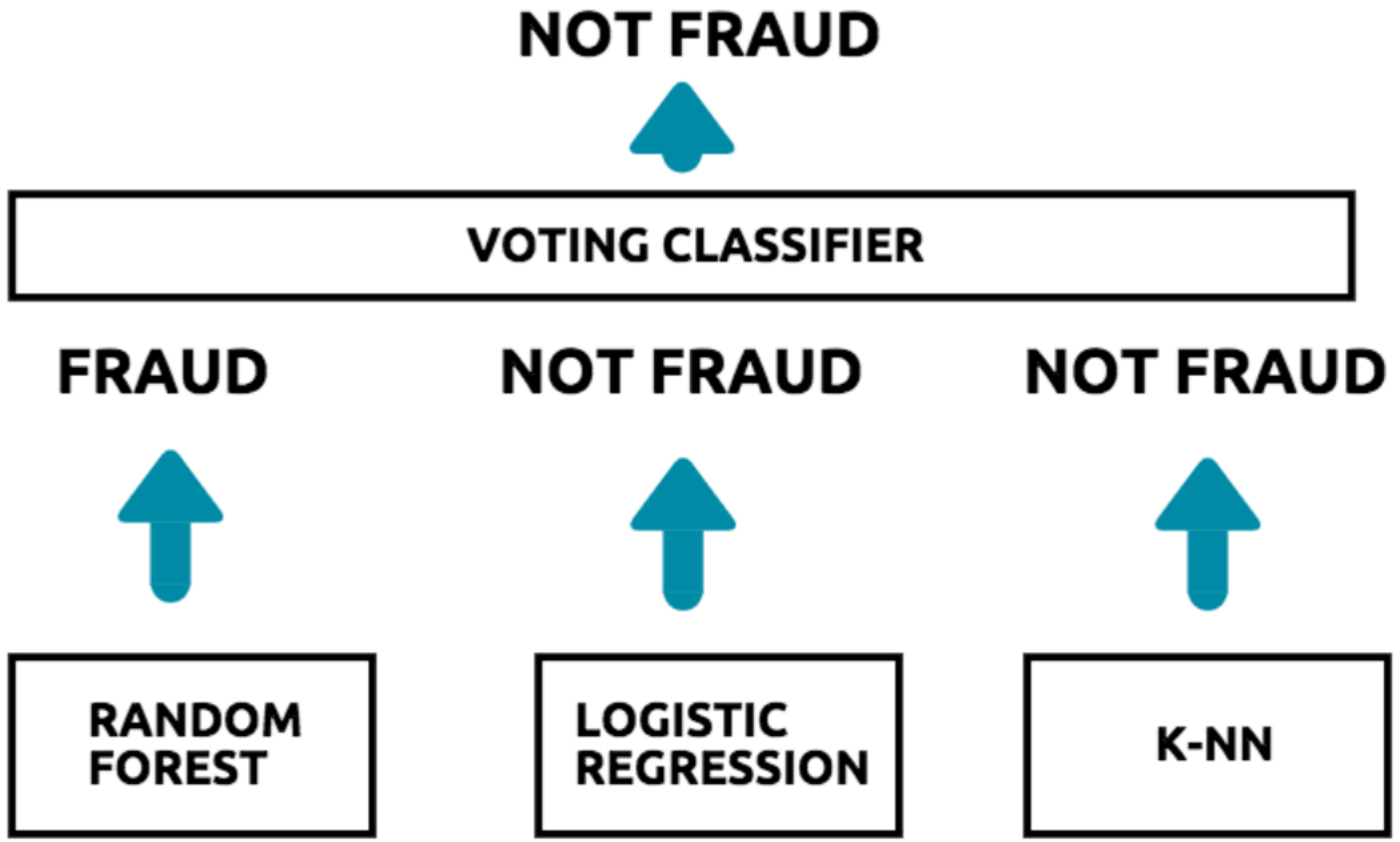
CS 522 – Selected Topics in CS

Ensemble Methods

Outline

- Key terms
- Ensemble Method
- Why use Ensemble Method
- When use Ensemble Method
- Types of Ensemble Method
- Pros and Cons of Ensemble Method

+ Ensemble


NOT FRAUD
VOTING CLASSIFIER
FRAUD
NOT FRAUD
NOT FRAUD
RANDOM FOREST
LOGISTIC REGRESSION
K-NN

Ensemble Learning

- ***Suppose you want a buy a laptop***
  - *A: You may ask **one** of your **friends** to **rate** the laptop for you.*
    - *B: Another way could be by asking **5 colleagues** of yours to rate the laptop.*
      - *C: How about asking **50 people** to rate the laptop?*
  - *The responses, in this case, would be **more generalized** and **diversified** since now you have people with **different sets of skills**.*
- *A **diverse group** of people are likely to **make better decisions** as compared to **individuals***
- *This **diversification** in Machine Learning is achieved by a technique called **Ensemble Learning.***

Motivation

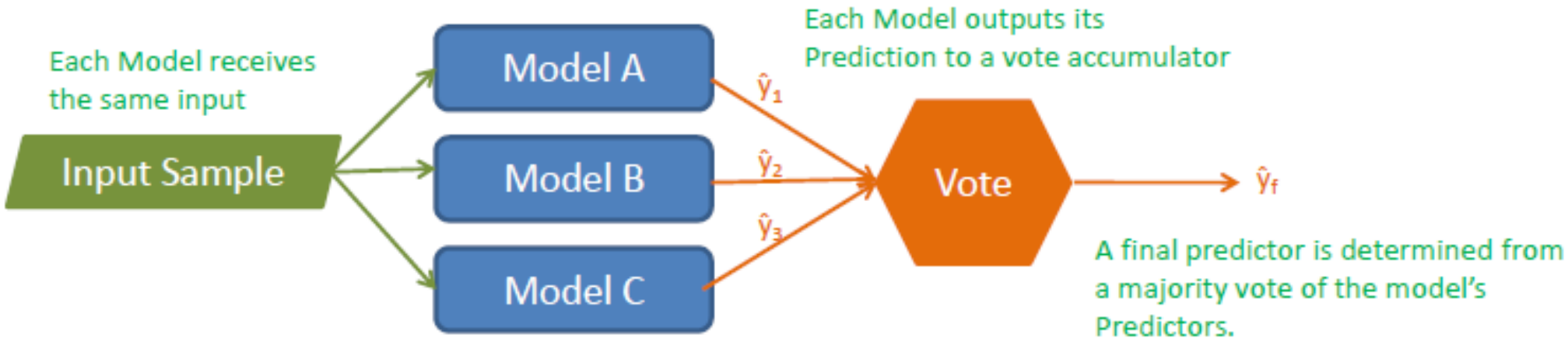
- Suppose that you are a ***patient*** with a ***set*** of ***symptoms***
- Instead of taking opinion of just ***one doctor*** (classifier), you decide to take opinion of a ***few doctors***!
- *Is this a good idea? Indeed it is.*
- Consult ***many doctors*** and then ***based*** on their ***diagnosis***; you can get an accurate idea of the diagnosis.

What is Ensemble Methods?

