VIT-AP UNIVERSITY

Ensemble Methods

-

Random Forest Algorithm

Dr. Kuppusamy .P
Associate Professor / SCOPE

SAMPLING

Sampling is process of selecting a subset of the training examples from the dataset for training the model.

Types of Sampling

Simple random sampling

- Subset is selected randomly from the dataset
- Each training example has equal probability of selection from the dataset.
- It takes more time.

Sampling without replacement

- Once a training example / row is selected for a single model, it is removed from the dataset. The selected sample will not be used for another model.

Sampling with replacement

- A selected training example is not removed from the dataset. The selected training example can be used for another models.

Stratified sampling:

- Partition the data set into strata (i.e., divided using gender or age, etc.,), and draw samples from each partition proportionally, i.e., approximately the same percentage of the data.

Ensemble Learning

- Let consider, developing the mobile application for online learning like MSTeams.
- The developer will get preliminary feedback (ratings) about the application before post in google playstore.

- **Possible approaches for receiving feedback:**

  1. *Asking the friend to rate the app.*
  2. *Feedback from Five software developers (from other team).*
  3. *Feedback from 50 people .*

- Diverse group of people are likely to make better decisions compared to individuals.
- Likewise, diverse set of models provides **better performance** compared to single model.
- In Machine Learning, the diversification is achieved using Ensemble Learning.

Basic Ensemble Techniques

1. **Max Voting**
2. **Averaging**
3. **Weighted average**

Max Voting:

- The max voting is generally used for classification problems.
  - Multiple models are used to make predictions for each data example.
  - The predictions of each model are considered as a ‘vote’.
  - Finally, the majority of the model’s predictions are used as the final prediction.
- E.g., *Feedback from Five software developers (from other team).*
- Three are rated as 4 while two of them rate as 5. The majority gave a rating of 4. The final rating will be taken as 4.

Developer1	Developer2	Developer3	Developer4	Developer5	Final Prediction
5	4	5	4	4	**4**

Averaging - Basic Ensemble Technique

- Multiple predictions are made for each data point in averaging.
- Calculate an average of predictions from all the models to make the final prediction.
- **Averaging** can be used for making predictions in **regression** problems or while calculating **probabilities** for **classification** problems.
- E.g., *Feedback from Five software developers (from other team).*
  - Three are rated as 4 while two of them rate as 5. The majority gave a rating of 4. The final rating will be taken as 4.
  - Average = (5 + 4 + 5 + 4 + 4) / 5 = **4.4**

Developer1	Developer2	Developer3	Developer4	Developer5	Final Prediction
5	4	5	4	4	**4.4**

Weighted Average - Basic Ensemble Technique

- All models are assigned **different weights (between 0 and 1)** defining the importance of each model for prediction.

Choose weights:

- Weights are selected based classification accuracy or negative error in which large weights denotes model performance is good.
- Performance is calculated on the dataset used for training or a holdout(test) dataset. Test dataset is more relevant to measure the performance.
- The scores of each model can be used directly or converted into a different value, such as the relative ranking for each model.
- Search algorithm can be used to test different combinations of weights.

