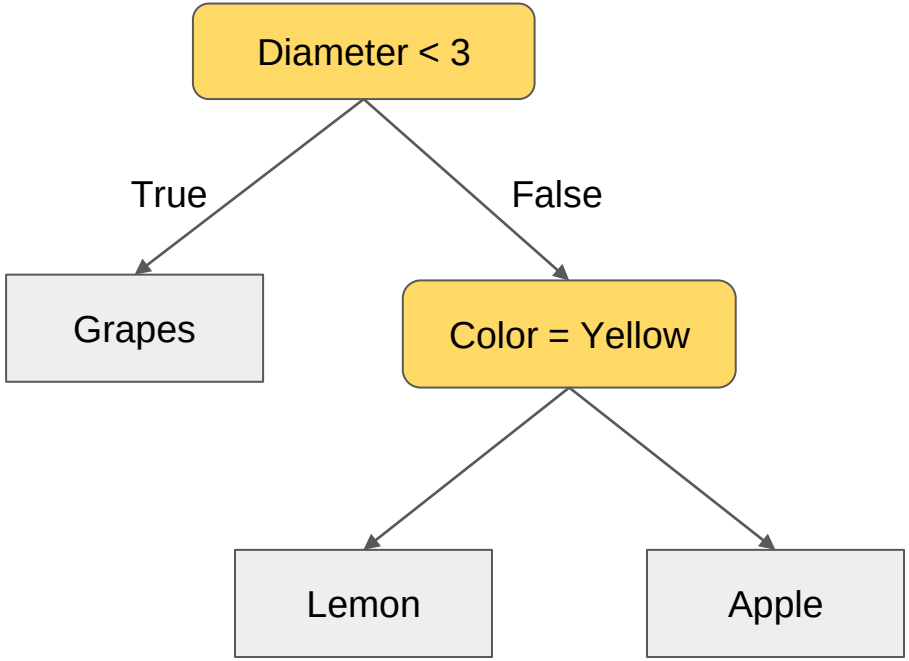
RANDOM
FORESTS

INNOMATİCS
RESEARCH LABS

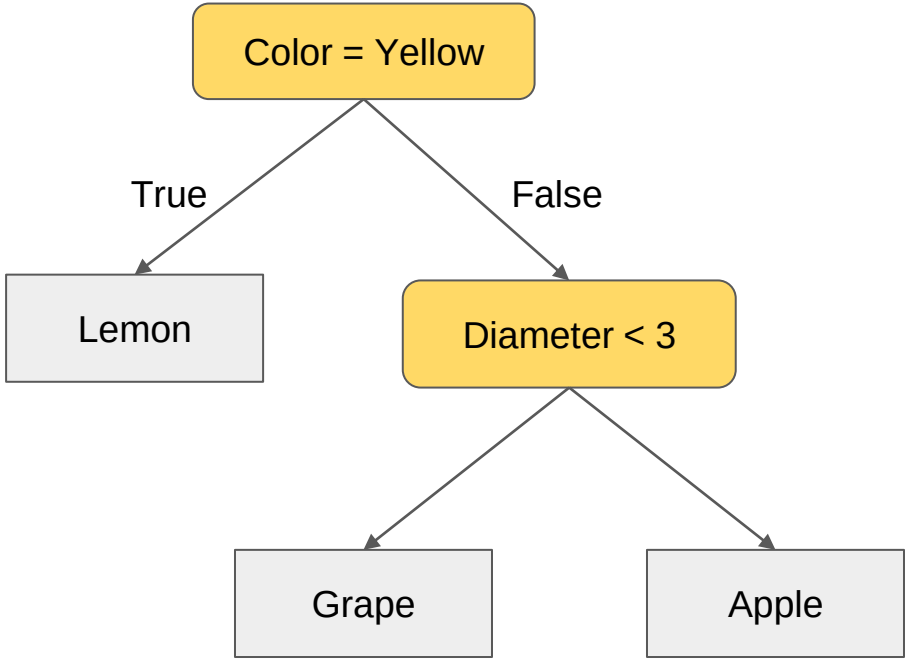
DECISION TREE: RECAP

Color	Diameter	Label
Red	3	Apple
Yellow	3	Lemon
Purple	1	Grapes
Red	3	Apple
Yellow	3	Lemon
Purple	1	Grapes



