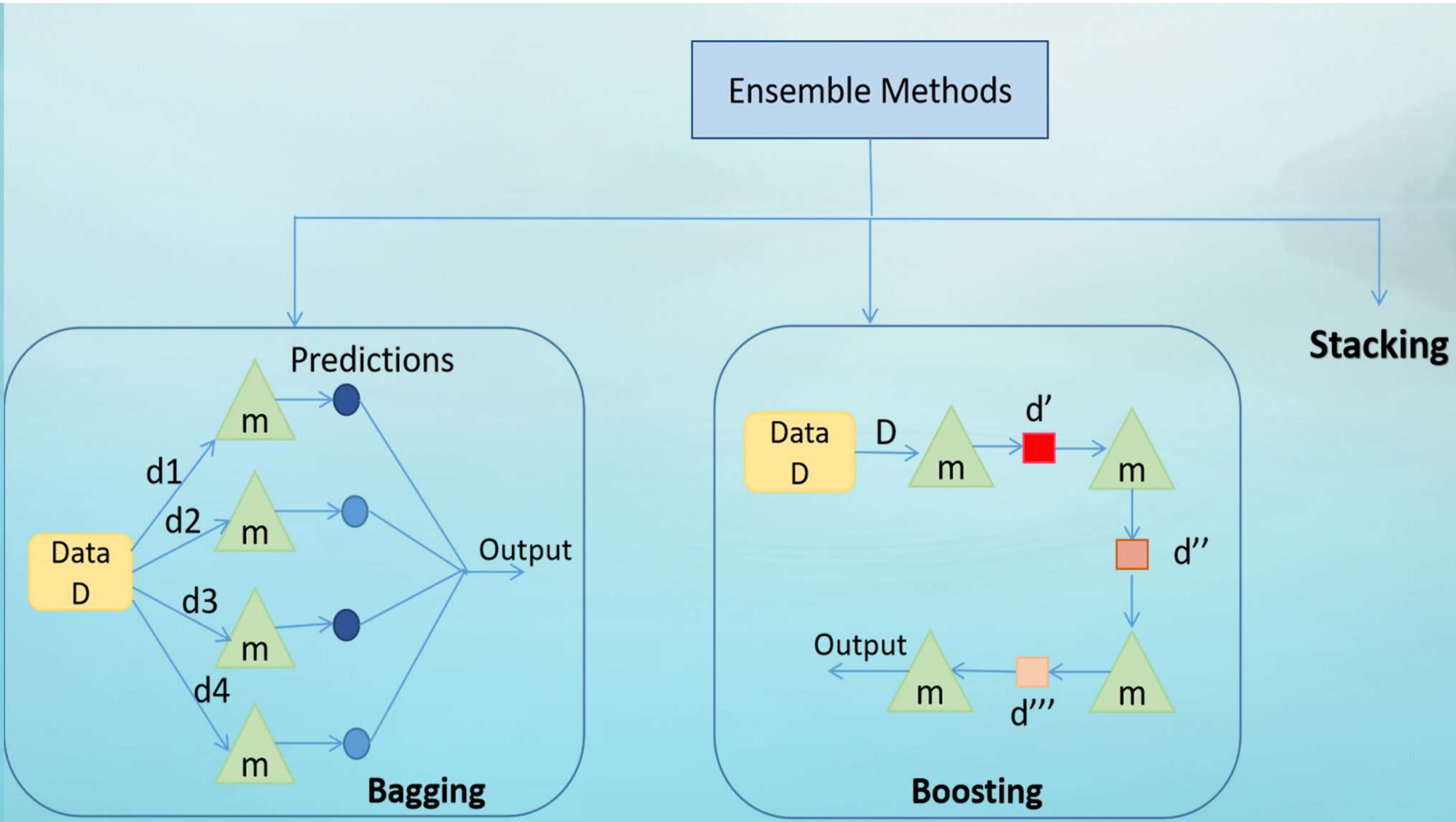
Ensemble Learning

Bagging, Boosting, Stacking

Ensemble techniques

- **Ensemble techniques** are the methods that use multiple learning algorithms or models to produce one optimal predictive model.
- The model produced has better performance than the base learners taken alone.
- Other applications of ensemble learning also include selecting the important features, data fusion, etc.
- Ensemble techniques can be primarily classified into Bagging, Boosting, and Stacking.