Weighted Average - Basic Ensemble Technique

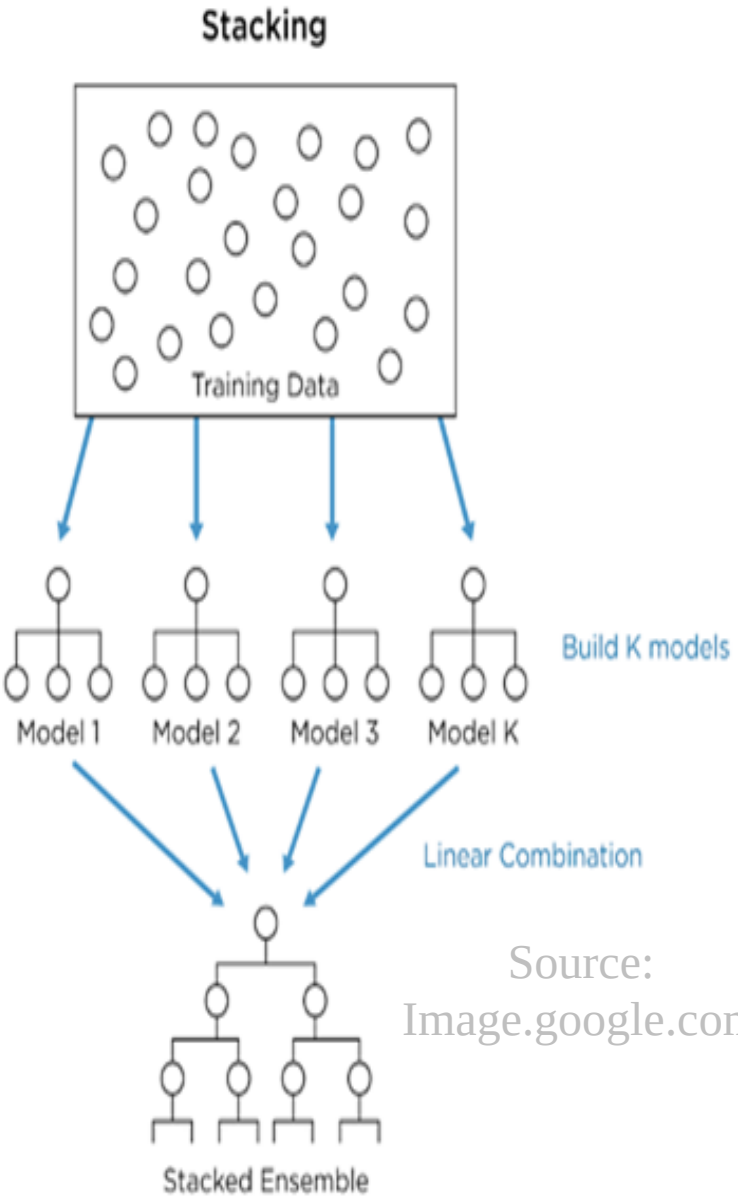
- Let consider, if two developers are experienced, while others have no prior experience in this field, then the answers by these two developers are given more importance compared to others.
- E.g., *Feedback from Five software developers (from other team).*
  - Three are rated as 4 while two of them rate as 5. The majority gave a rating of 4. The final rating will be taken as 4.
  - Average = (5 *0.35) + (4 *0.35)+ (5*0.1) + (4 *0.1) + (4 *0.1)
    = 1.4 + 1.4 + 0.5 + 0.4 + 0.4
    = 4.1

	Developer1	Developer2	Developer3	Developer4	Developer5	Final Prediction
Weight	0.35	0.35	0.1	0.1	0.1	
Rating	5	4	5	4	4	**4.1**

Advanced Ensemble Techniques

1. **Stacking**
2. **Blending**
3. **Bagging**
4. **Boosting**

- **Stacking:** It combines predictions from multiple (base-level) models to build a new model (meta-model). This meta-model is used for making predictions on the test set.
- Base level algorithms are trained based on a complete training data-set using k-fold cross validation.
- Meta model is trained on the predictions combination of all base level model as feature.
- Stacking is useful when the results of the individual algorithms are very different.