ALTERNATE DECISION TREE

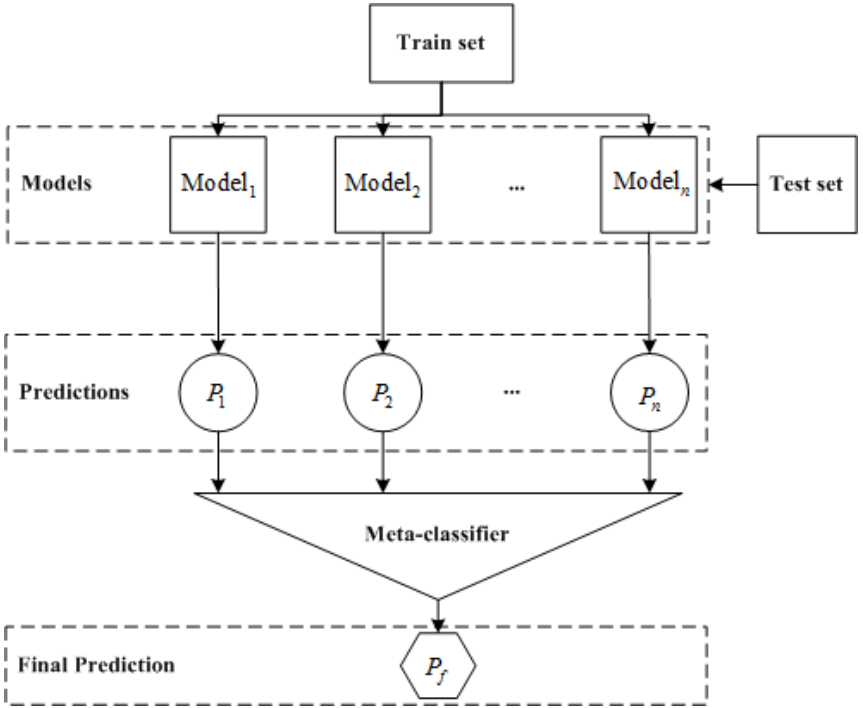
Color	Diameter	Label
Red	3	Apple
Yellow	3	Lemon
Purple	1	Grapes
Red	3	Apple
Yellow	3	Lemon
Purple	1	Grapes



ENSEMBLE LEARNING

Ensemble Learning is the process of combining multiple models with relatively lower accuracy in order to create a system that eventually produces a high accuracy.

We explicitly use ensemble learning to seek better predictive performance, such as lower error on regression or high accuracy for classification

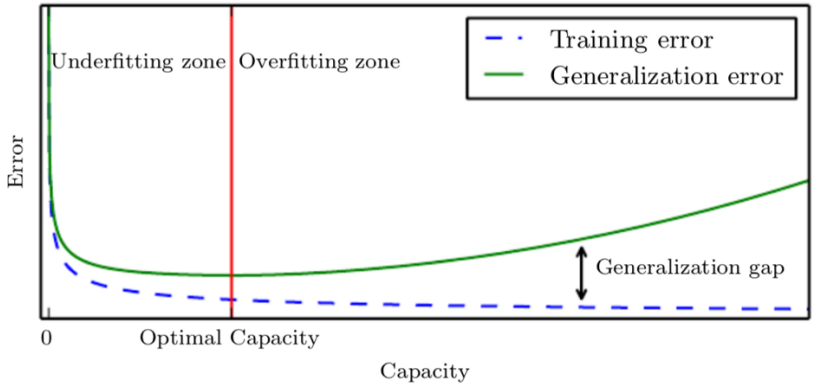


ENSEMBLE LEARNING

The fundamental principle of the **ensemble model** is that **a group of weak learners come together to form a strong learner, which increases the accuracy of the model.** When we try to **predict the target variable by any machine learning technique**, the **main causes of the difference between the actual and predicted values are noise, variance and bias**. The set reduces these factors (except noise, which is an irreducible error).

