dog
2	No	Yes	Dog food	Dog
3	No	Yes	Cat food	Cat
4	No	Yes	Bacon	Cat
5	No	No	Cat food	Cat
6	No	Yes	Bacon	Cat
7	No	Yes	Cat food	Cat
8	No	No	Dog food	Dog

Instance	Plays fetch	Is grumpy	Favorite food	Animal
1	Yes	No	Bacon	Dog
2	No	Yes	Dog food	Dog
3	Yes	No	Bacon	Dog
4	No	Yes	Bacon	Cat
5	No	No	Cat food	Cat
6	Yes	No	Bacon	Dog
7	No	Yes	Cat food	Cat
8	No	No	Dog food	Dog

Sub-Dataset (D_1) or Bag 1

Instance	Plays fetch	Is grumpy	Favorite food	Animal
1	Yes	No	Bacon	Dog
2	No	Yes	Dog food	Dog
3	No	Yes	Cat food	Cat
4	No	Yes	Dog food	Dog
5	No	No	Cat food	Cat
6	No	Yes	Bacon	Cat
7	No	Yes	Cat food	Cat
8	No	Yes	Dog food	Dog

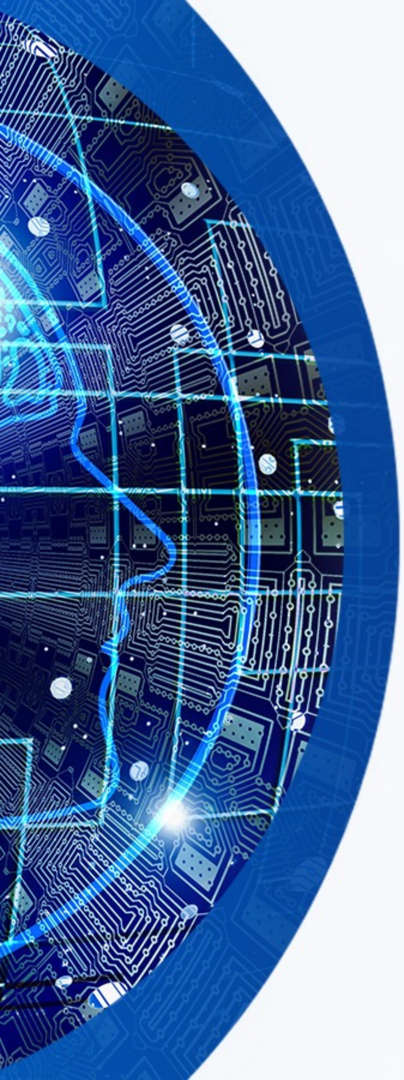
Sub-Dataset (D_3) or Bag 3

Instance	Plays fetch	Is grumpy	Favorite food	Animal
1	Yes	No	Bacon	Dog
2	No	Yes	Dog food	Dog
3	No	Yes	Cat food	Cat
4	No	Yes	Bacon	Cat
5	No	No	Cat food	Cat
6	No	Yes	Cat food	Cat
7	No	Yes	Cat food	Cat
8	No	Yes	Cat food	Cat

Sub-Dataset (D_2) or Bag 2

Create Decision Tree for Each Sub-Dataset

- Use **Information Gain** or **Gini Impurity** to Find the Root Node for Each Tree.
- Do **Decision Tree Procedure** to Create Tree for Each Sub-Dataset.
- **Check** the Created-Trees Performance
  - **Training Error**. Do the Test by Using the trained data, the ones included in Sub-Dataset
  - **Prediction Error**. Do the Test by Using the Out-of-Bag data, the ones not included in Sub-Dataset.
- The **final prediction result** is obtained from majority vote from all Created-Trees results.