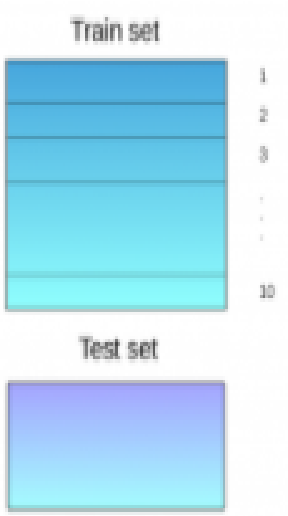


Source: Image.google.com

Blending Ensemble Technique

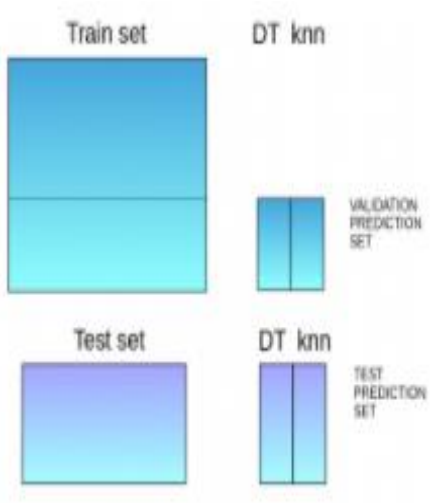
Blending follows the same approach as stacking but uses only a holdout (validation) set from the train set to make predictions.
In other words, unlike stacking, the predictions are made on the holdout set only. The holdout set and the predictions are used to build a model which is run on the test set.

Step 1:
First Data is divided into train and test set.



Step 2:
The train set is split into training and validation sets.

Step 3:
Base models are fitted on the training set and predictions are made on the validation set and test set.



Step 4:
The validation set and its predictions are used as features to build a new model and this model is used to make final predictions on the test features.

Bagging Ensemble Technique

- Bagging is combining the results of multiple models (for example, all decision trees) to get a generalized result.
- If **all the models** utilizes the **same set of data** and combine it, will it be useful?
  - Mostly, All models provides the **same result** due to the **same dataset as input**.

- **Bootstrapping** solves this problem.
- In Bootstrapping, **divide dataset** into **few subsets** with data examples replacement (row sampling). **Bagging /Bootstrap Aggregating uses** these subsets (bags) to get better distribution (complete set). The size of subsets less than the original dataset.