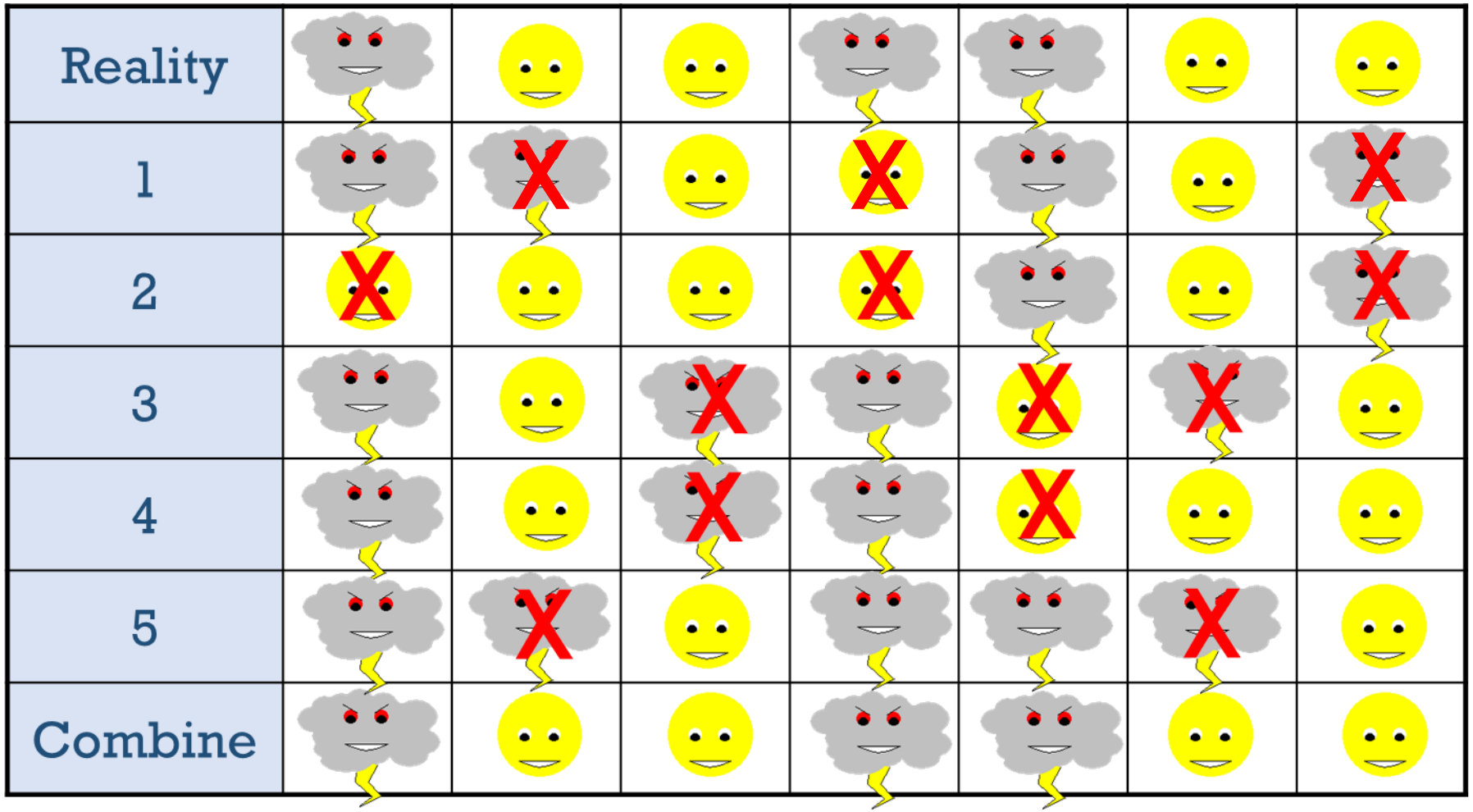
- An ***ensemble*** of ***classifiers is a set of classifiers*** whose ***individual decisions*** are ***combined*** in some way to classify ***new examples*** (Dietterich, 2000)
- ***Predicts class labels*** on previously ***unseen records*** by ***aggregating*** predictions made be ***multiple classifiers***
- The main ***principle*** behind ensemble modelling is to ***group weak learners together to form one strong learner***.
- Using Multiple Learning algorithms ***together*** for the ***same task***.
- ***Better predictions than individual learning***

Ensemble Method

- An ensemble method is a combination of multiple and diverse models.
- ***Each model*** in the ensemble makes a ***prediction***.
- A final prediction is determined by a ***majority vote*** among the ***models***.

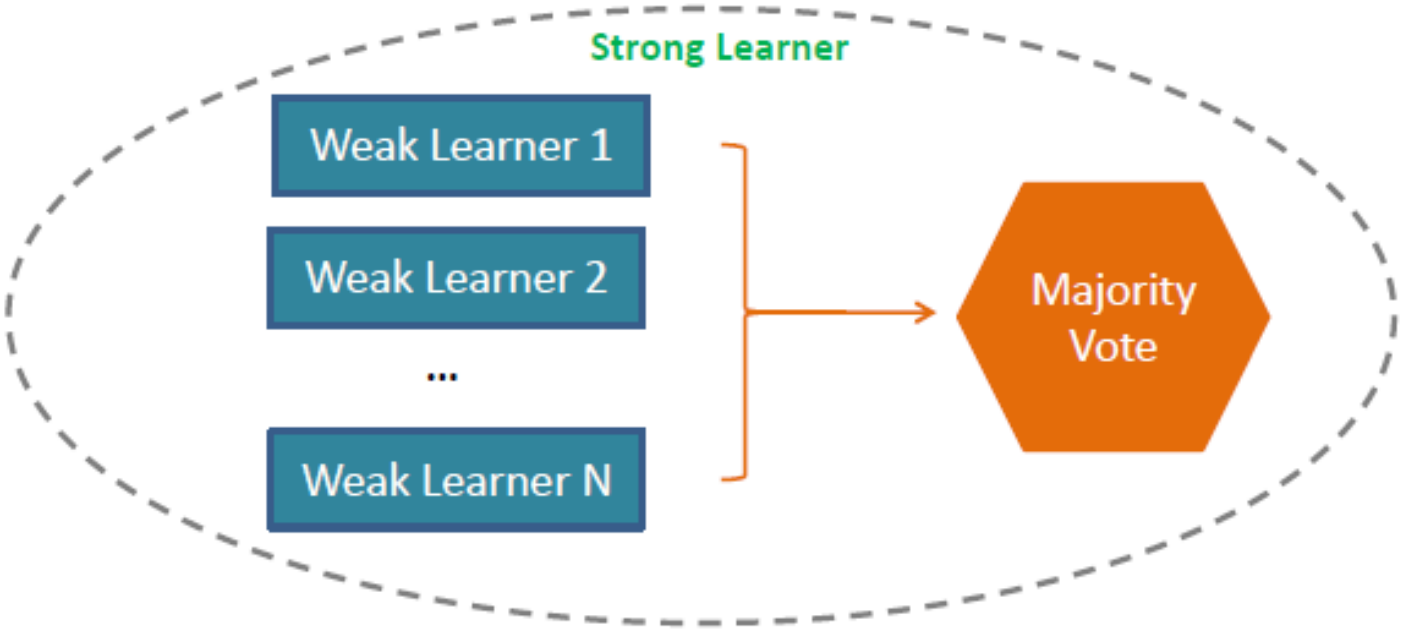


+ Example: Weather Forecast



Weak Learner

- In an Ensemble method, ***one combines multiple weak learners*** to make a ***strong learning model***.



Reasons to use ensemble

- ***The dataset is too large or small***
  - If dataset is ***too large*** or ***small***, we must use ***sampling*** to choose sample to take ***average*** of the result.
- ***Complex(Non-linear) data***
  - Real time dataset is mostly in ***non-linear fashion***.
  - So, when we train a ***single model*** which ***cannot*** define the ***class boundary*** clearly and model become ***under-fit***.
  - That case we must take ***different sub sample*** and take ***average*** of different model.

Reasons to use ensemble

- ***High Confidence:***
  - When we train a model with ***multiple classifier*** and get ***high correlated output*** these situation lead the ***High Confidence***.
  - So, In this case most of the model predict the same class which lead that high confidence.

Why use Ensemble Models

- Better Accuracy (***Low Error***)
- Higher Consistency (***Avoids Overfitting***)
- ***Reduce Bias*** and ***Variance Errors***

When and Where to use Ensemble Models

- Single Model ***Overfits***
- Results worth the ***extra training***
- Can be used for ***classification*** as well as ***regression***

Categorization of Ensemble technique

- Categorization based on classifier used
  - Homogenous Ensemble
  - Heterogenous Ensemble

- Differ in training strategy, and combination method
  - ***Parallel training*** with different training sets: ***bagging, Voting***
  - ***Sequential training***, ***iteratively reweighting*** training examples so current classifier focuses on hard examples: ***boosting***

- Also known as ***meta learning***

1. Homogenous Ensembles

- Use a ***same, arbitrary learning algorithm*** but manipulate different training data to make it learn multiple models.
  - ***Data1 ≠ Data2 ≠ … ≠ Data m***
  - ***Learner1 = Learner2 = … = Learner m***
- Different methods for changing training data:
  - **Bagging: Resample training data**
  - **Boosting: Reweight training data**
- In WEKA, these are called ***meta-learners***, they take a learning ***algorithm*** as an ***argument (base learner)*** and create a ***new learning*** algorithm.

2. Heterogeneous Ensembles

- Use ***different, arbitrary learning algorithms*** in a combined form but:
- Either keeping the data fixed to make the combination learn multiple models.
  - ***Learner1 ≠ Learner2 ≠ … ≠ Learner m***
  - ***Data1 = Data2 = … = Data m***
- Or manipulate different training data to make the combination learn multiple models.
  - ***Learner1 ≠ Learner2 ≠ … ≠ Learner m***
  - ***Data1 ≠ Data2 ≠ … ≠ Data m***

Steps in Ensemble Method

- There are two steps in Ensemble learning
- ***Step 01:*** ***Multiples machine learning models*** were generated using ***same*** or ***different*** machine ***learning algorithms***. These are called ***base models***.
- ***Step 02:*** The ***prediction*** perform based on the ***base model***.