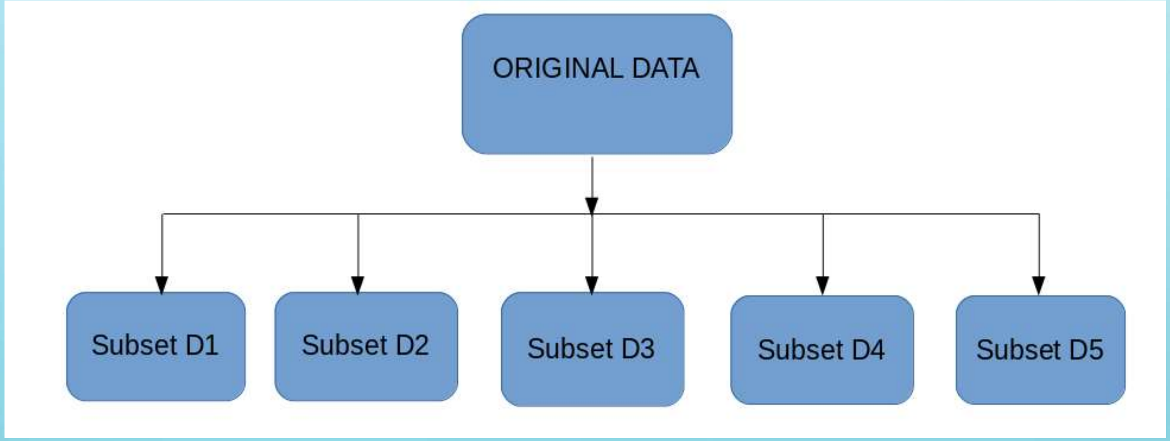
Here, m represents a weak learner; d1, d2, d3, d4 are the random samples from Data D; d’, d”, d”’ are updated training data based on the results from the previous weak learner.

Bagging

- The idea behind bagging is combining the results of multiple models (for instance, all decision trees) to get a generalized result.
- Here's a question: If you create all the models on the same set of data and combine it, will it be useful? There is a high chance that these models will give the same result since they are getting the same input.
- So how can we solve this problem? One of the techniques is bootstrapping.
- Bootstrapping is a sampling technique in which we create subsets of observations from the original dataset, with replacement.
- The size of the subsets is the same as the size of the original set.