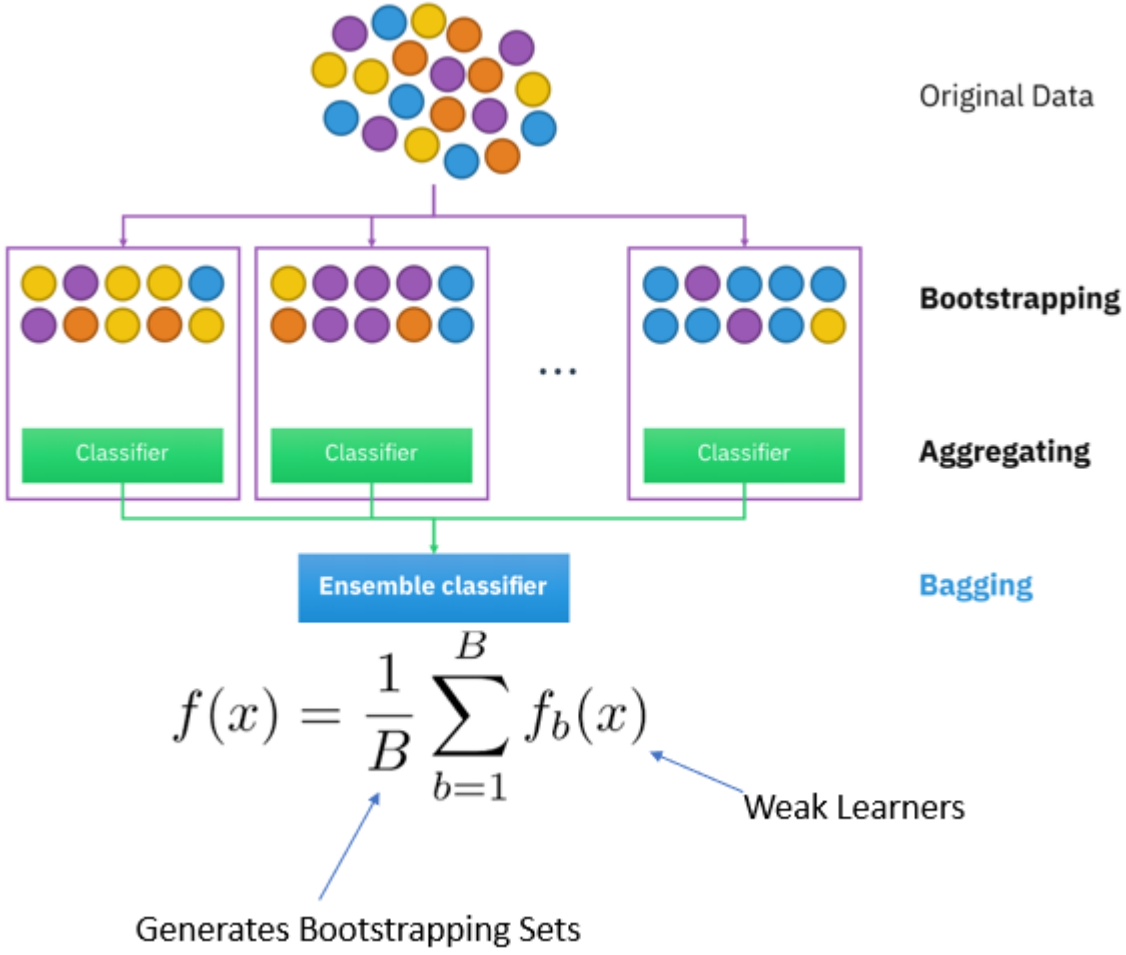