Boosting

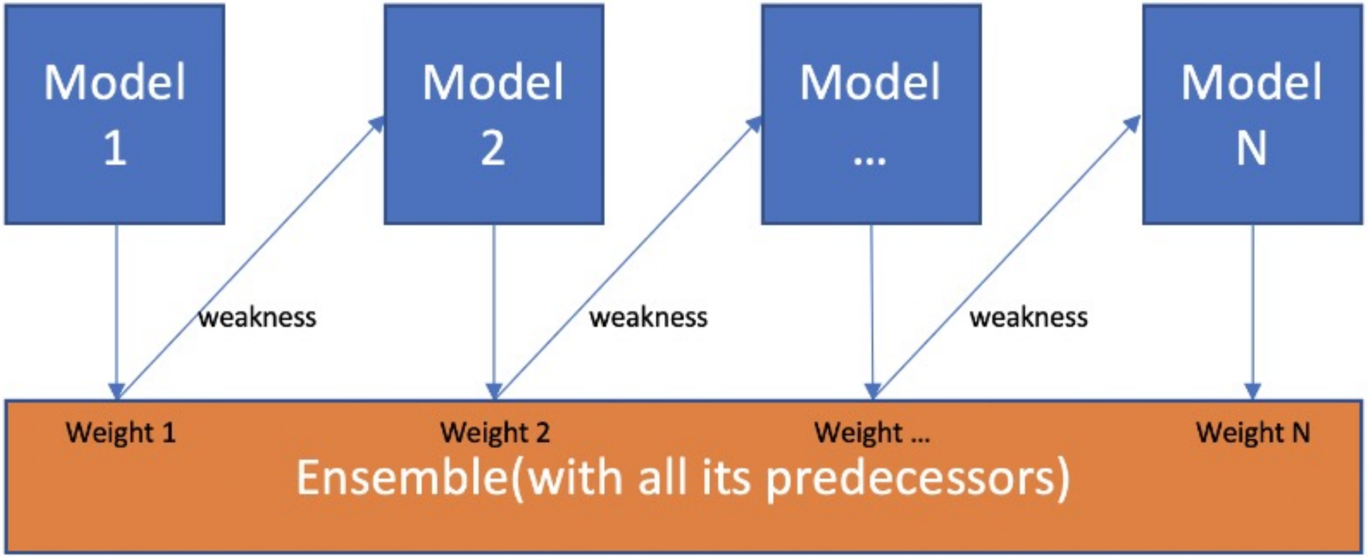
- Formulated by Yoav Freund and Robert Schapire in 1995.
- A family of ensemble methods that are primarily used to **reduce the bias of an estimator**.
- Boosting can be used in **classification and regression** tasks.
- Boosting creates ensembles of **homogeneous estimators**.
- **Sequential** processing.

AdaBoost

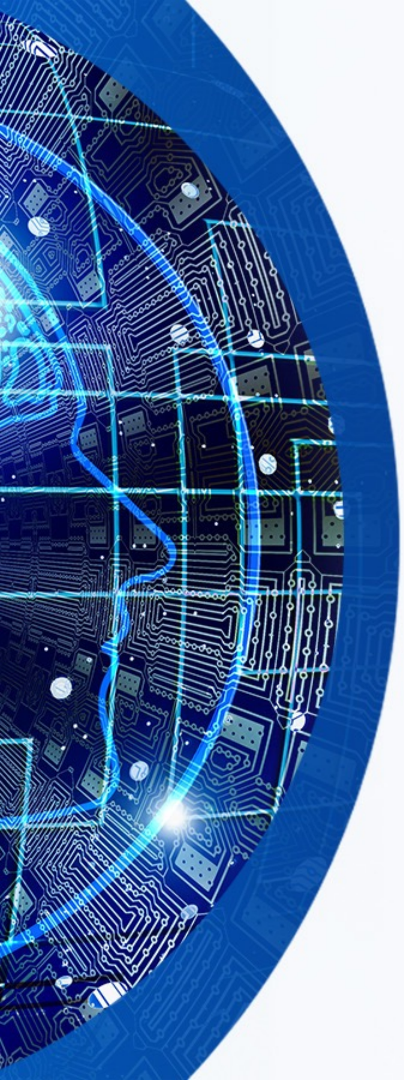
- From **Adaptive Boosting**.
- Two kinds of learner:
  - **Weak learner** (or weak classifier, weak predictor, and so on), is defined only as an estimator that performs slightly better than random chance, such as a decision tree with one or a small number of nodes.
  - **Strong learner** is defined as an estimator that is arbitrarily better than a weak learner.