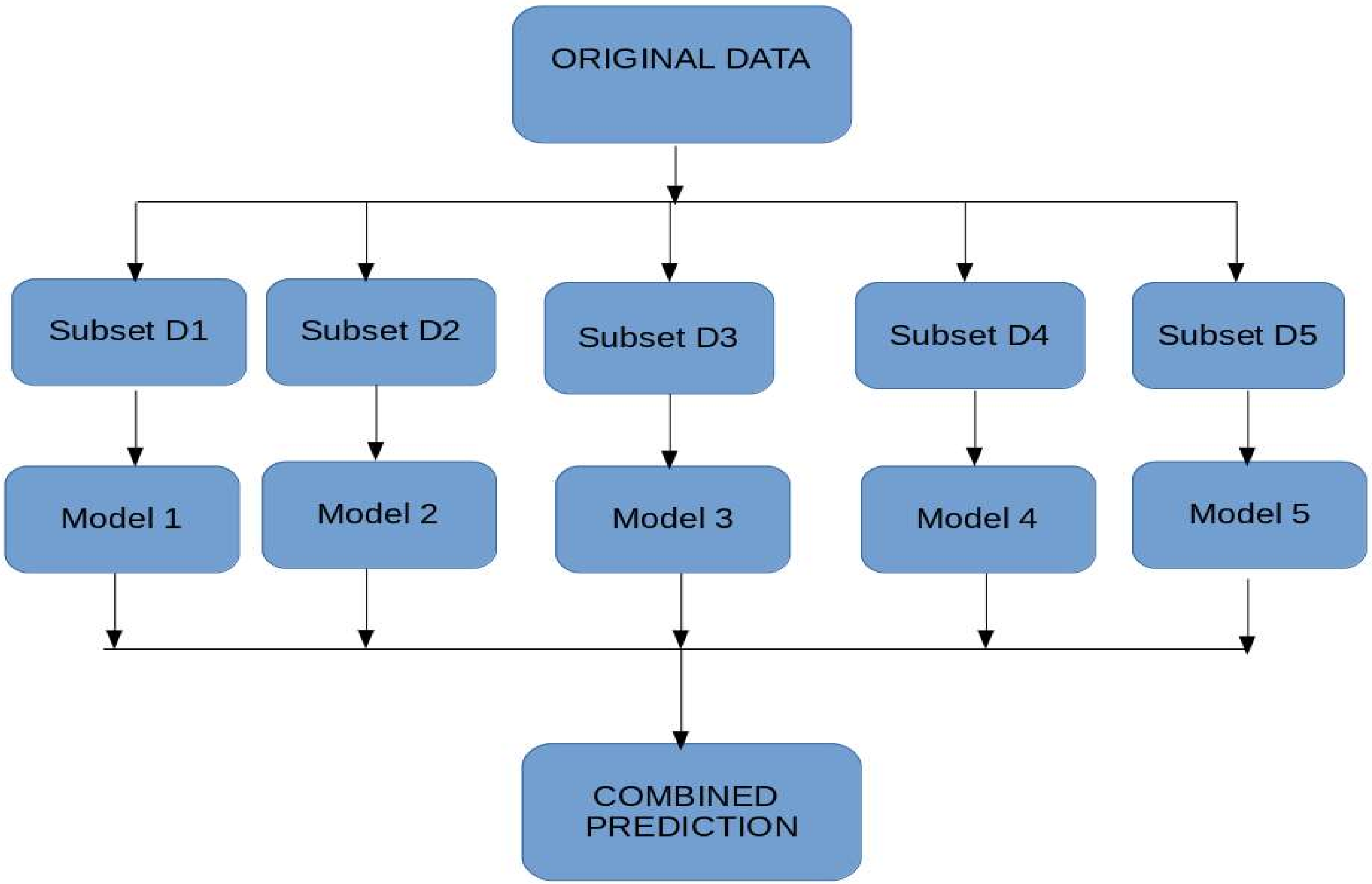
Bagging

- Bagging (or Bootstrap Aggregating) technique uses these subsets (bags) to get a fair idea of the distribution (complete set).
- The size of subsets created for bagging may be less than the original set.



Bagging

1. Multiple subsets are created from the original dataset, selecting observations with replacement.
2. A base model (weak model) is created on each of these subsets.
3. The models run in parallel and are independent of each other.
4. The final predictions are determined by combining the predictions from all the models.



ORIGINAL DATA

Subset D1

Subset D2

Subset D3

Subset D4

Subset D5

Model 1

Model 2

Model 3

Model 4

Model 5

COMBINED PREDICTION

Boosting

- Before we go further, here’s another question for you:
- If a data point is incorrectly predicted by the first model, and then the next (probably all models), will combining the predictions provide better results?
- Such situations are taken care of by boosting.
- Boosting is a sequential process, where each subsequent model attempts to correct the errors of the previous model.
- The succeeding models are dependent on the previous model.

