# 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**.

# 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"*.

Random

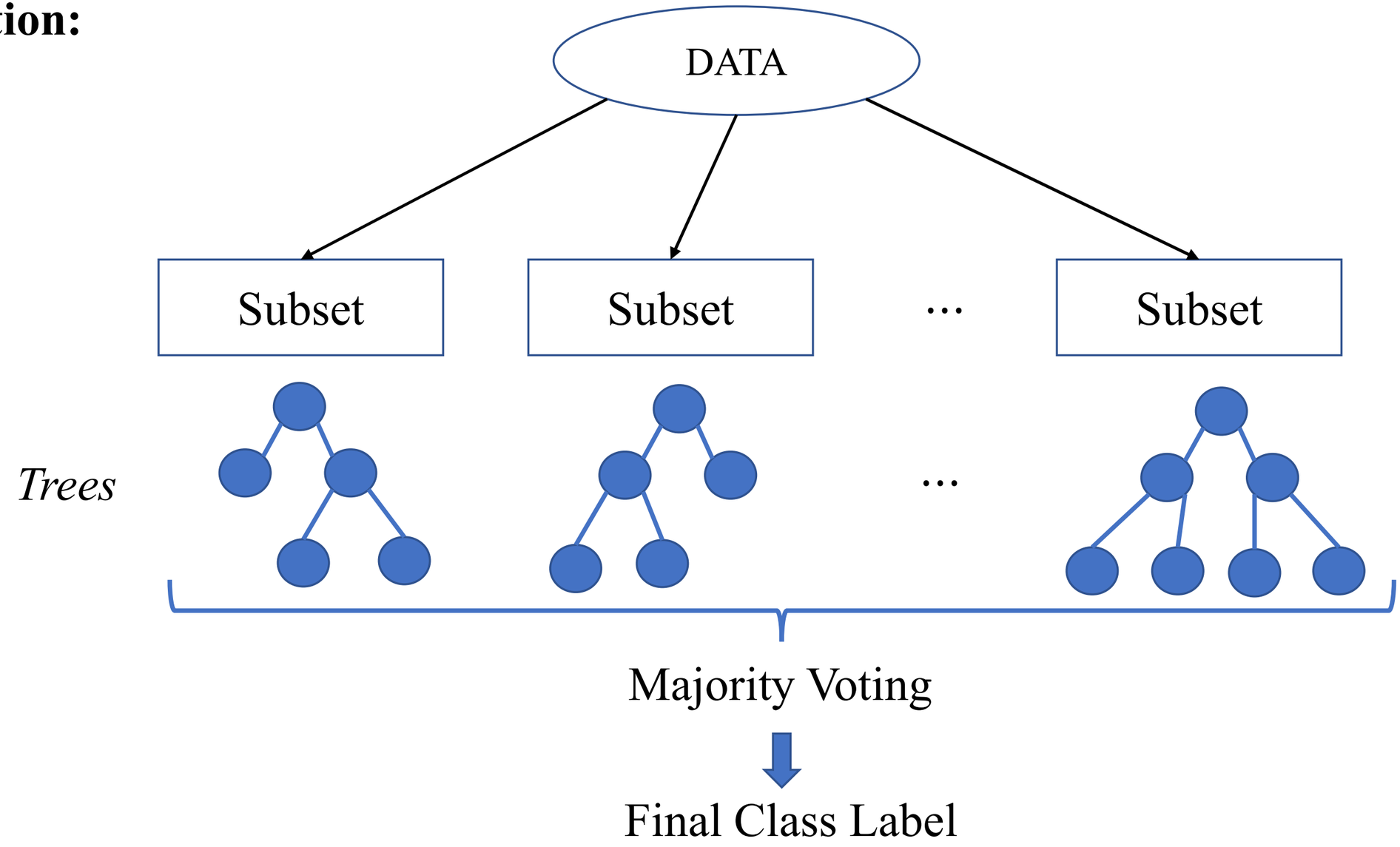
Trees (training set subsets) and features are selected at random with replacement.

Forest

It is a collection of decision trees. Hence Forest.

# RANDOM FOREST

**Intuition:**



# Bootstrapping :-

Dataset

| Sl No. | $x_1$ | $x_2$ | $x_3$ | .... | $x_k$ | $y$ |
|---|---|---|---|---|---|---|
| 1. | | | | | | |
| 2. | | | | | | |
| 3. | | | | | | |
| 4. | | | | | | |
| - | | | | | | |
| - | | | | | | |
| 1000 | | | | | | |

We will select random subsets (of same number of rows) from the data with replacement.