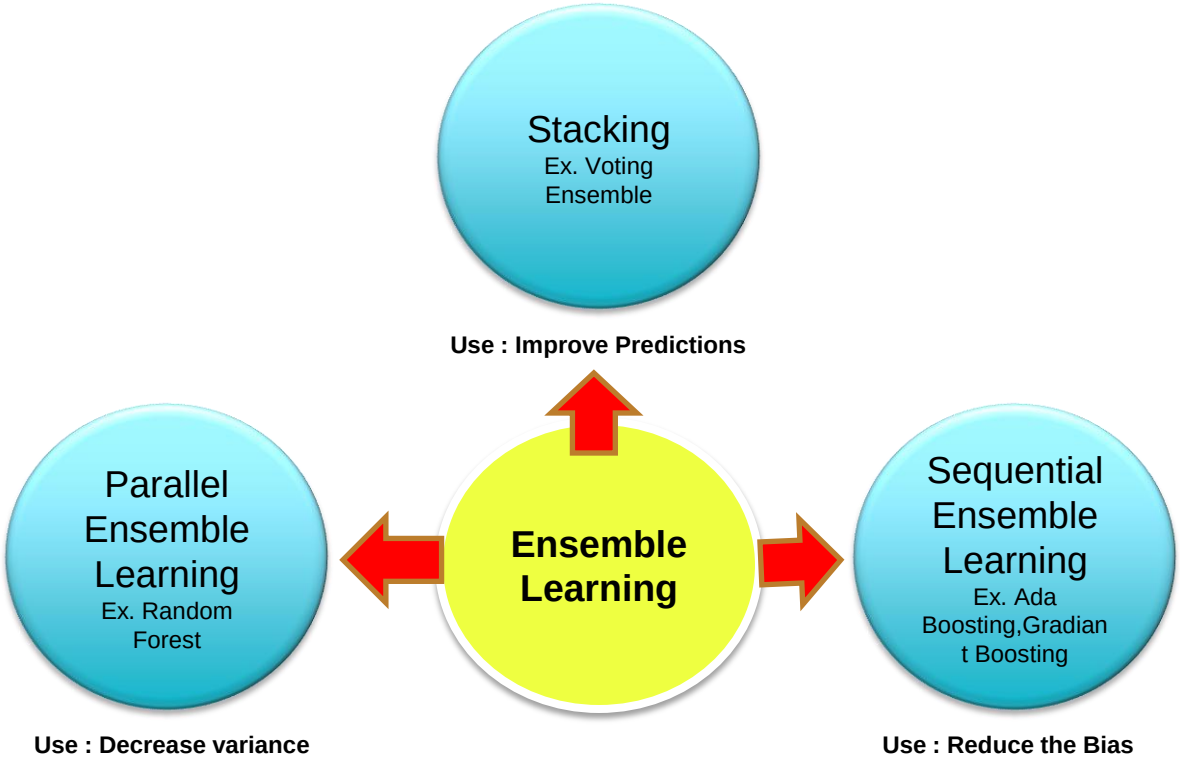
ENSEMBLE LEARNING

Here, we can see that if we will get the the **variance, noise and bias** in the raw data, image or any other format of the data. So, our model is going **either under-fitting or over-fitting.**