Techniques of Ensemble Methods

1. Voting (improves prediction)
2. Bagging (Bootstrap Aggregating) (decrease variance)
3. Boosting (reduce the Bias)
4. Stacking

Variance : Variance measures how much the predictions for a given model change if the training data changes. High variance indicates that the model is very sensitive to the specific data it was trained on, leading to overfitting.

Bias: Bias measures how much the predictions of the model are off from the true values. High bias indicates that the model makes strong assumptions about the data, leading to underfitting.

Simple methods

1. *Max Voting*

Colleague 1	Colleague 2	Colleague 3	Colleague 4	Colleague 5	Final rating
5	4	5	4	4	4

2. *Averaging*

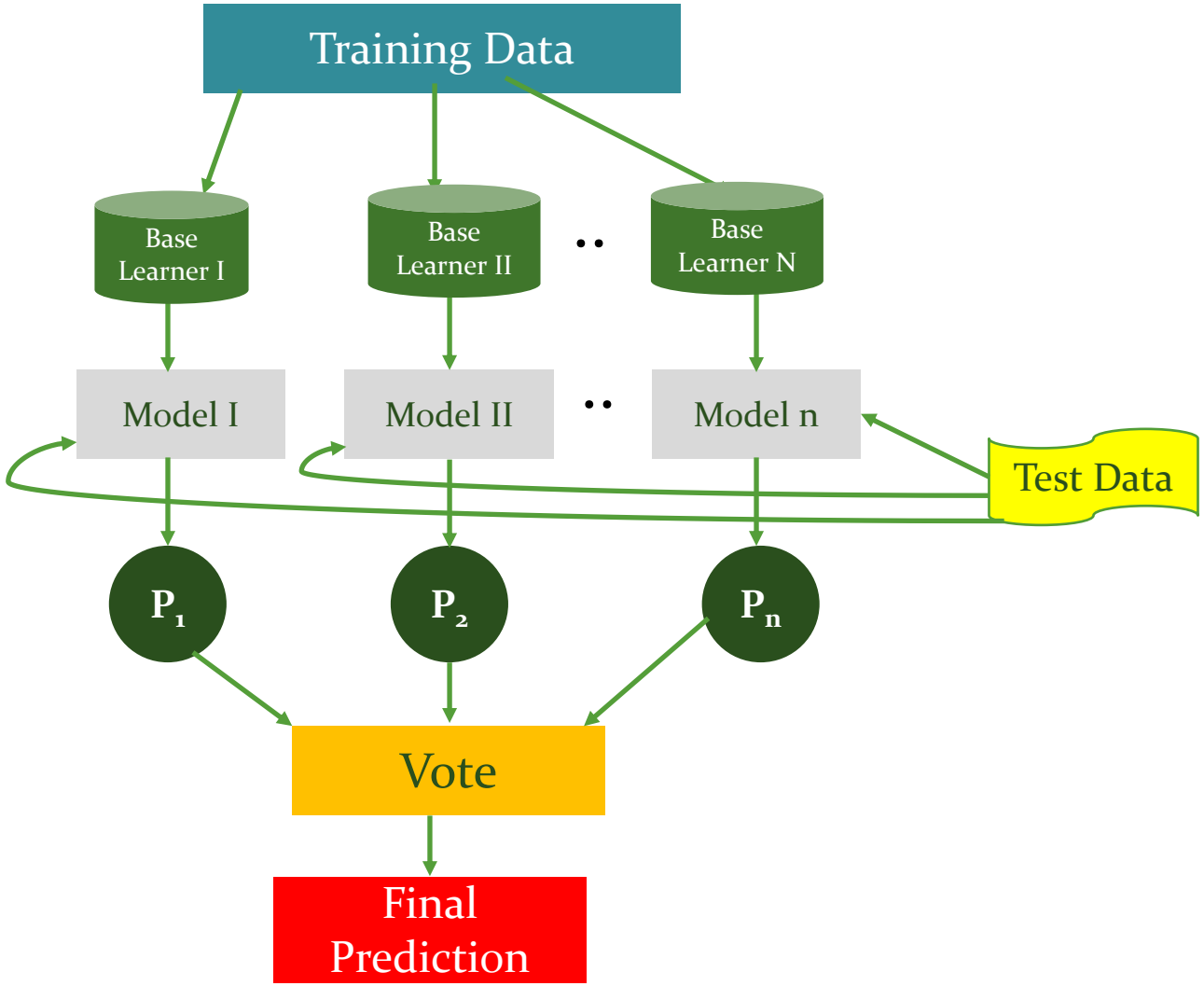
Colleague 1	Colleague 2	Colleague 3	Colleague 4	Colleague 5	Final rating
5	4	5	4	4	4.4

3. *Weighted Averaging*

Colleague	Colleague 1	Colleague 2	Colleague 3	Colleague 4	Colleague 5	Final rating	
weight	0.23	0.23	0.18	0.18	0.18	0.18	
rating	5	4	5	4	4	4	4.41

The result is calculated as [(5*0.23) + (4*0.23) + (5*0.18) + (4*0.18) + (4*0.18)] = 4.41.

1. Voting

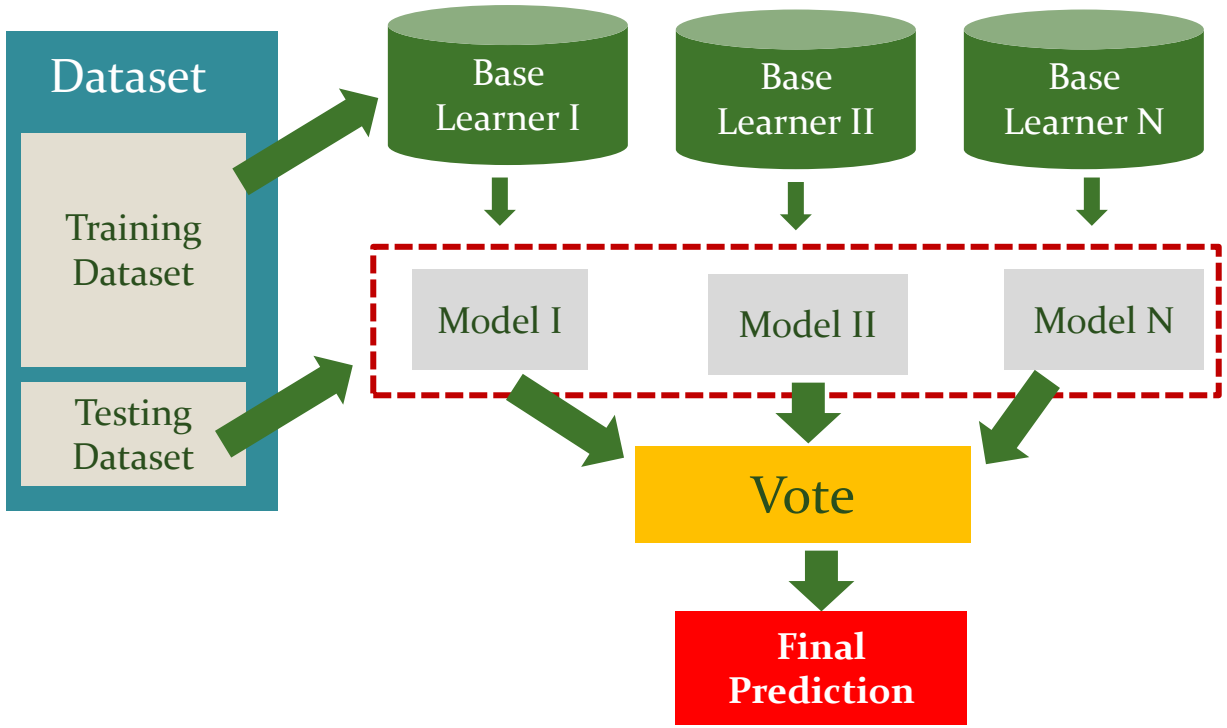


- All Training Data
- Different Classifier
- Parallel Execution

- Train several ***base learner*** on the ***training data***.
- ***Each*** learner will make the ***predictions*** on the test data.
- For ***classification***, take ***majority of vote*** on the prediction of base learners.
- For ***Regression***, take ***average*** of the prediction of base learners.