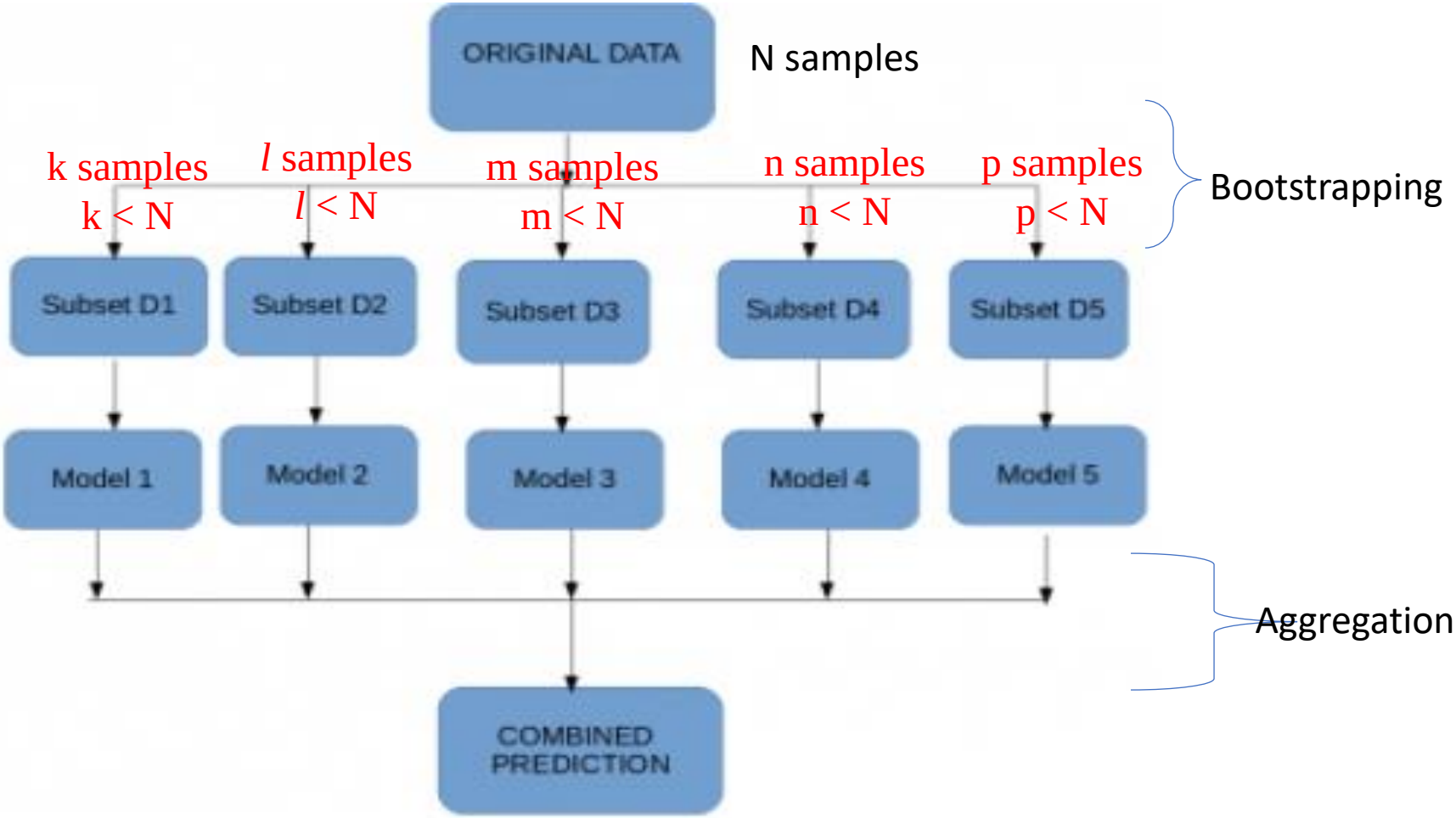