- The term ‘Boosting’ refers to a family of algorithms which converts weak learner to strong learners.
- Consider the example of spam email identification
- How would you classify an email as SPAM or not? Like everyone else, our initial approach would be to identify ‘spam’ and ‘not spam’ emails using following criteria. If:

1. Email has only one image file (promotional image), It’s a SPAM
2. Email has only link(s), It’s a SPAM
3. Email body consist of sentence like “You won a prize money of $ xxxxxx”, It’s a SPAM
4. Email from our official domain “www.icici.com” , Not a SPAM
5. Email from known source, Not a SPAM

- Above, we've defined multiple rules to classify an email into 'spam' or 'not spam'. But, do you think these rules individually are strong enough to successfully classify an email? No.
- Individually, these rules are not powerful enough to classify an email into 'spam' or 'not spam'. Therefore, these rules are called as weak learner.
- To convert weak learner to strong learner, we'll combine the prediction of each weak learner using methods like:
  - Using average/ weighted average
  - Considering prediction has higher vote
- For example:
  - Above, we have defined 5 weak learners.
  - Out of these 5, 3 are voted as 'SPAM' and 2 are voted as 'Not a SPAM'.
  - In this case, by default, we'll consider an email as SPAM because we have higher(3) vote for 'SPAM'.

How Boosting Algorithms works?

- How boosting identify weak rules?
  - To find weak rule, we apply base learning (ML) algorithms with a different distribution.
  - Each time base learning algorithm is applied, it generates a new weak prediction rule.
  - This is an iterative process.
  - After many iterations, the boosting algorithm combines these weak rules into a single strong prediction rule.
- Here's another question which might haunt you, 'How do we choose different distribution for each round?'
- For choosing the right distribution, here are the steps as follows:

Boosting

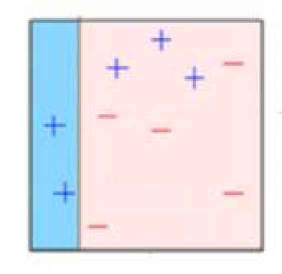
- Let’s understand the way boosting works in detail with the following steps :

1. A subset is created from the original dataset.

2. Initially, all data points are given equal weights.

3. A base model is created on this subset.

4. This model is used to make predictions on the whole dataset.

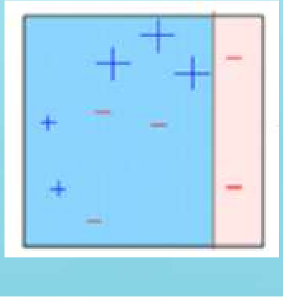


Boosting

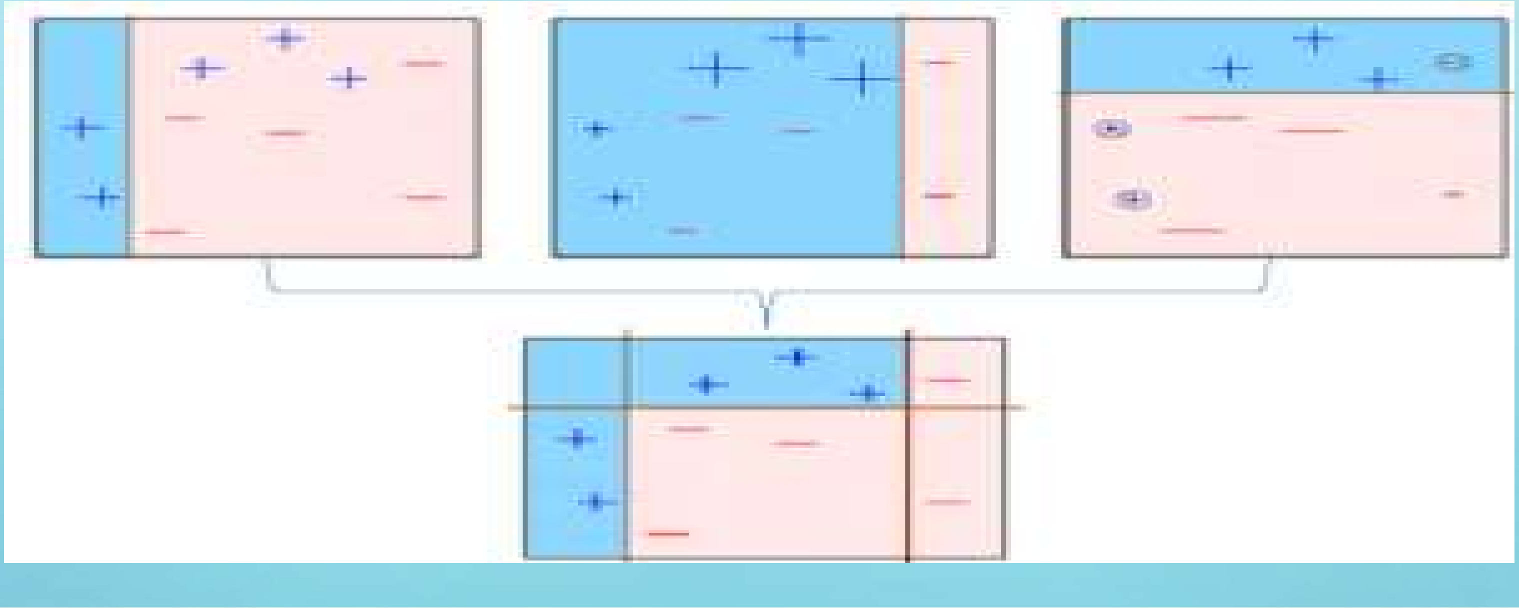
5. Errors are calculated using the actual values and predicted values.