AdaBoost Mechanisme

Model 1,2,..., N are individual models (e.g. decision tree)



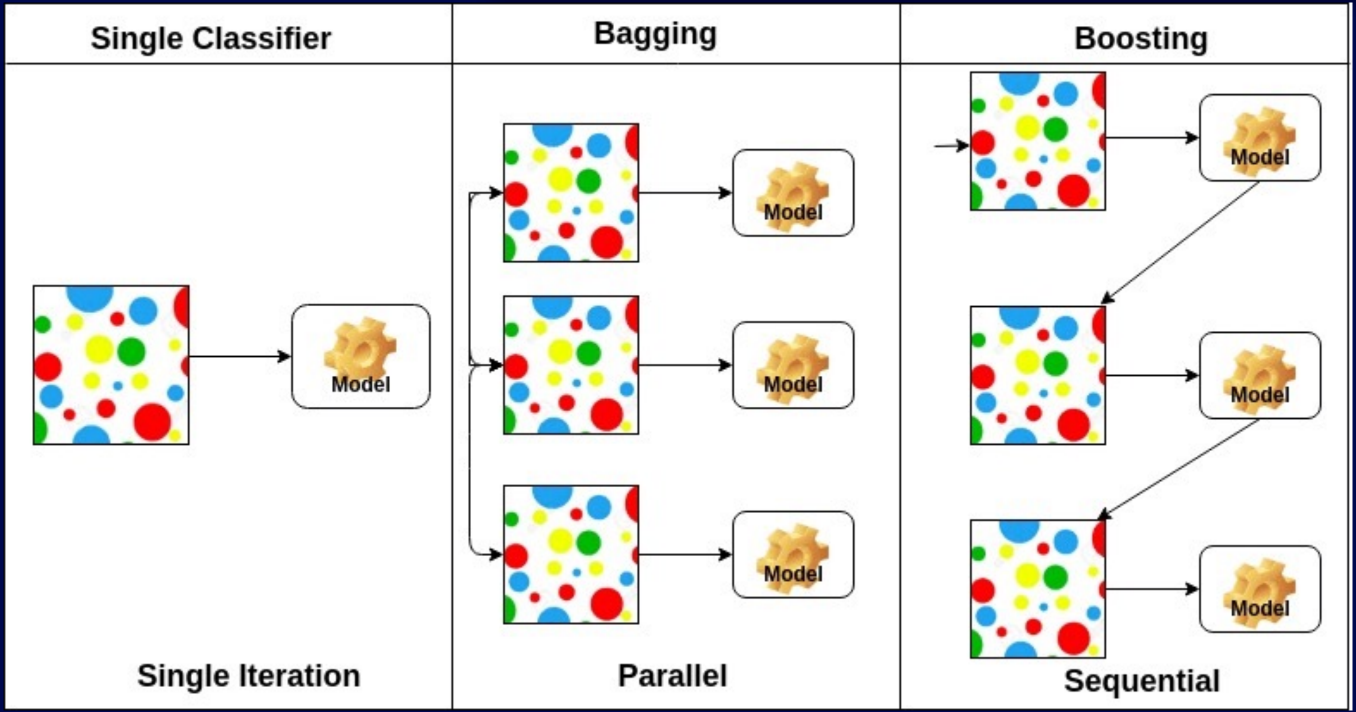
Source: Google



AdaBoost Algorithm

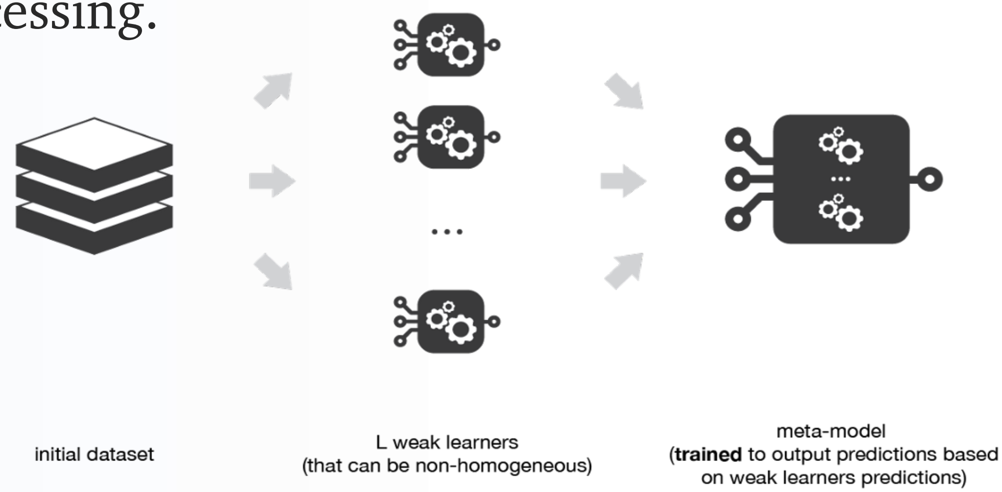
- Creating **First Base Learner**
  - By creating **Stump** for Each Feature. Stump is a tree with only leaves.
  - Assigns equal weights to all of the training instances.
- Calculating the **Total Error (TE)**
  - TE is **the sum of all the errors in the classified record for sample weights**.
- Calculating **Performance of Stump**
  - $Stump_{perf} = \frac{1}{2} log_e \left(\frac{1-TE}{TE}\right)$
- Updating **Weights**
  - For incorrectly classified feature, use: $new_{sample_{weight}} = sample_{weight} * e^{Stump_{perf}}$
  - For correctly classified feature, use: $new_{sample_{weight}} = sample_{weight} * e^{-Stump_{perf}}$
- **Normalize** Weights
  - $Total\_newsampleweight = \sum_{i=1}^{n} new_{sample_{weight}}(i)$, n = the number of sample
  - $Normalize\_weight(i) = \frac{new_{sample_{weight}}(i)}{Total\_newsampleweight}$
- Creating **New Dataset**