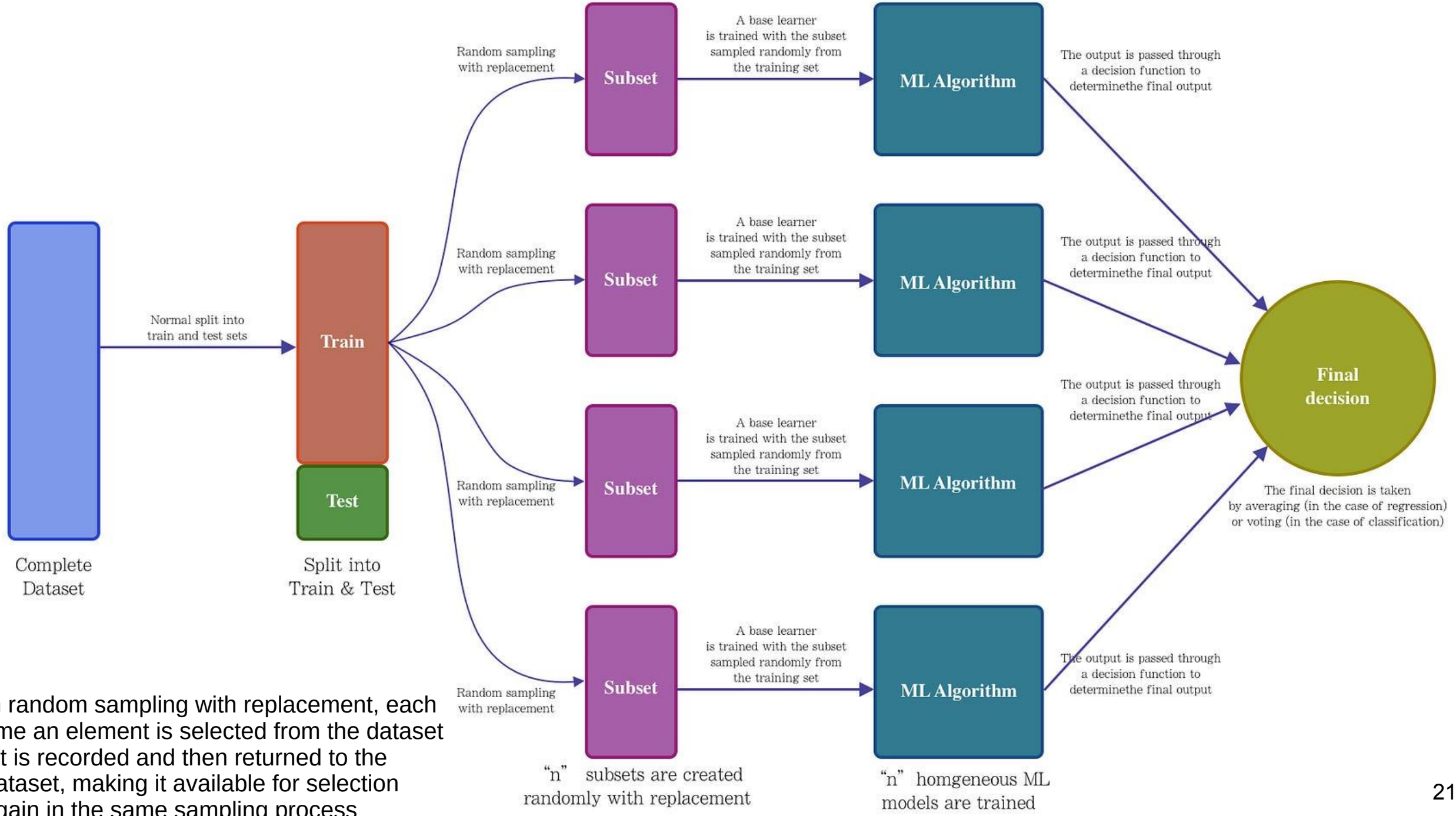
2. Bagging (Bootstrap Aggregation)

- Multiple models of the ***same learning algorithm*** is trained with the subset of dataset ***randomly picked*** from the ***training dataset***.
- It uses the ***simple random sampling with replacement also known as bootstrap sampling***.



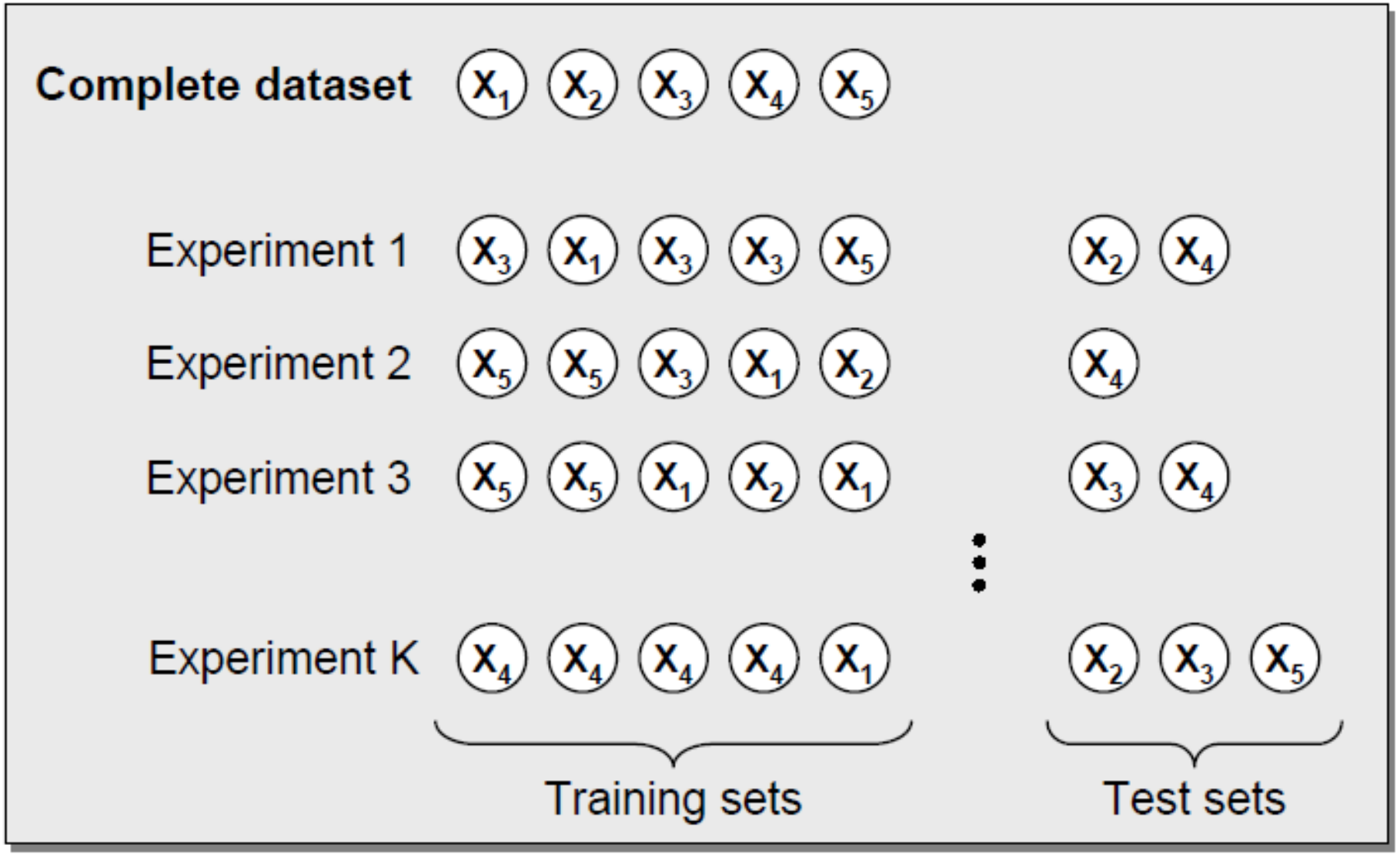
- Subset Training Data
- Same Classifier
- Parallel Execution

- Bagging is a method of ***deriving multiple models*** from the ***same training data***, where each model uses a ***subset*** of the training data ***selected*** at ***random***.
- A ***prediction*** is then made based on a ***majority vote*** of the models.