6. The observations which are incorrectly predicted, are given higher weights.
   (Here, the three misclassified blue-plus points will be given higher weights)

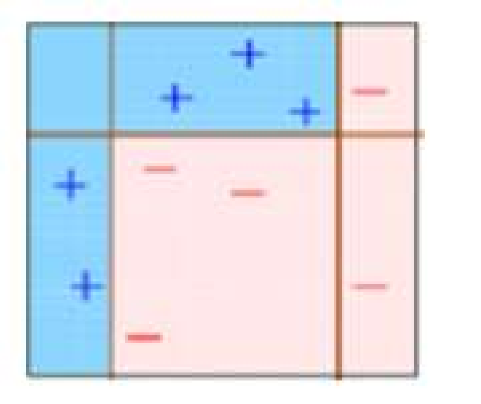
7. Another model is created and predictions are made on the dataset.
   (This model tries to correct the errors from the previous model)



8. Similarly, multiple models are created, each correcting the errors of the previous model.
9. The final model (strong learner) is the weighted mean of all the models (weak learners).



- Thus, the boosting algorithm combines a number of weak learners to form a strong learner.
- The individual models would not perform well on the entire dataset, but they work well for some part of the dataset.
- Thus, each model actually boosts the performance of the ensemble.



Types of Boosting Algorithms

- Underlying engine used for boosting algorithms can be anything. It can be decision stamp, margin-maximizing classification algorithm etc. There are many boosting algorithms which use other types of engine such as:

1. AdaBoost (Adaptive Boosting)
2. Gradient Tree Boosting
3. XGBoost

AdaBoost (Adaptive Boosting)

- Boosting is a fairly simple variation on bagging that strives to improve the learners by focusing on areas where the system is not performing well.
- One of the most well-known algorithms in this area is called Ada boost.
- It fits a sequence of weak learners on different weighted training data.
- It starts by predicting original data set and gives equal weight to each observation.

AdaBoost (Adaptive Boosting)

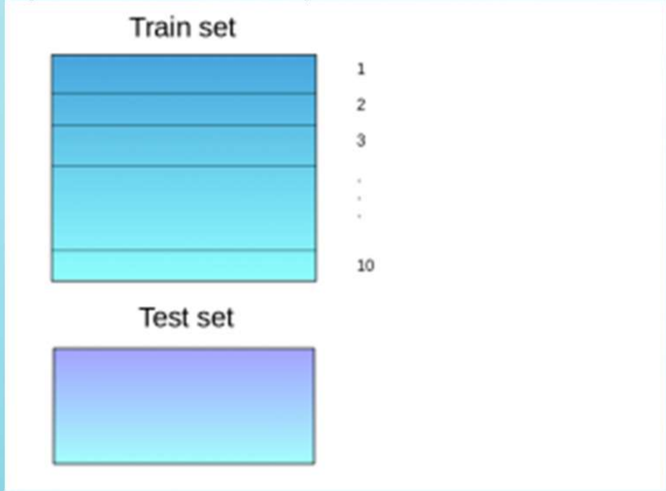
- If prediction is incorrect using the first learner, then it gives higher weight to observation which have been predicted incorrectly.
- Being an iterative process, it continues to add learner(s) until a limit is reached in the number of models or accuracy.
- Mostly, we use decision stamps with AdaBoost.
- But, we can use any machine learning algorithms as base learner if it accepts weight on training data set.
- We can use AdaBoost algorithms for both classification and regression problem.