Classifier Difference

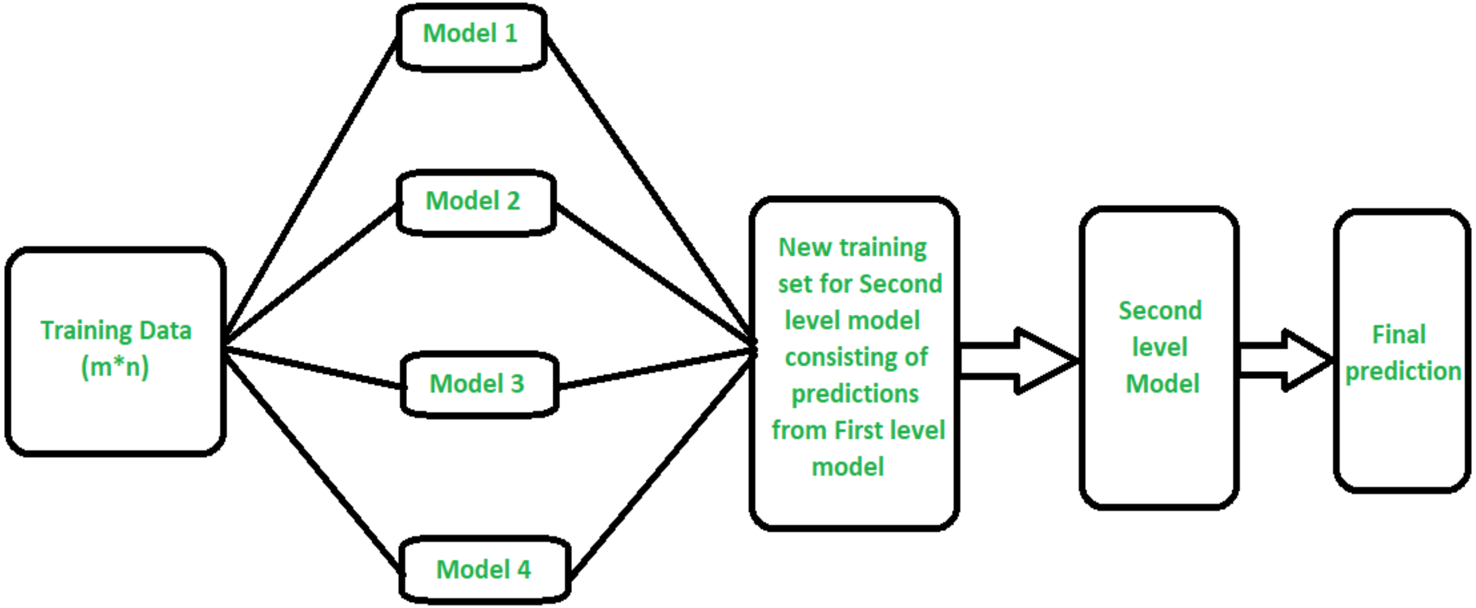


Stacking

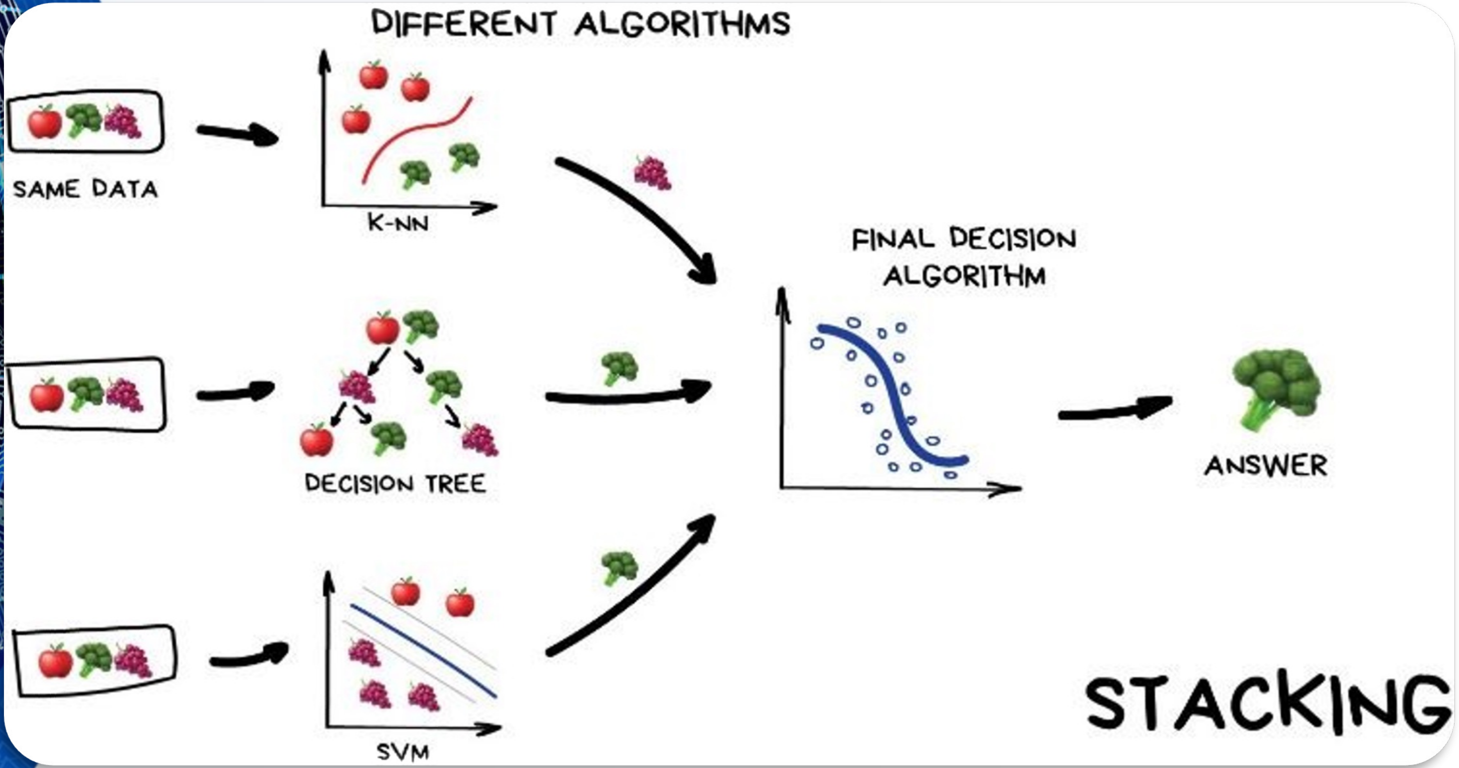
- **Heterogeneous weak learners**, different learning algorithms are combined.
- Learns to **combine the base models** using a meta-model.
- **Parallel** processing.