Image Source: analyticsvidhya

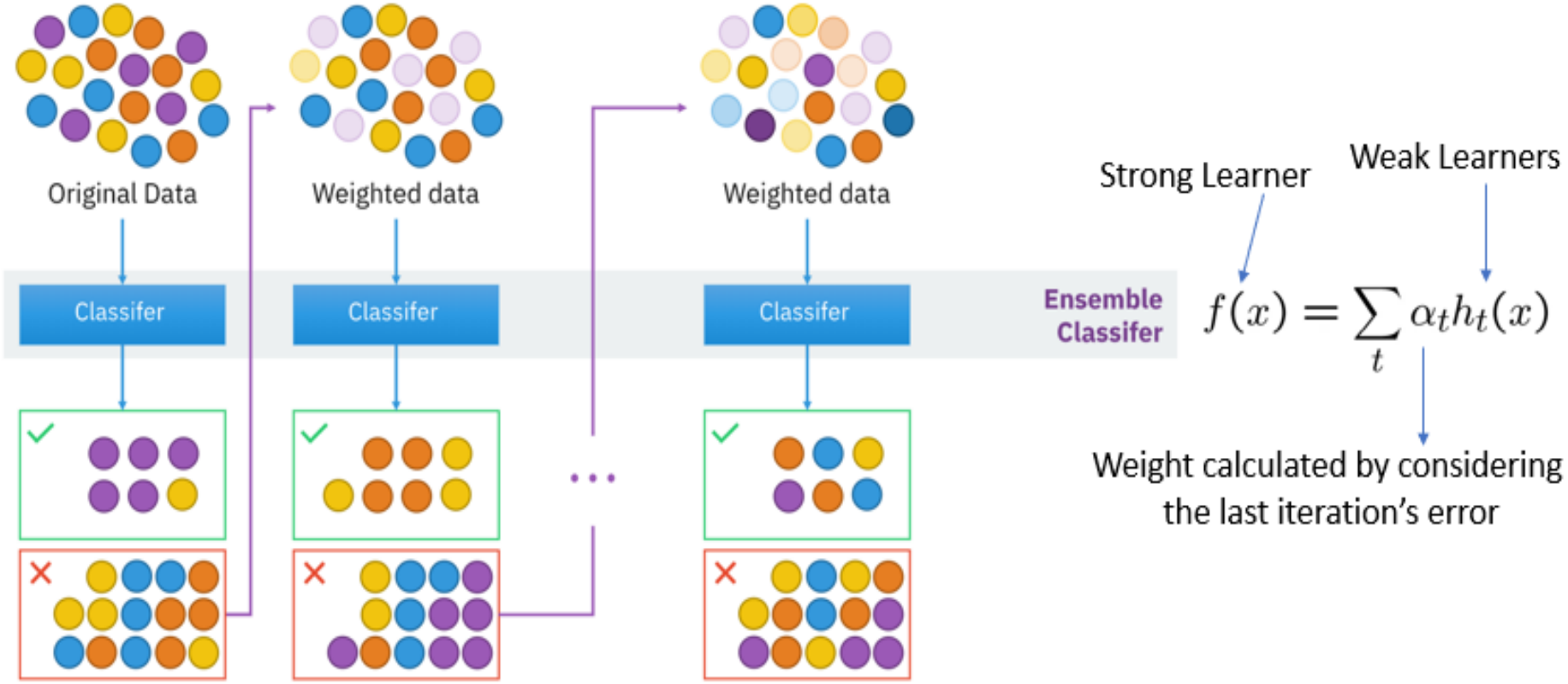
Bagging Ensemble Technique

- A base model (weak model) is created on each of these subsets (subsets are selected using sampling process).
- The models run in parallel and are independent of each other.
- The final predictions are determined by combining the predictions from all the models.



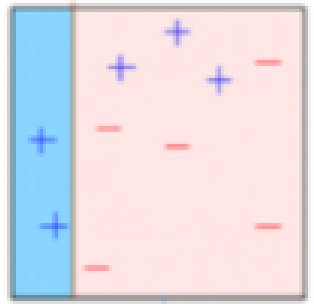
Boosting Ensemble Technique

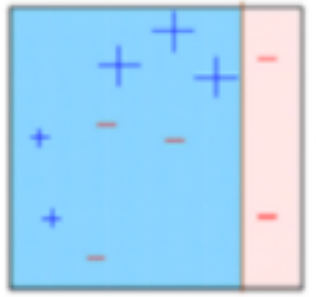
- If a data point is incorrectly predicted by the first model, and then the next model (probably all models), will combining the predictions provide better results?
- It is solved by **boosting.**
- **Boosting** is a sequential process, where each subsequent model attempts to correct the errors of the previous model.
- The succeeding models are dependent on the previous model.



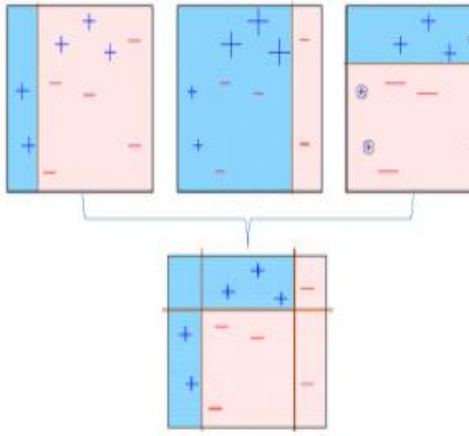
Boosting Ensemble Technique

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.
5. Errors are calculated using the actual values and predicted values.
6. The observations which are incorrectly predicted, are given higher weights. (E.g., three misclassified blue-plus points will be given higher weights)
7. Another model is created and predictions are made on the dataset. (E.g. 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)
   - 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.





weak learners



strong learner

Ensemble Technique based on Bagging and Boosting

- **Bagging / Bootstrap Aggregation algorithms:**
    1. Bagging meta-estimator
    2. Random forest
- **Boosting algorithms**:
    1. AdaBoost
    2. GBM
    3. XGBM
    4. Light GBM
    5. CatBoost

RANDOM FOREST ALGORITHM

Random Forest Algorithm – Ensemble Learning

- Ensemble learning is combining multiple learning algorithms (classifiers / learners) to improve the performance is solving the problem.
- Random Forests is supervised learning method for classification & regression

- Random Forest contains several decision trees that trained over various subsets of the given dataset and computes the average or majority votes to improve the predictive accuracy of that dataset.