Gradient Boosting

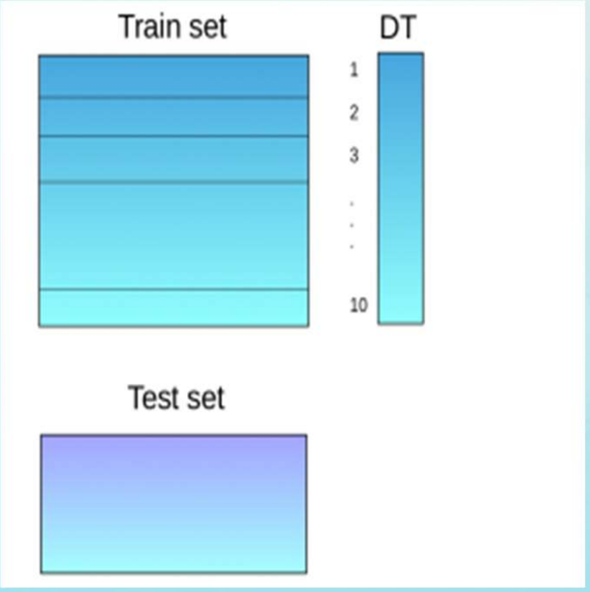
- In gradient boosting, it trains many model sequentially.
- Each new model gradually minimizes the loss function (y = ax + b + e, e needs special attention as it is an error term) of the whole system using Gradient Descent method.
- The learning procedure consecutively fit new models to provide a more accurate estimate of the response variable.
- The principle idea behind this algorithm is to construct new base learners which can be maximally correlated with negative gradient of the loss function, associated with the whole ensemble.

Stacking

- Stacking is an ensemble learning technique that uses predictions from multiple models (for example decision tree, knn or svm) to build a new model.
- This model is used for making predictions on the test set.
- Below is a step-wise explanation for a simple stacked ensemble:
  1. The train set is split into 10 parts.

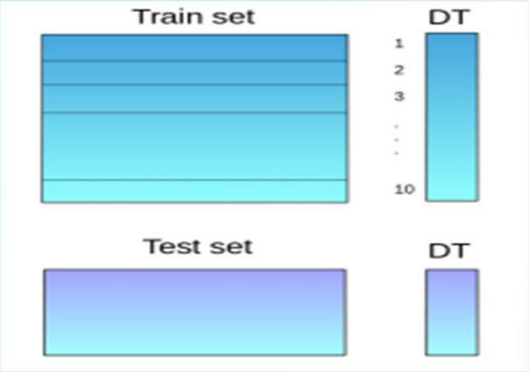


2. A base model (suppose a decision tree) is fitted on 9 parts and predictions are made for the 10th part. This is done for each part of the train set.