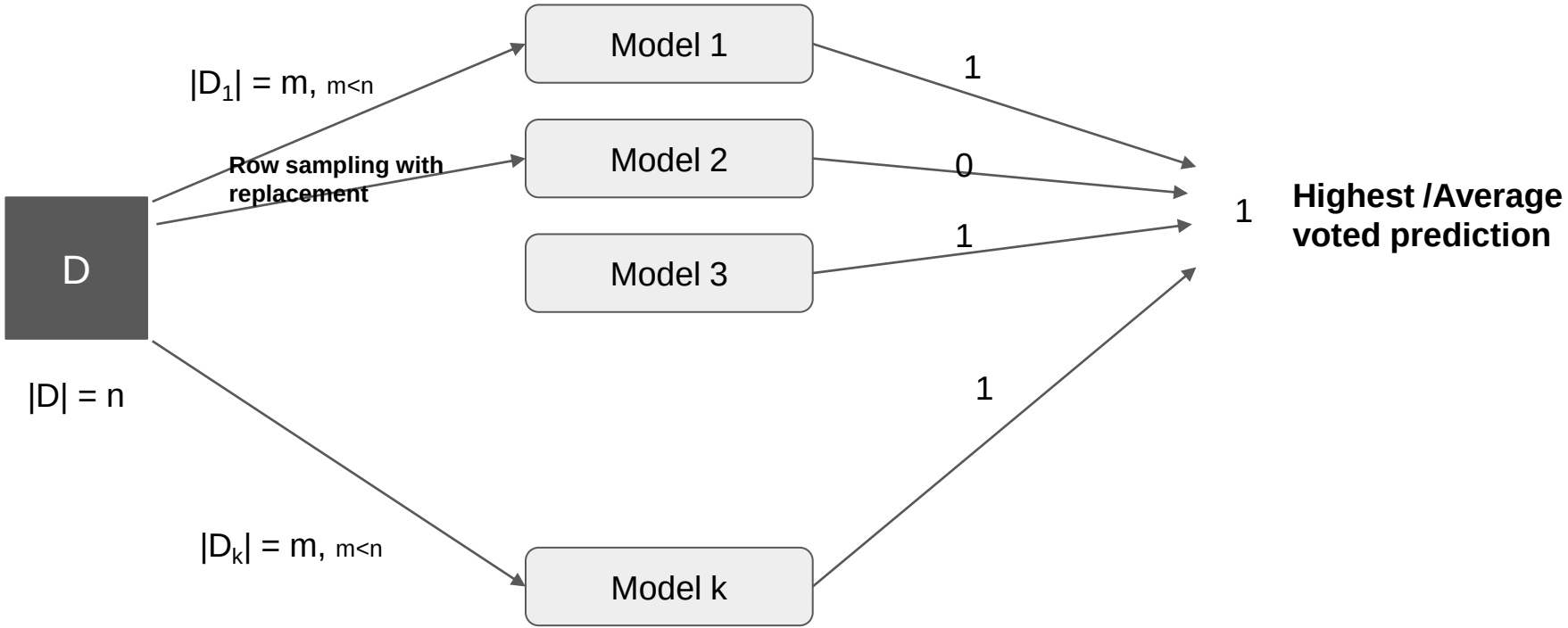


This reason is create big impact on your model directly here the ensemble learning comes in the picture. **Training error** and **generalization error** has gap which is represent as Generalization gap, which is show's that model is **under-fit** or **over-fit.**

ENSEMBLE LEARNING



ENSEMBLE LEARNING: BAGGING



Bootstrapping and Aggregation

ENSEMBLE LEARNING: BAGGING

The image shows that the bagging example has three steps :

- **Bootstrap data(Sampling)**
- **Aggregation or Model fit**
- **Combination different model with Result aggregation.**

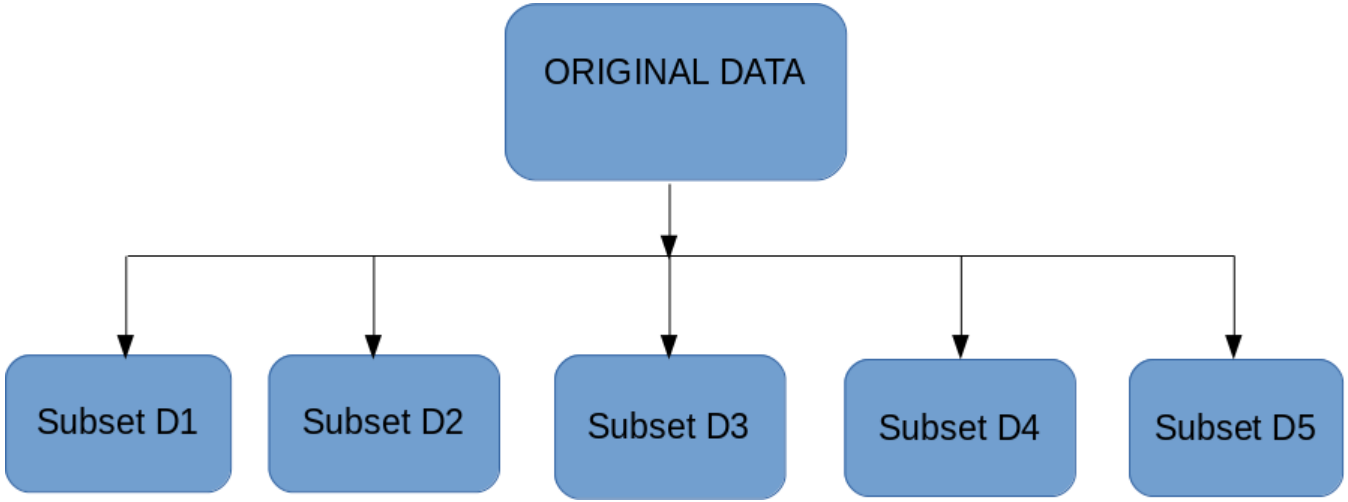
- **Bootstrapping** is a sampling technique in which we create multiple random sample from our training data-set.