Need of Random Forest:

- The single Decision Tree results the Low bias (Low Error in training data) and High Variance (High Error in testing data).
- But Many trees in the forest provides **higher accuracy** and **prevents** the problem of **overfitting.**
- The prediction of the decision tree is in the mode of
  - Classes for classification, and
  - Numerical value for regression

Random Forest Algorithm

1. Collects the training dataset
2. Divide the training dataset into **subsets** randomly using row sampling and feature sampling. (Number of subsets equal to the number of decision trees to be grown)
3. Assume each subset contains **‘m’** random training examples and **n** features.
4. Build a decision tree for first subset of training dataset.
5. Repeat the step 3 and 4 until build the decision tree for each subset.
6. Ensemble of the decision trees generated is the Random Forest.

Hyperparameters in Random Forest Algorithm

- **N_estimators:** Number of decision trees in the forest. Default values: 100.
- N_estimators are mostly correlated to the dataset size. To encapsulate the **trends** in the data, more number of DTs are needed.
- **Criterion:** Measure the quality of splits. For classification, Gini impurity **or** entropy denotes information gain. For Regression, Mean Absolute Error (MAE) or Mean Squared Error (MSE). Default is gini and MSE.
- **Max_depth:** Maximum levels allowed in a decision tree. If set to nothing, DT will keep on splitting until purity is reached.
- **Max_features:** Maximum number of features used for a node split process. **Types:** sqrt, $\log_2$. If total features are n_features then: sqrt(n_features) or log2(n_features) can be selected as max features for node splitting.
- **Bootstrap:** Bootstrap samples (subsets) are used when building decision trees. if True is selected in bootstrap, else entire dataset is used for every decision tree.

Hyperparameters in Random Forest Algorithm

- **Min_samples_split:** Minimum samples required to split an internal node. Default value =2. For small value, condition is checked on the terminal node. If the data points in the node >2, proceeds the further splitting. Whereas if a more lenient value like 6 is set, then the splitting will stop early and the decision tree wont overfit on the data.

- **Min_sample_leaf:** Minimum data point requirements in a node. It controls the depth of the tree. If data points in a node < min_sample_leaf number, the split will be stopped at the parent node.

Less important parameters

- **n_jobs:** Number of processors can be used for training. (-1 for no limit)
- **max_samples:** Maximum data that can be used in each Decision Tree
- **random_state**: the model with a specific random_state will produce similar accuracy/ outputs.
- **Class_weight:** dictionary input, that can handle imbalanced data sets.

Hyperparameter Tuning

- Hyperparameters are tuned to increase the specific metrics such as accuracy or f1 score using GridSearchCV and RandomSearchCV.