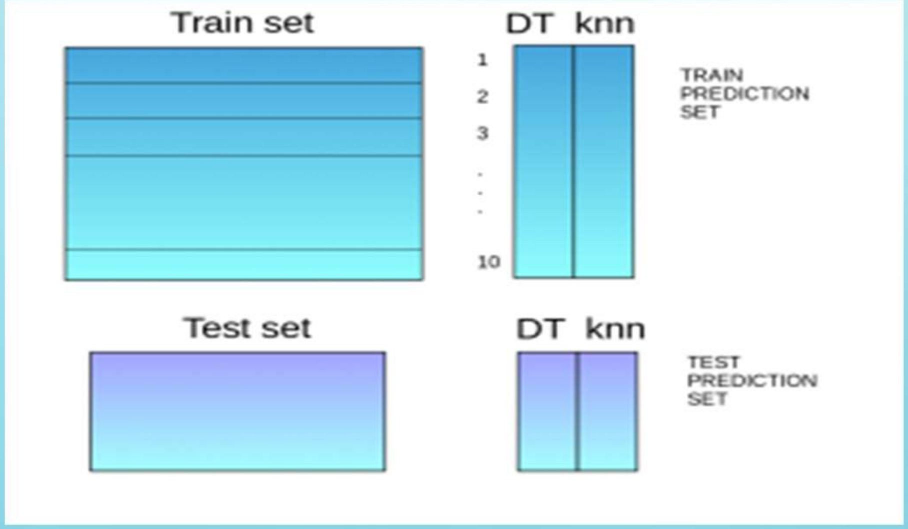


3. The base model (in this case, decision tree) is then fitted on the whole train dataset.

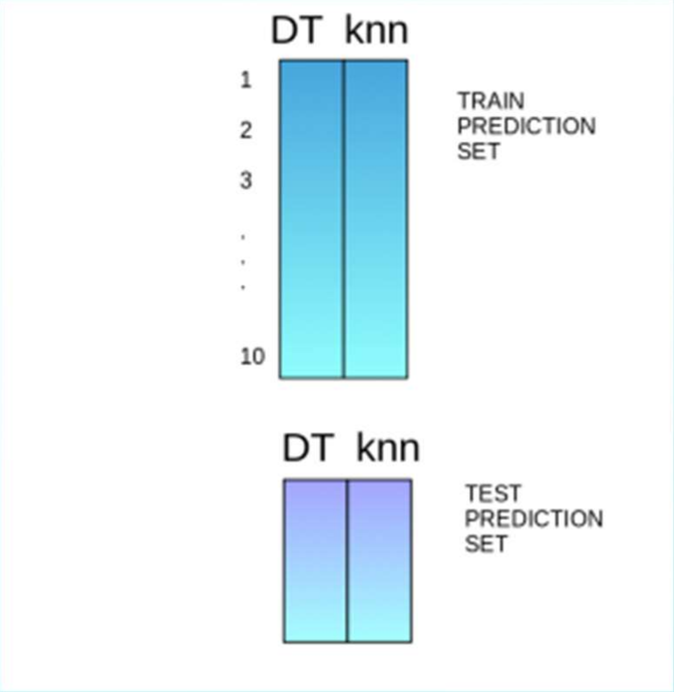
4. Using this model, predictions are made on the test set.