ENSEMBLE LEARNING: BAGGING

- The idea behind bagging is combining the results of multiple models (for instance, all decision trees) to get a generalized result.
- Bagging uses a sampling technique called – **Bootstrapping.**
- **Bootstrapping** is a sampling technique in which we create subsets of observations from the original dataset, **with replacement**.
- Bagging (or **B**ootstrap **Agg**regat**ing**) technique uses these subsets (bags) to get a fair idea of the distribution (complete set).
- The size of subsets created for bagging may be same or less than the original set.

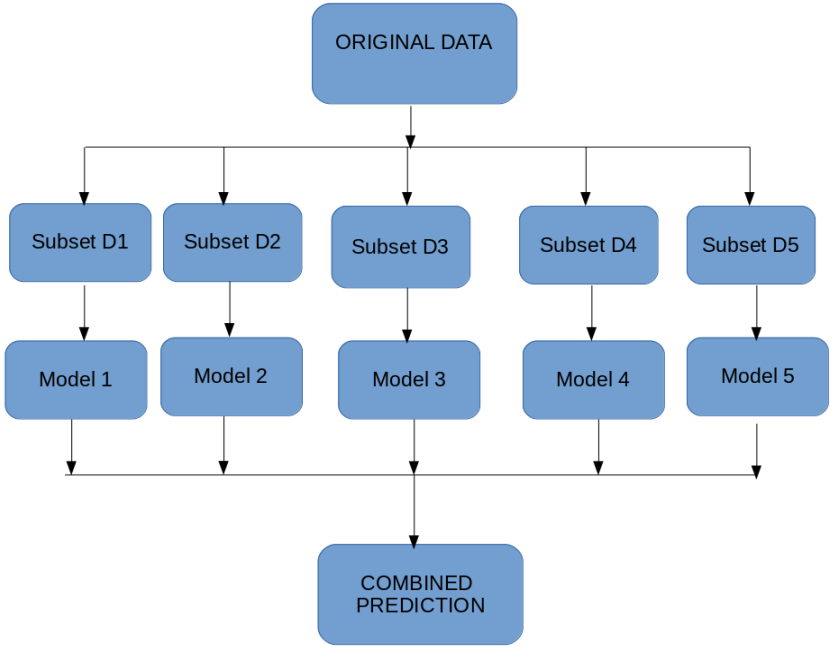
ENSEMBLE LEARNING: BAGGING

- Multiple subsets are created from the original dataset, selecting observations with replacement.



ENSEMBLE LEARNING: BAGGING

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



HOW DOES RANDOM FOREST WORK?

- Decisions trees are very sensitive to the data they are trained on — small changes to the training set can result in significantly different tree structures.
- Random forest takes advantage of this by allowing each individual tree to randomly sample from the dataset with replacement, resulting in different trees – Bagging.
- RF consists multiple decision trees which act as base learners. Each decision tree is given a subset of random samples from the data set (hence the name ***random***).

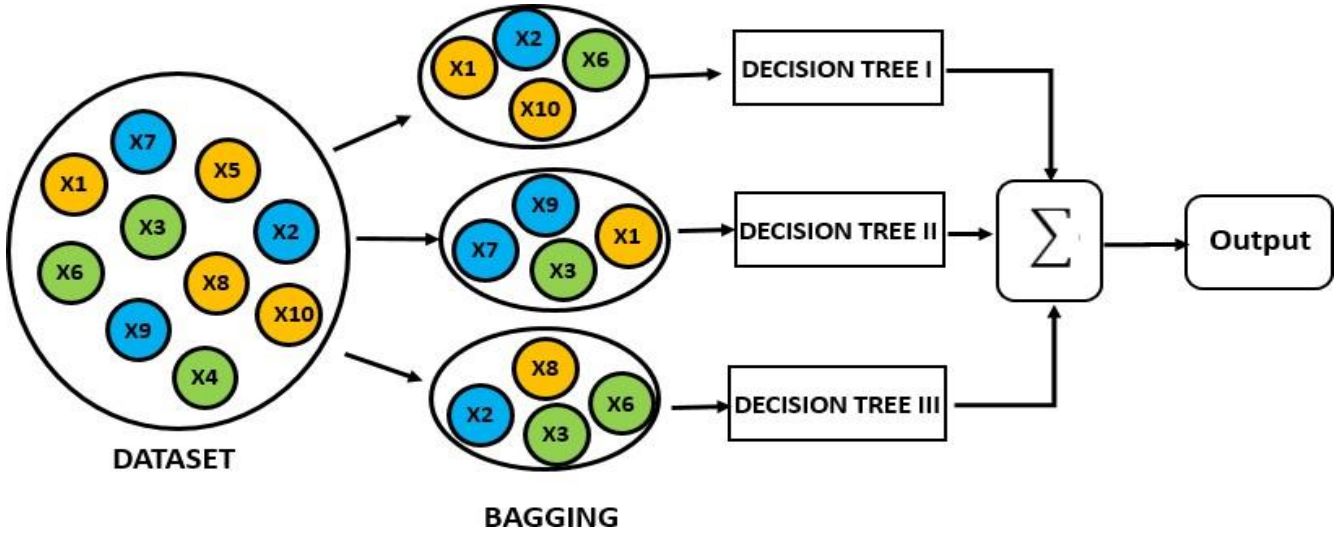
HOW DOES RANDOM FOREST WORK?