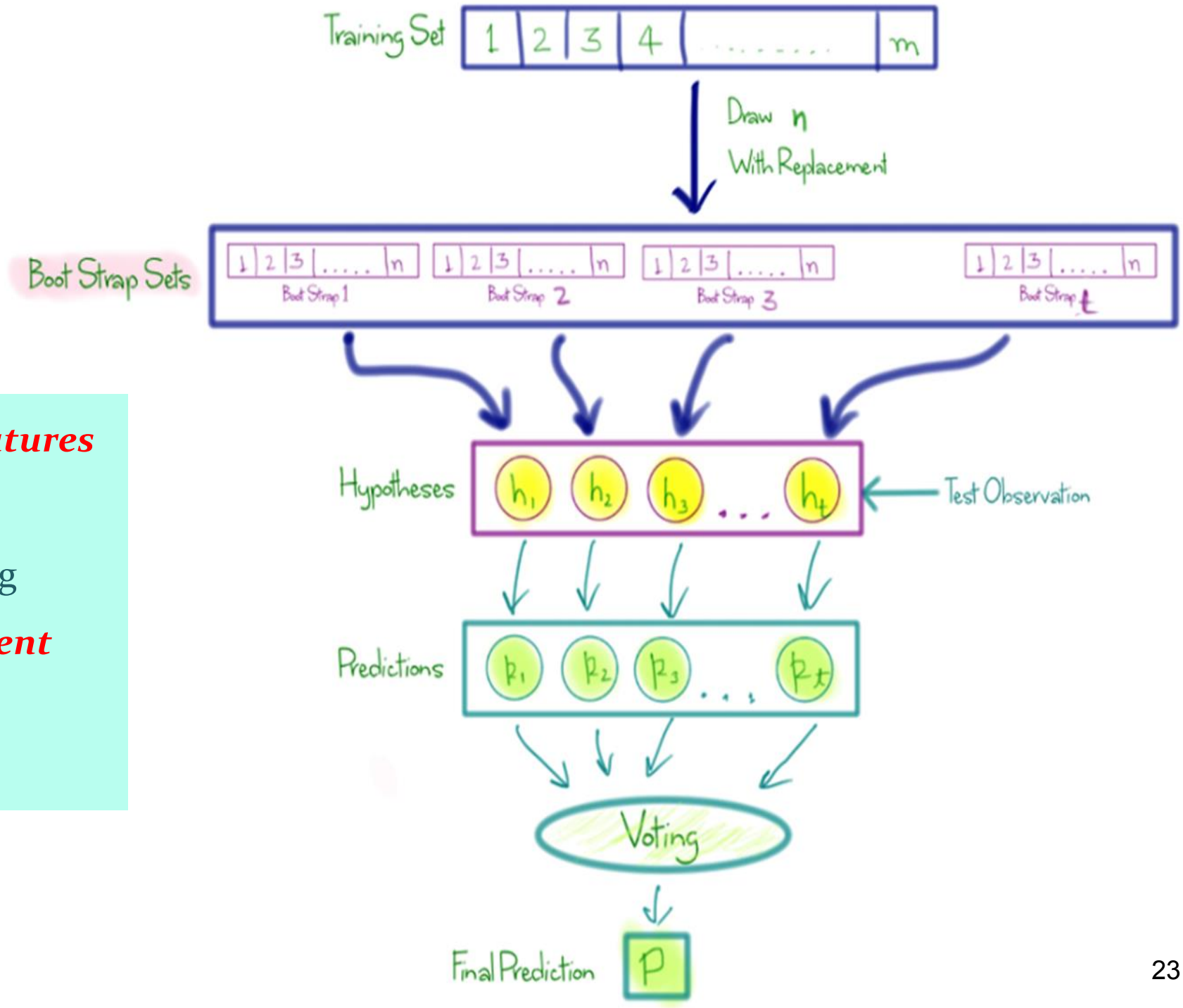
In random sampling with replacement, each time an element is selected from the dataset , it is recorded and then returned to the dataset, making it available for selection again in the same sampling process

+ Bootstrap sampling



2. Bagging

- ***Subset of features/ all features*** from Training Data.
- Subset of training data using ***Randomly with replacement***
- ***Same*** Classifier
- Parallel Execution



2. Bagging

- Bagging is used when the ***goal*** is to ***reduce*** the ***variance*** of a decision tree classifier.
- Here the objective is to ***create several subsets of data*** from ***training sample chosen randomly*** with ***replacement***.
- Each ***collection of subset data is used to train their decision trees***.
- As a result, we get an ensemble of different models.
- Average of all the predictions from different trees are used which is ***more robust*** than a ***single decision*** tree classifier.
- Prediction is given based on the aggregation of predictions from all the models

2. Steps in Bagging

1. The key to bagging is the art of creating ***bootstrap training sets***. Let's say there are ***m*** observations. Then create ***t bootstrap*** sets of ***size n*** each, by choosing ***n observations***(***with replacement***) from ***m observations***. If ***n*** tends to ***m***, then ***each set*** will have ***63% of original observations***, and rest will be ***duplicates***.

2. The ***training observations*** that are ***not chosen*** in a ***specific bootstrap*** set are referred to as ***out-of-bag (OOB)*** observations. Each base learner can ***report errors*** on the ***OOB observations***.

3. Create ***t*** models using ***each of these bootstrap training sets***.

4. Each of the ***base learners*** can make ***predictions***.
   - Like in ***Voting Ensemble***, for a regression-based prediction, ***average*** the predictions from base learners.
   - For classification, take ***majority vote*** on the predictions of base learners.

When Does Bagging Works?

- Learning algorithm is ***unstable***, if ***small changes*** to the training set cause ***large changes*** in the learned classifier
- If the learning algorithm is ***unstable***, then ***Bagging always improves*** performance
- ***Bagging stable classifiers is not a good idea***
- Which ones are ***unstable?***
  - ***Neural nets, decision trees, regression trees, linear regression***
- Which ones are ***stable***?
  - ***K-nearest neighbors***

+ 2. Bagging (Review)

Bagging	
Partitioning of data	Random with Replacement
Goal to achieve	Minimum variance
Methods used	Random subspace
Functions to combine single model	Average, majority, Weighted average
Example	Random Forest, Bagged Decision Trees, Extra Trees

+ Bagging

Pros

- ***Reduces over-fitting*** of the model.
- Reduces variance
- Handles ***higher dimensionality*** data very ***well***.
- Maintains ***accuracy*** for ***missing data***.

Cons

- Since final prediction is based on the ***mean predictions*** from ***subset trees***, it ***won't*** give ***precise*** values for the regression model.

3. Boosting

- Analogy: Consult ***several doctors***, based on a ***combination of weighted diagnoses***—***weight assigned*** based on the previous diagnosis accuracy
- The boosting algorithm can be ***extended for the prediction of continuous values.***
- Comparing with bagging: ***boosting*** tends to ***achieve greater accuracy***, but it also ***risks overfitting*** the model to ***misclassified data.***
- Though it is quite ***successful***, the ***disadvantage*** of the ***original boosting*** method is that it ***requires*** a ***very large*** training ***sample***.

