RANDOM FOREST CLASSIFIER

Sourav Karmakar

souravkarmakar29@gmail.com

Where Decision Tree Fails

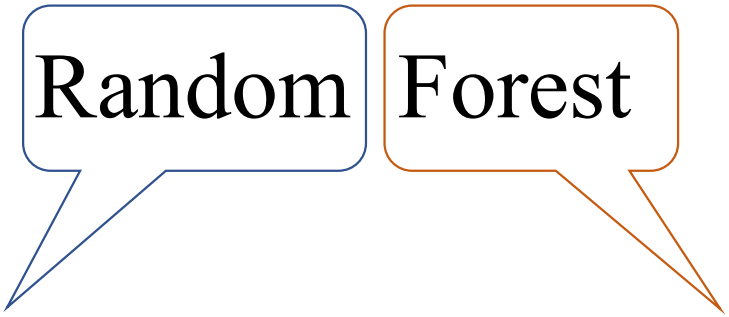
- An ideal classifier should be able to minimize both error in training and test datasets.
- But, decision trees are prone to **overfitting**, especially when a tree is particularly deep in pursuit of designing a perfect tree which classifies the training data very accurately but fails to generalize on unseen data / test data.
- Overfitting results in a decision tree which is more complex than necessary.

One way to deal with the overfitting problem of decision tree is to design a **ensemble** of decision trees, popularly known as **Random Forest**.

In machine learning, **ensemble** refers to a technique where multiple individual models are combined to create a more powerful and accurate predictive model. Instead of relying on a single model, ensemble methods leverage the strengths of various models to improve performance, reduce overfitting, and enhance generalization.

Random Forest

- It is an ***Ensemble Classifier*** made using many decision tree models to make better and more accurate predictions. It emphasizes on *"creating a strong and more accurate classifier by combining many weak learners"*.

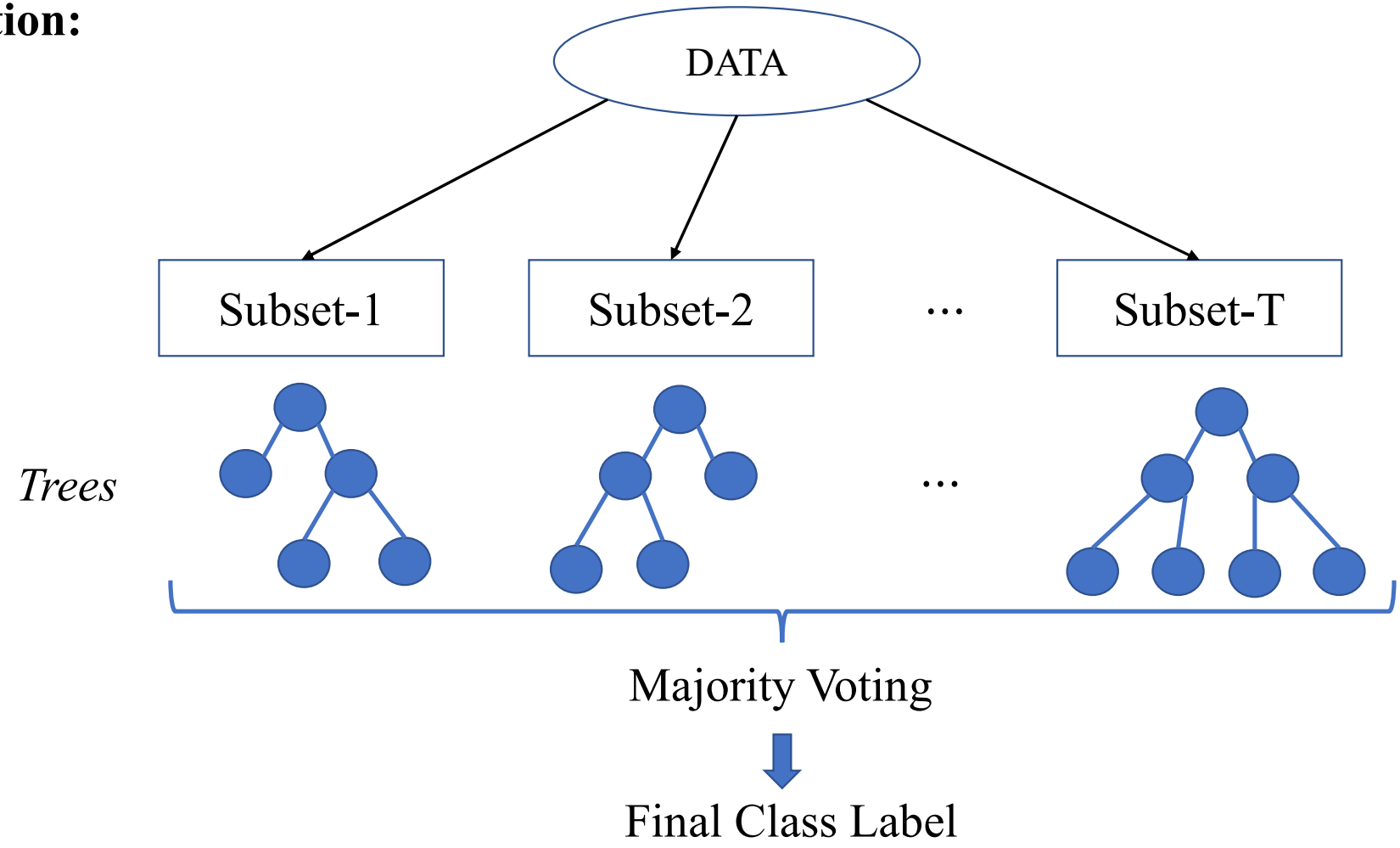


Trees (training set subsets) and features are selected at random with replacement.