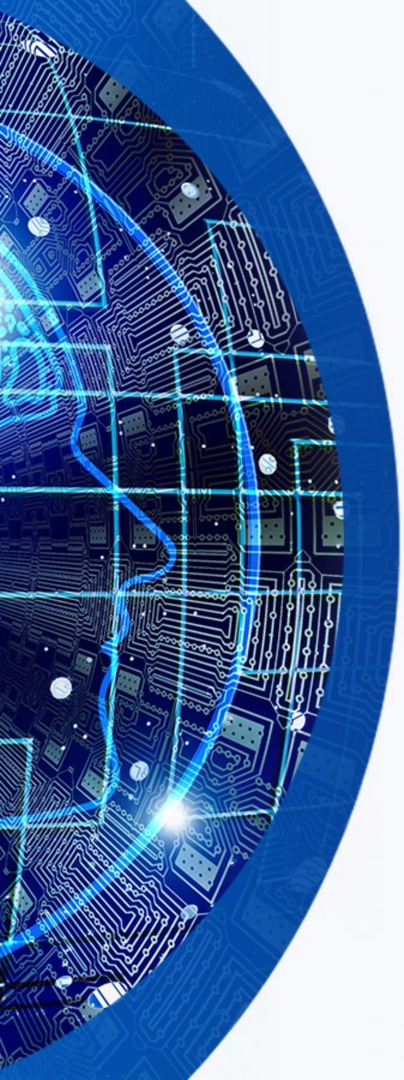


Stacking Architecture



Stacking Example





Stacking Algorithm

- **Split the training data** in two folds (training data and test data).
- Choose **L weak learners** and fit them to data of the first fold.
- For **each of the L weak learners, make predictions** for observations in the second fold.
- Fit the **meta-model on the second fold, using predictions** made by the weak learners as inputs.

Homework

Person ID	Age	Income	Credit Rating	Buys Car
1	25-30	High	Fair	No
2	25-30	High	Excellent	No
3	>40	Medium	Fair	Yes
4	>40	Low	Fair	Yes
5	31-40	Low	Excellent	Yes
6	31-40	Medium	Excellent	Yes
7	>40	High	Excellent	Yes

- Your jobs are:
  1. **Create Ensemble Method structure** completed with all the manual computations using spreadsheet. The rule, student number:
     - 1 – 8 use Bagging.
     - 9 – 17 use AdaBoost.
     - 18 – 26 use Stacking.
  2. **Create computer program** for solving this problem using Python or combined with scikit-learn.
  3. Do it **yourself**.
  4. **Do not plagiat**.

Thank You for Today
Always Keep Your Spirit!