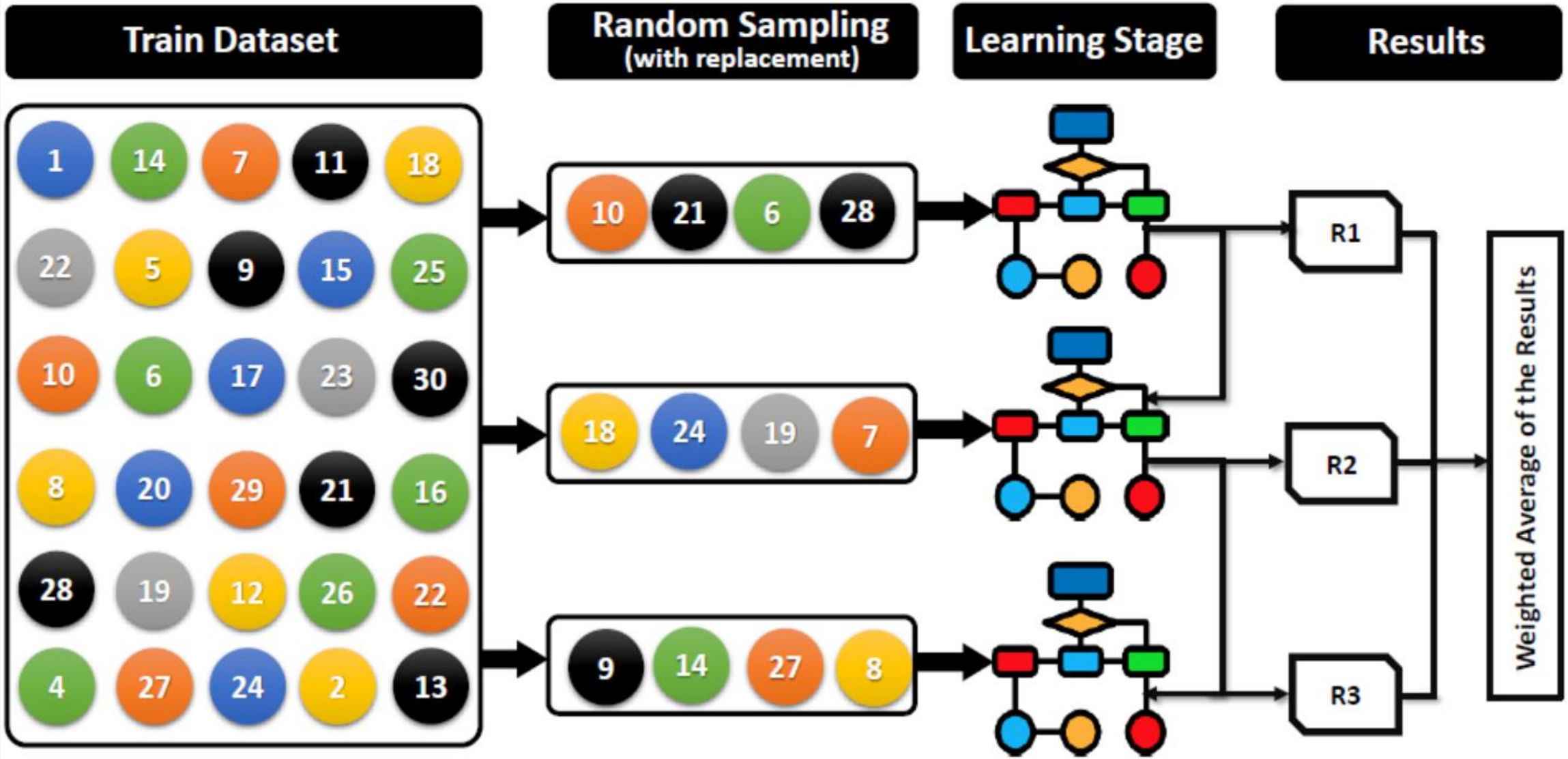
3. Boosting

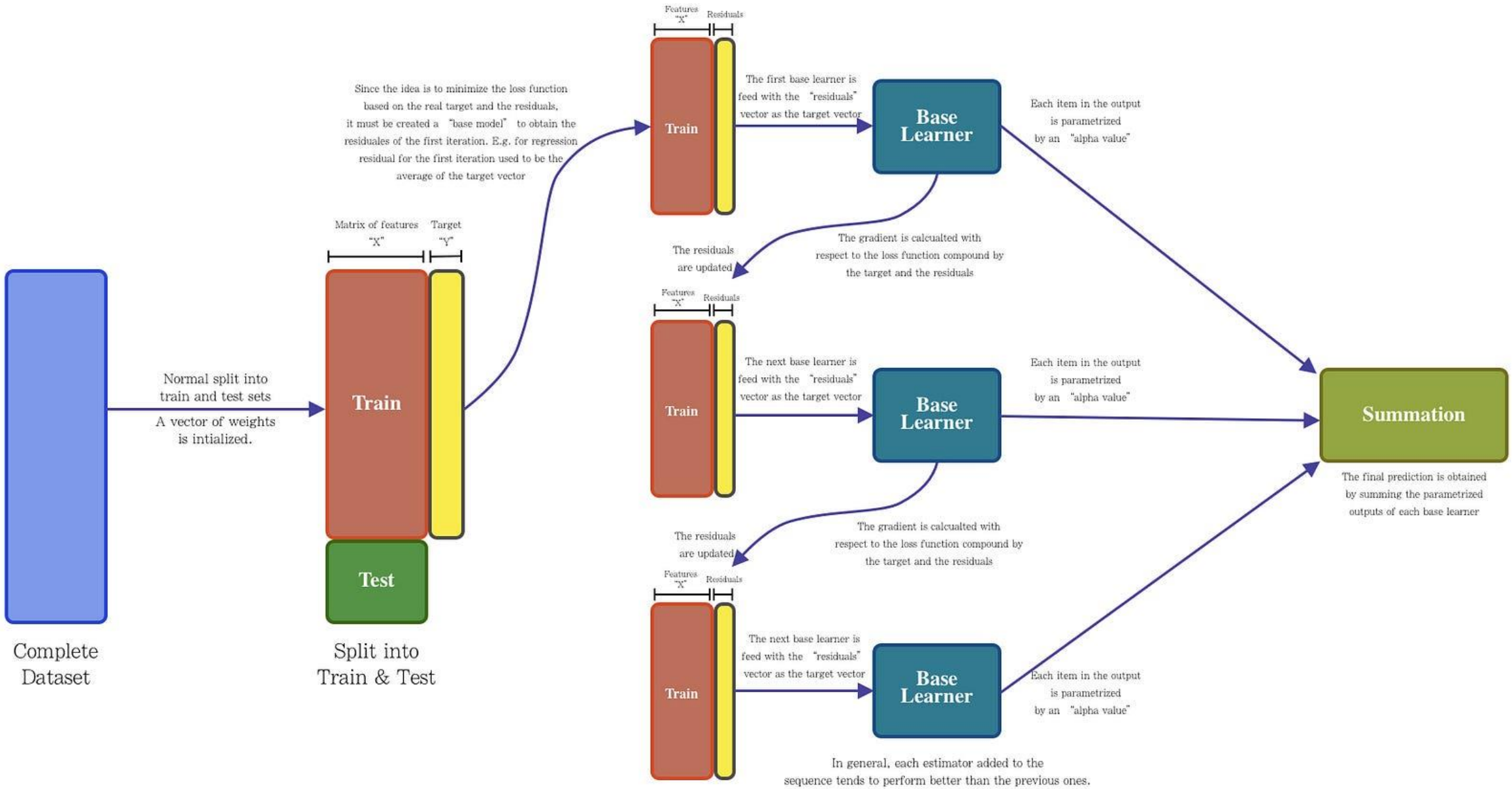
- Boosting is used to create ***a collection of predictors***.
- In this technique, learners are learned ***sequentially*** with ***early learners*** fitting ***simple models*** to the data and then ***analyzing*** data for errors.
- ***Consecutive trees*** (random sample) are ***fit*** and at ***every step***, the goal is to ***improve*** the ***accuracy*** from the prior tree.
- When an ***input*** is ***misclassified*** by a ***hypothesis***, its ***weight*** is ***increased*** so that ***next hypothesis*** is more likely to ***classify*** it ***correctly***.
- ***This process converts weak learners into better performing model.***

Example (Boosting)

- Suppose there are just ***5 training examples {1,2,3,4,5}.***
- Initially each example has a ***0.2 (1/5)*** probability of being sampled.
- 1st round of boosting samples (with replacement) ***5 examples: {2, 4, 4, 3, 2}*** and ***builds*** a ***classifier*** from them.
- Suppose examples ***2, 3, 5*** are ***correctly*** predicted by this classifier, and examples ***1, 4 are wrongly predicted***:
  - Weight of examples ***1 and 4*** is ***increased***,
  - Weight of examples ***2, 3, 5 is decreased.***
- 2nd round of boosting samples again ***5 examples***, but now ***examples 1 and 4*** are ***more likely*** to be ***sampled.***
- And so on …until some ***convergence is achieved.***