It is a collection of decision trees. Hence Forest.

Random Forest

Intuition:



Random Forest

Creation:

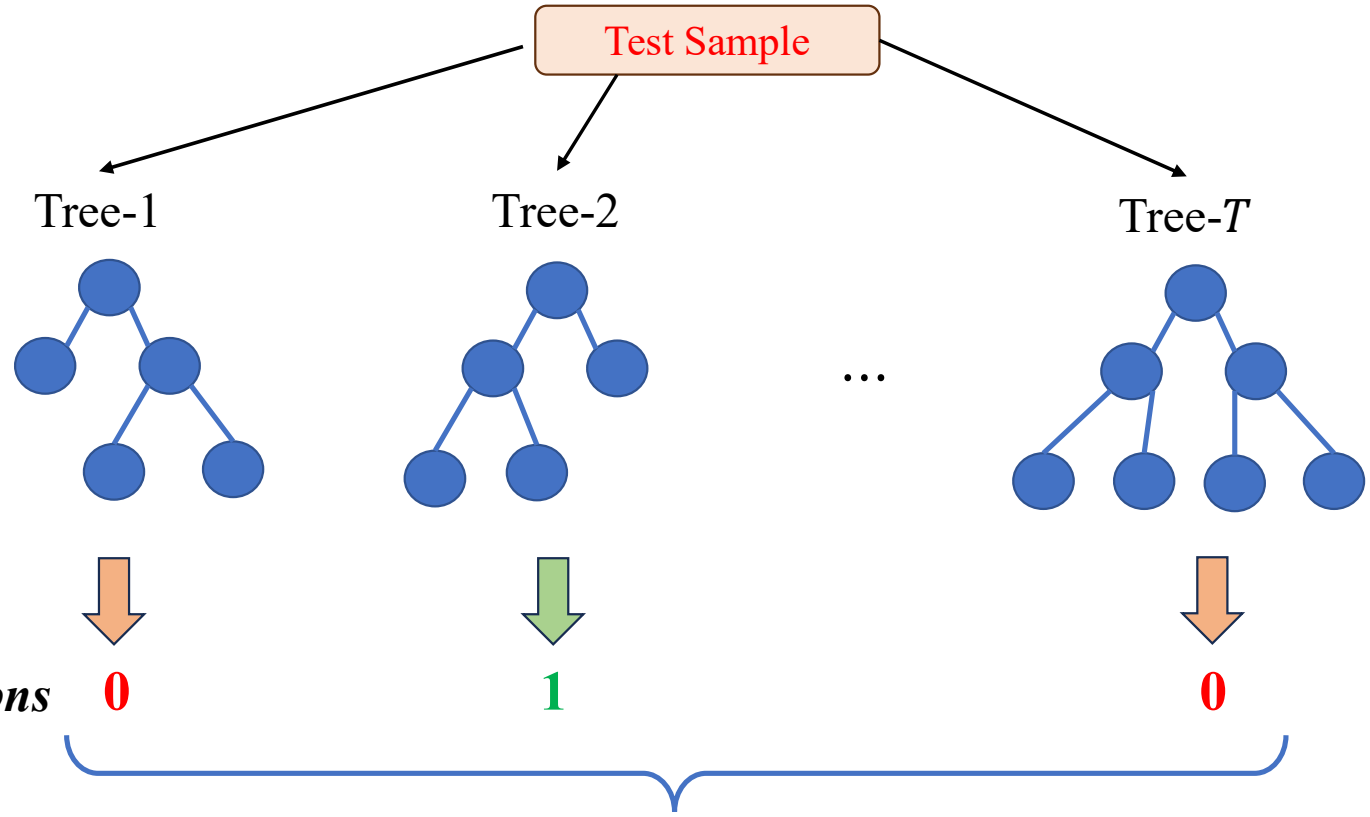
- **Bootstrapping:** Each tree in the forest is trained on a different bootstrap sample of the training data. A *bootstrap* sample is created by randomly sampling with replacement from the original dataset. This means that each tree sees a slightly different subset of the data, which helps to reduce the correlation between the trees.
- **Random Subspace Method:** When growing each individual tree, at each node, only a random subset of the features is considered for the best split. This is a key difference from a standard decision tree, which considers all features at each split. By restricting the features available, the trees are forced to be more diverse and less correlated with one another.

Prediction:

- Take the test sample and run through each constituent decision tree to predict the target.
- Consider the majority voted predicted target as the final prediction from the random forest algorithm for the test sample.

  This method is called **Bagging**, which came form of ***Bootstrap Aggregating*.**

Random Forest



- probability of the test sample to be of class-1:

$$p = \frac{\tau}{T}$$

- If $p \geq 0.5$ predict the class of the test sample as class-1 else class-0.

Random Forest

Number of Decision Trees in a Random Forest:

- The number of decision trees to decide for a Random Forest is problem specific.
  - Depends largely on the number of data records in the training dataset and the number of features available.
- The "Out of Bag (OOB)" estimate is computed to decide on the number of decision trees which will yield the optimum results. OOB samples are those which are not sampled and not fitted by any decision tree.
- For a particular problem, we create n Random Forests with different number of trees. We record their OOB error rate and see the number of trees where OOB error rate stabilizes and reaches minimum. Finally we choose that Random Forest for our particular problem.

Random Forest

Advantages:

- Better Accuracy as compared to a single decision tree as it inherently reduces overfitting.
- It runs efficiently on large datasets.
- It produces very competitive result as compared to many state of the art classifiers.

Disadvantages:

- Takes longer time to train and test than single decision tree.

Thank You