Sample - 1

| Sl No. | $x_2$ | $x_3$ | $x_5$ | $y$ |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | | | |
| : | | | | |
| 700 | | | | |

Sample - 2

| Sl No. | $x_1$ | $x_3$ | $x_4$ | $y$ |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | | | |
| : | | | | |
| 700 | | | | |

col_sampling_ratio = 0.7 (user defined)

Suppose you have 20 columns in the dataset. Then in the sample you will have $(0.7 \times 20) = 14$ columns. at random

row_sampling_ratio = 0.8 (user defined) fraction of the rows to be used in resampling.

# RANDOM FOREST

BAGGING

Bootstrap

Aggregating

**Creation:**

- **Bootstrapping:** It is a kind of *resampling* where large numbers of smaller samples of the same size are repeatedly drawn, with replacement from a single original large dataset.
- **Tree Bagging:** Randomly select '$k$' features from total $'T'$ features in the dataset ($k < T$)
- **Fit Decision Trees:** Fit decision tree for each of the bootstrapped sample and with reduced number of randomly selected features.

**Prediction:**

- Take the test sample and use the rules of each randomly created decision tree to predict the outcome and store the target.
- Calculate the vote for each predicted target.
- Consider the majority voted predicted target as the final prediction from the random forest algorithm.

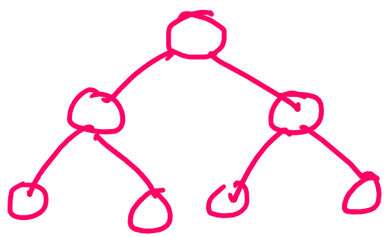
yes → 1
no → 0

Data

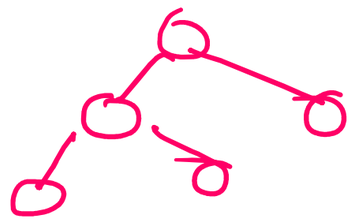
Sample-1

Sample-2

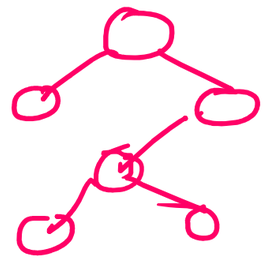
Sample-n



DT-1
(No)

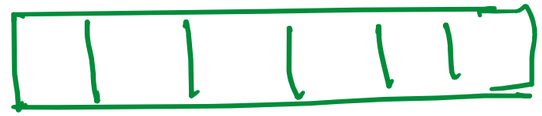


DT-2
(YES)

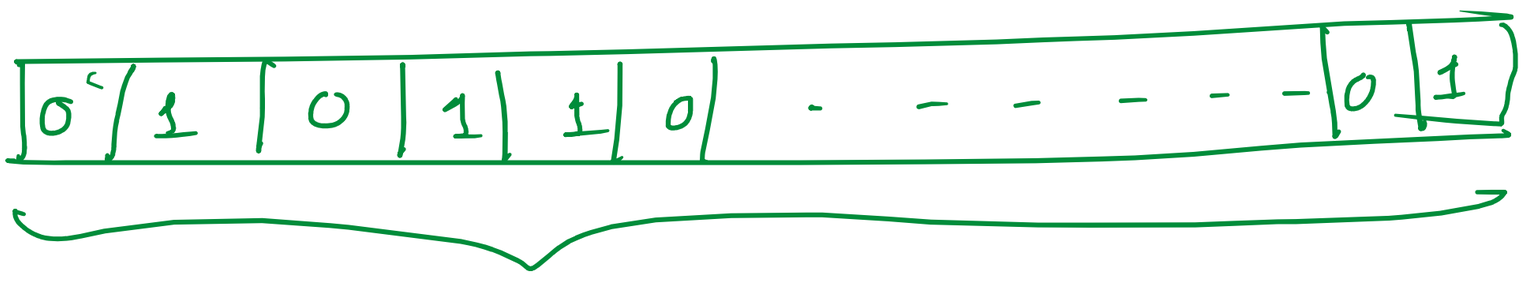


DT-n
(YES)

test instance (t)



$P(yes|t) = \frac{m}{n}$




array length = n

Count the occurences of 1 & 0.

1 → m

0 → n-m

# 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***