+ Boosting

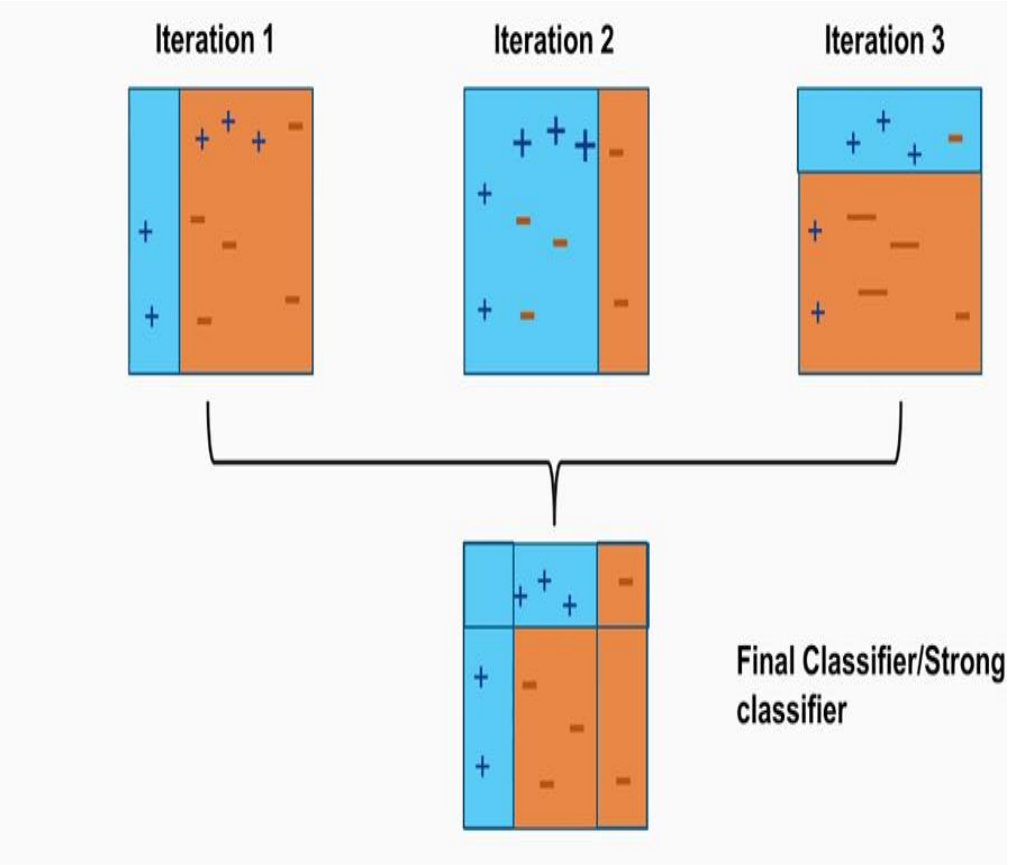

Train Dataset
Random Sampling
(with replacement)
Learning Stage
Results
1 14 7 11 18
22 5 9 15 25
10 6 17 23 30
8 20 29 21 16
28 19 12 26 22
4 27 24 2 13
10 21 6 28
18 24 19 7
9 14 27 8
R1
R2
R3
Weighted Average of the Results

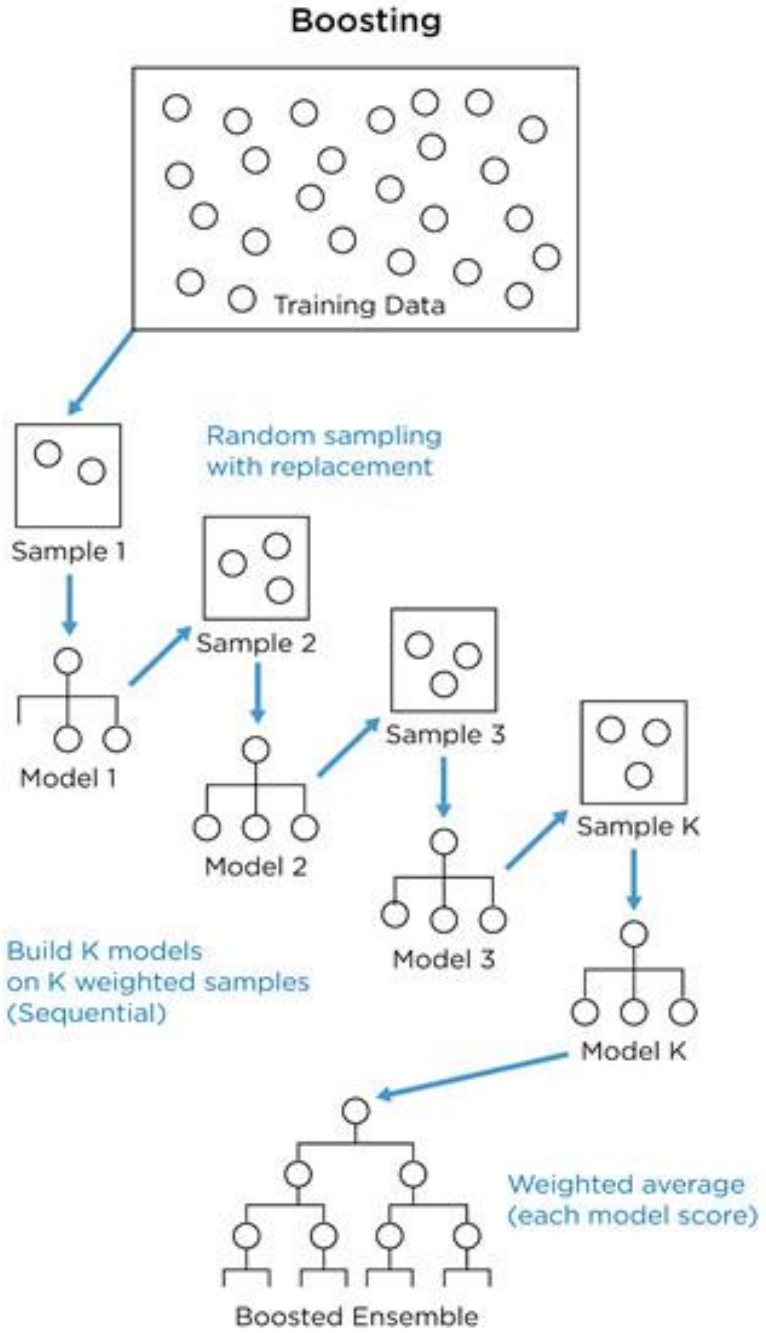

Complete Dataset
Normal split into train and test sets
A vector of weights is intialized.
Matrix of features "X"
Target "Y"
Train
Test
Split into Train & Test
Since the idea is to minimize the loss function based on the real target and the residuals, it must be created a "base model" to obtain the residuales of the first iteration. E.g. for regression residual for the first iteration used to be the average of the target vector
Features "X"
Residuals
Train
The first base learner is feed with the "residuals" vector as the target vector
Base Learner
Each item in the output is parametrized by an "alpha value"
The residuals are updated
The gradient is calcualted with respect to the loss function compound by the target and the residuals
The next base learner is feed with the "residuals" vector as the target vector
In general, each estimator added to the sequence tends to perform better than the previous ones.
Summation
The final prediction is obtained by summing the parametrized outputs of each base learner

+ Steps in Boosting

1. ***Weights*** are ***assigned*** to each ***training tuple***.
2. A series of ***k*** classifiers is ***iteratively learned***.
3. After a classifier M_i is learned, the ***weights*** are ***updated*** to allow the ***subsequent classifier***, M_{i+1}, to pay ***more attention*** to the training tuples that were misclassified by M_i
4. The final ***M**** combines the ***votes*** of each ***individual classifier***, where the weight of each ***classifier's vote*** is a ***function*** of its ***accuracy***.

+ Example


Iteration 1
Iteration 2
Iteration 3
Final Classifier/Strong classifier


Boosting
Training Data
Random sampling with replacement
Sample 1
Sample 2
Sample 3
Sample K
Model 1
Model 2
Model 3
Model K
Build K models on K weighted samples (Sequential)
Weighted average (each model score)
Boosted Ensemble

+ 3. Boosting

Boosting	
Partitioning of data	Higher vote to misclassified samples
Goal to achieve	Increase accuracy, reduce Bias
Methods used	Gradient descent *(for optimization)*
Functions to combine single model	Weighted majority vote
Example	Ada Boost, Stochastic Gradient Boosting, GBM, XGBM, Light GBM, CatBoost

XGBoost

XGBoost started as a research project by Tianqi Chen as part of the Distributed machine learning community group.

XGBoost is a scalable and flexible boosting algorithms optimised for preforming data science tasks.

XGBoost supports distributed computing and out of core processing for machine learning training purposes.

XGBoost has the capacity for handling missing data and regularization

XGBoost based models have won many machine learning challenges.

+ CatBoost

Categorical + Boosting (CatBoost) is a boosting algorithm Developed by Yandex researchers and engineers.

CatBoost produces good results even without extensive hyper-parameter tuning.

CatBoost has the inbuilt capacity to handle categorical features even the non-numerical ones.

Catboost offers improved performance as it reduces overfitting when building model.

Catboost is fast and scalable and provides GPU support.

Ada Boost

AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm formulated by Yoav, Freund and Robert Schapire.