- RF algorithm uses an Ensemble method – Bagging (**B**ootstrap **Agg**regat**ing**)

- Then, Random Forest train each base learner (i.e Decision Tree) on a different sample of data and the sampling of data points happens with replacement.

RANDOM FOREST WORKING

- Consider a training dataset : [X1, X2, X3, … X10, Y].
- Random forest will create decision trees taking the input from subset using bagging as shown below:



RANDOM FOREST WORKING

- Note that with bagging we are not sub-setting the training data into smaller chunks and training each tree on a different chunk.
- Rather, if we have a sample of size N or 1000 rows, we are still feeding each tree a training set of size N.
- But instead of the original training data, we take a random sample of size N with replacement.
- For example, if our training data was [10, 11, 12,13, 14, 15, 16] then we might give one of our trees the following list [11, 12,12, 13,13,16, 16].

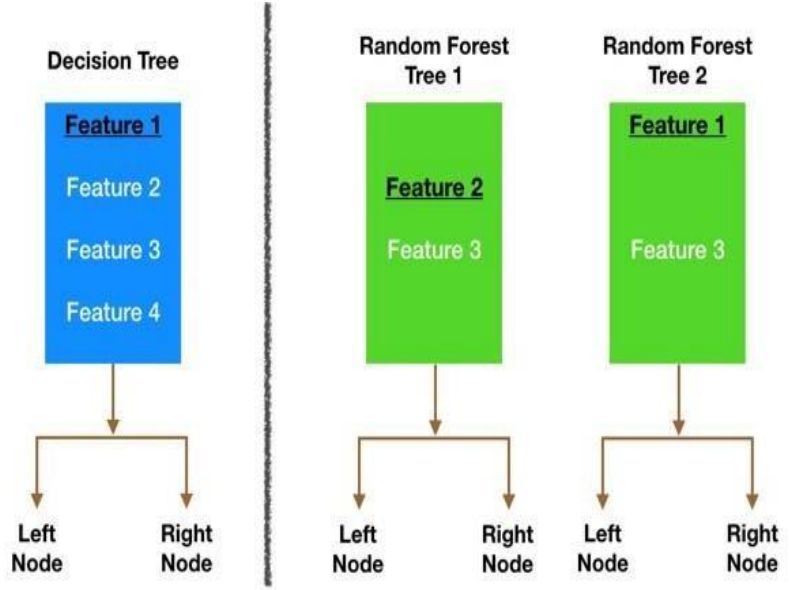
RANDOM FOREST WORKING

- **Feature Randomness**

- In a normal decision tree, we consider the variable with highest gain.

- In contrast, each tree in a random forest can pick only from a random subset of features.

- This forces even more variation amongst the trees in the model and ultimately results in lower correlation across trees and more diversification.



HYPER-PARAMETERS RANDOM FOREST:

- Optimization of RF depends on few inbuilt parameters.
- **n_estimators*** - number of decision trees that the algorithm creates. As the number tree increases, the performance increases and the predictions are more stable but it slows down the computation.
- **max_features*** - maximum number of features that are considered for splitting a node.
- **n_jobs** - number of jobs to run in parallel. If n_jobs=1, it uses one processor. If n_jobs=-1, then the number of jobs is set to the number of cores available.

HYPER-PARAMETERS RANDOM FOREST:

- **max_depth** is the maximum depth of the tree. The deeper the tree, the more splits it has and it captures more information about the data.

- **criterion** is the function to measure the quality of a split. Supported criteria are “**gini**” for the Gini impurity and “**entropy**” for the information gain.

WHY RANDOM FOREST?

- Reduces Risk of Over fitting
- High Accuracy
- Maintain Accuracy even if more data is missing

THAT'S ALL FOLKS