Graphical Representation – Random Forest

Fruits classification Problem

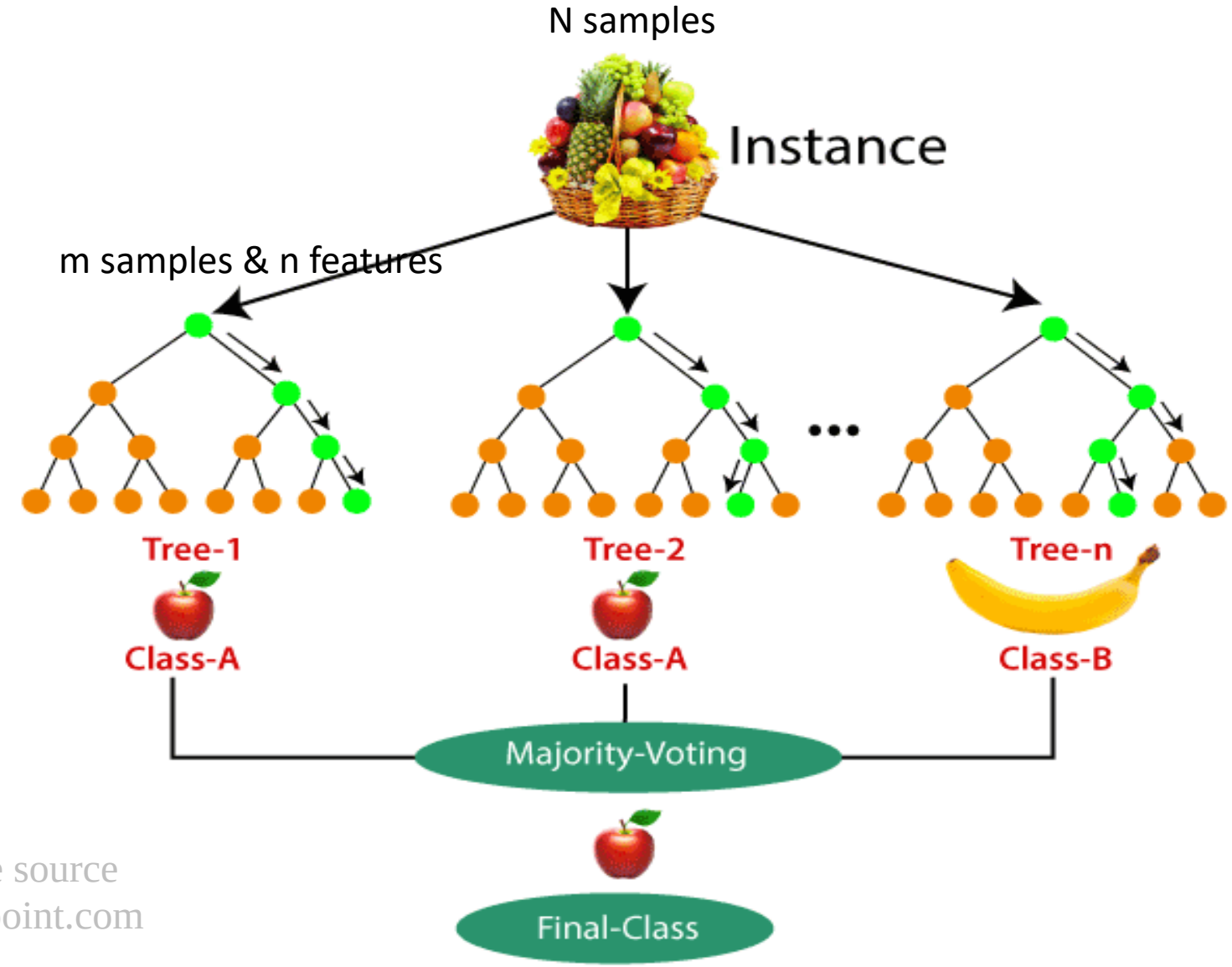


Image source
javatpoint.com

Applications

- Banking: Identification of loan risk and Credit card fraud detection
- Manufacturing Quality Control
- Medicine: Identification of disease.
- Marketing: Marketing trends identification and Retail Sales prediction

Advantages

- Random Forest handles Classification and Regression tasks.
- Improves the accuracy and prevents the overfitting.
- Handles large datasets with high dimensionality

Disadvantages

- Large memory needed to store built models.
- It is not much suitable for Regression problems.

Difference of Decision Tree and Random Forest algorithm

SNo	Decision trees	Random Forest
1	Unable to solve overfitting if tree is allowed to grow without any control.	Solves the overfitting due to Random forests are created from subsets of data and the final output is based on average or majority ranking.
2	A single decision tree is faster in computation.	It is comparatively slower.
3	When a data set with features is taken as input by a decision tree it will formulate some set of rules to do prediction.	Random forest randomly selects observations, builds a decision tree and the average result is taken. It doesn’t use any set of formulas.