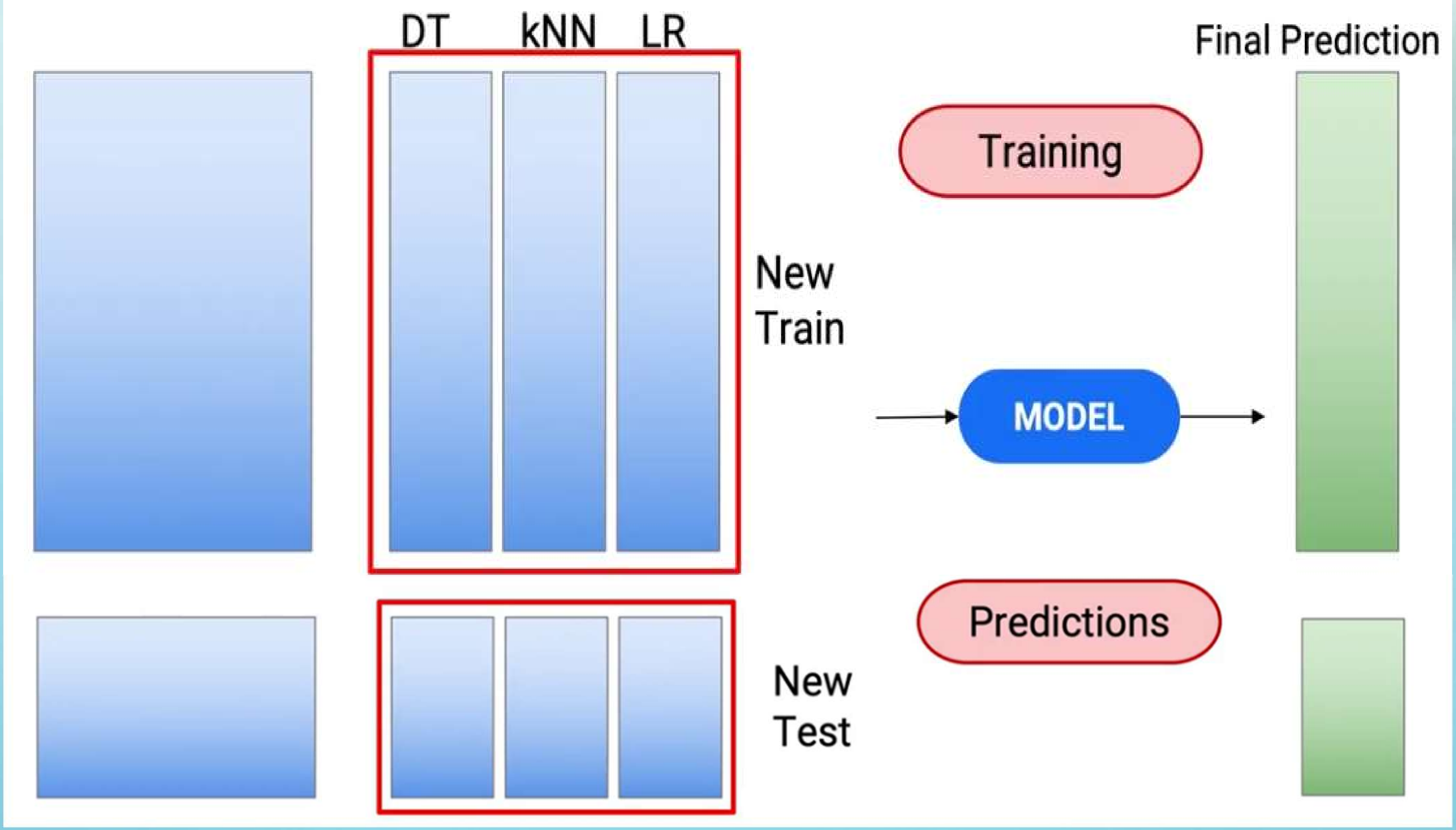
5. Steps 2 to 4 are repeated for another base model (say knn) resulting in another set of predictions for the train set and test set.



- The predictions from the train set are used as features to build a new model.



- This model is used to make final predictions on the test prediction set.


DT
kNN
LR
Final Prediction
Training
New Train
MODEL
Predictions
New Test

Similarities Between Bagging and Boosting

- Bagging and Boosting, both being the commonly used methods, have a universal similarity of being classified as ensemble methods. Here we will explain the similarities between them.

- 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 i.e Majority Voting).

- Both are good at reducing variance and provide higher stability.

Differences Between Bagging and Boosting

S.NO	Bagging	Boosting
1.	The simplest way of combining predictions that belong to the same type.	A way of combining predictions that belong to the different types.
2.	Aim to decrease variance, not bias.	Aim to decrease bias, not variance.
3.	Each model receives equal weight.	Models are weighted according to their performance.
4.	Each model is built independently.	New models are influenced by the performance of previously built models.
5.	Different training data subsets are randomly drawn with replacement from the entire training dataset.	Every new subset contains the elements that were misclassified by previous models.
6.	Bagging tries to solve the over-fitting problem.	Boosting tries to reduce bias.
7.	If the classifier is unstable (high variance), then apply bagging.	If the classifier is stable and simple (high bias) the apply boosting.
8.	Example: The Random Forest model uses Bagging.	Example: The AdaBoost uses Boosting techniques