AdaBoost can be less susceptible to the overfitting problem than other learning algorithms.

AdaBoost is sensitive to noisy data and outliers.

AdaBoost was designed to put more weight on misclassified samples and less weight on correctly classified samples. The final prediction is a weighted average of all the weak learners, where more weight is placed on stronger learners. .

LightGBM

LightGBM was developed by Microsoft machine learning group.

LightGBM has the capability for distributed computing and has GPU support.

LightGBM is built for model training speed and churning high performing model.

LightGBM is capable on of big data with its in-built parallel computing capacity.

LightGBM is optimised for low memory usage

+ Boosting

Pros

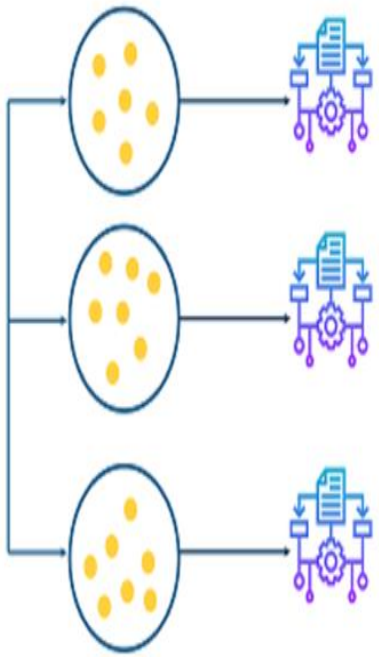
- ***Reduce Bias***.
- Very ***simple*** to ***implement***
- Fairly ***good generalization***
- ***Increase accuracy***
- ***Prior error need*** to be ***known*** ahead of time

Cons

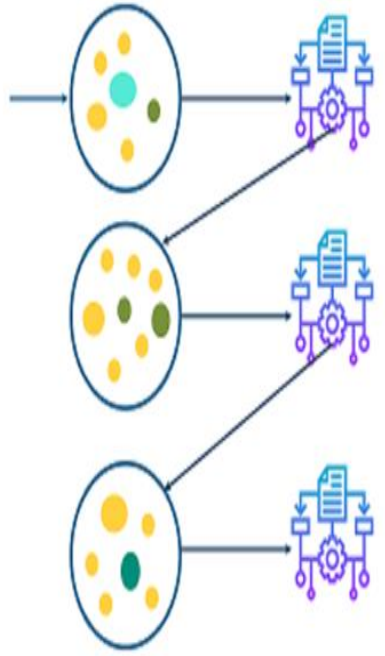
- Can ***over fit*** in presence of ***noise***
- Requires careful ***tuning*** of different ***hyper-parameters***.

+ Bagging Vs. Boosting

Bagging	*Boosting*
Every item has the same probability to appear in a new dataset	The samples are weighted so some of them will occur more often
Parallel training stage	Build learners in sequential way
Final decision is the average of the N learners	Final decision is weighted average of the N learners Better classifiers have higher weights
Reduces variance and solves overfitting	Reduce Bias but increases overfitting a bit



Bagging - Parallel



Boosting - Sequential

Bagging and Boosting uses N learners and yields more stable models

Pros (Ensemble)

- Ensemble is a ***proven method*** for ***improving*** the ***accuracy*** of the model and works in most of the cases
- Ensemble makes the model ***more robust*** and ***stable*** thus ensuring ***decent performance*** on the test cases in ***most scenarios***
- You can use ensemble to capture ***linear and simple*** as well ***non-linear complex relationships*** in the data. This can be done by using two ***different models*** and ***forming*** an ensemble of two.

Cons (Ensemble)

- Ensemble ***reduces*** the model ***interpret-ability*** and makes it ***very difficult*** to ***draw*** any ***crucial business*** insights at the end. صعب تخيله ورسمه
- It is ***time-consuming*** and thus might not be the best idea for ***real-time applications***
- The ***selection*** of ***models*** for creating an ensemble is ***an art which is really hard to master.***

Conclusion

- Bagging to ***decrease*** the model's ***variance***;
- Boosting to ***decreasing*** the model's ***bias;***

- Using techniques like Bagging and Boosting ***helps to decrease*** the ***variance, bias*** and ***increased*** the ***robustness*** of the ***model***.
- Combinations of multiple classifiers ***decrease variance***, especially in the case of ***unstable classifiers***, and may produce a ***more reliable*** classification than a ***single classifier***.

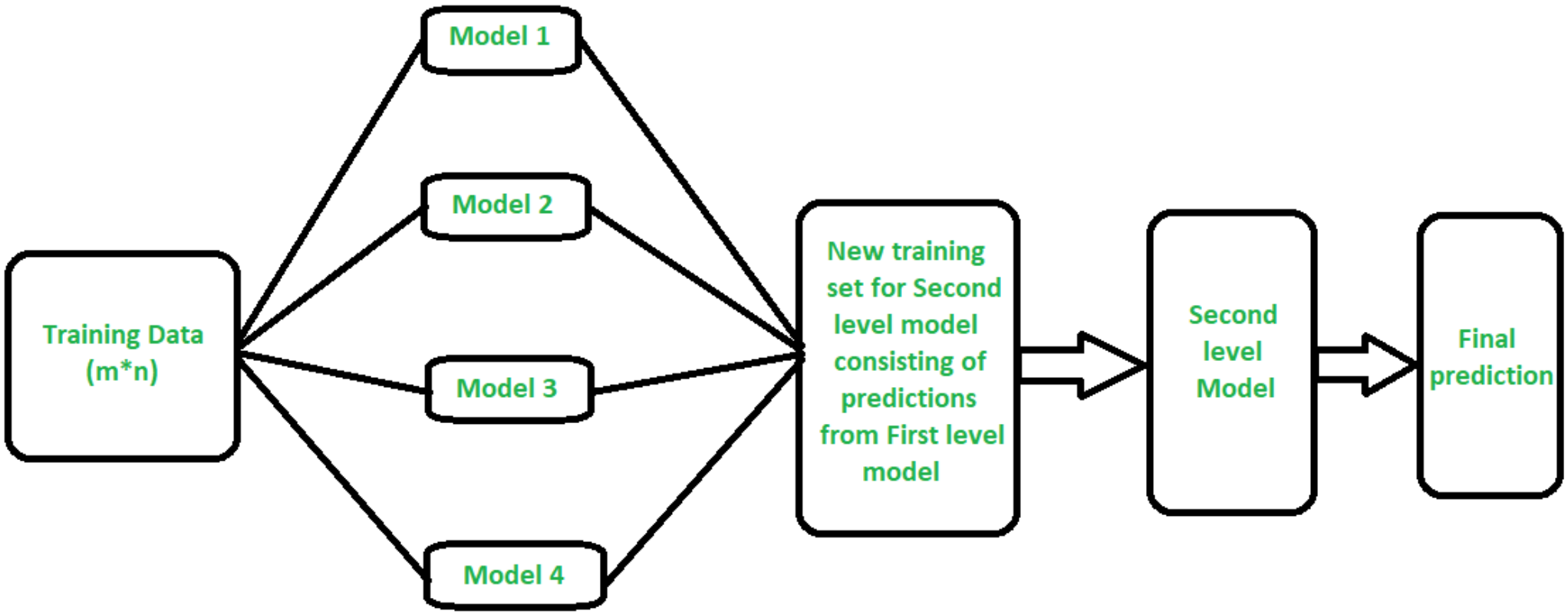
- The ***expected performance*** of the ensemble is guaranteed to be ***no worse*** than the ***average*** of the ***individual classifiers***

يضمن انه ليس اسوا من النماذج الفردية

+ Review

Ensemble Method	Classifier	Sampling	Training Strategy
Voting	Different types	Complete Training Set	Parallel
Bagging	Same type	Random Sample with Replacement Bootstrap Sampling	Parallel
Boosting	Typically, of same type	Random Sample with Replacement, weighting	Sequential

Stacking


Training Data
(m*n)
Model 1
Model 2
Model 3
Model 4
New training set for Second level model consisting of predictions from First level model
Second level Model
Final prediction

How stacking works?

1. We split the training data into K-folds just like K-fold cross-validation.
2. A base model is fitted on the K-1 parts and predictions are made for Kth part.
3. We do for each part of the training data.
4. The base model is then fitted on the whole train data set to calculate its performance on the test set.
5. We repeat the last 3 steps for other base models.
6. Predictions from the train set are used as features for the second level model.
7. Second level model is used to make a prediction on the test set.

End of Lecture - 08