Ensemble Learning

Grey Atom

Loveesh Bhatt

Ensemble Learning

- Combining multiple learners to get better at making predictions
- Typically done using three methodologies
  - Bagging
  - Boosting
  - Stacking
- Helps deal with overfitting (High Variance) and under-fitting (High Bias) problems

Basics of Ensemble models

Let's understand with an example to understand the basics of Ensemble learning

Suppose you want to buy a *XYZ* Car. You aren't sure about its performance though. So, you will look for advice on whether it meets all your expectations? You decide to approach various experts having diverse domain experience:

1.<u>*Sales person of Company XYZ*</u>: This person knows the internal functionality of the car and have the insider information about the functionality and performance of the car. But he lacks a broader perspective on how are competitors innovating, how is the technology evolving and what will be the impact of this evolution on XYZ car. **In the past, he has been right 80% times**.

2.<u>*Online Auto review*</u>: They have a broader perspective on how the performance of cay will fair of in the competitive environment. However, it lacks a view on how the company's internal functionality is faring. **In the past, it has been right 85% times**.

3. Friend who has the same car: This person has observed the car's performance over past 8 months. He is also a knowledgeable in this domain, due to his love for cars. He also has developed a strong intuition on how car might perform over time. **In the past, he has been right 80% times**.

Given the broad spectrum of access we have, we can probably combine all the information and make an informed decision.

Ensemble is the art of combining diverse set of learners (individual models)
together to improvise on the stability and predictive power of the model.
In the above example, the way we combine all the predictions together will be termed as Ensemble Learning.

Why we need Ensemble models

•Decision trees are prone to over fitting, especially when a tree is particularly deep. One way to combat this issue is by setting a max depth. This will limit our risk of over fitting; but at the expense of error due to bias.

•Thus if we set a max depth to a lower number, it would be simpler model with less variance but ultimately will not be a strong predictive model.

•Ideally, we would like to minimize both error due to bias and error due to variance.

Here comes the role of ensemble models.

Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to **decrease variance**, **bias**, or **improve predictions**.

Lets us understand the terms Bias and Variance.

Introduction to Ensemble models

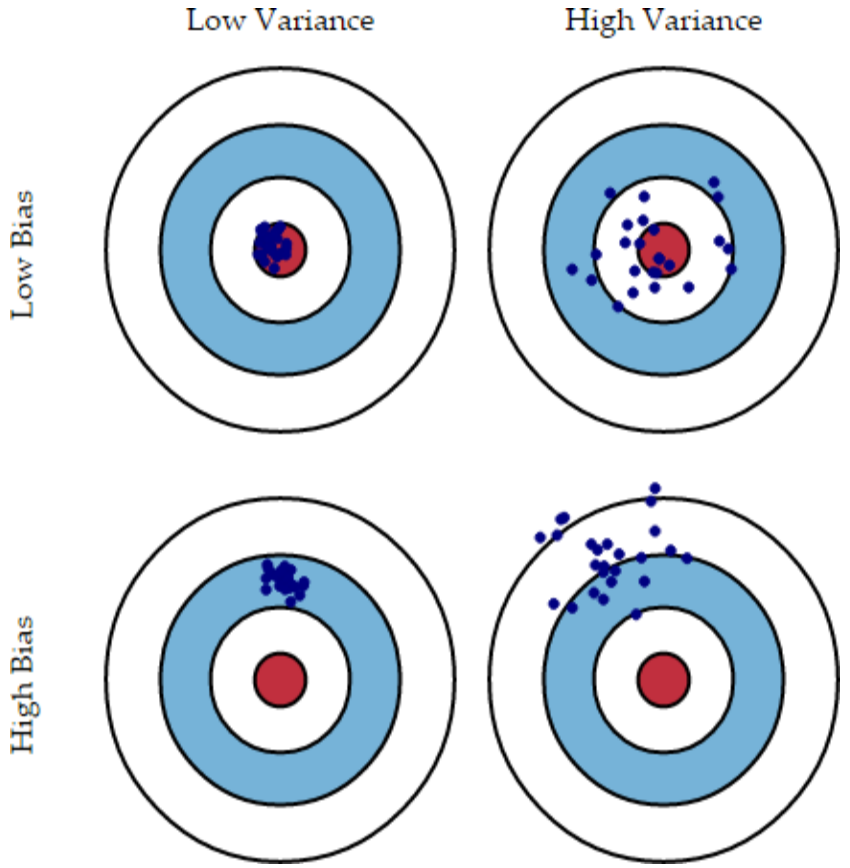
The error emerging from any model can be broken down into three components mathematically.

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

Bias error: is useful to quantify how much on an average are the predicted values different from the actual value. A high bias error means we have a under-performing model which keeps on missing important trends.

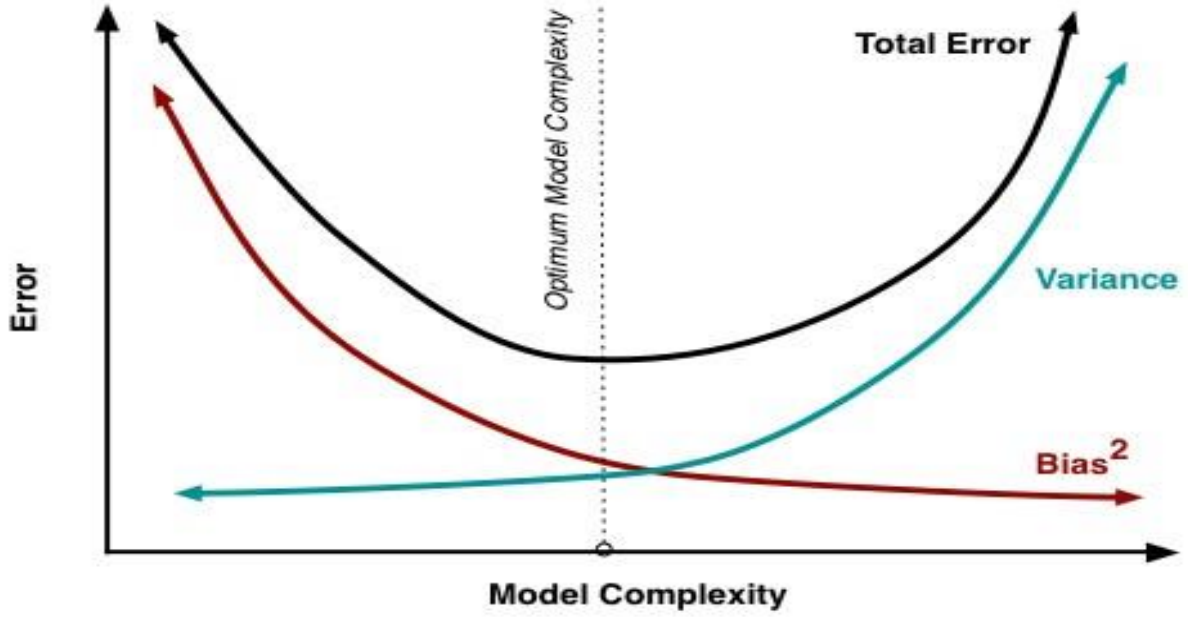
Variance : quantifies how are the prediction made on same observation different from each other. A high variance model will over- fit on your training population and perform badly on any observation beyond training.

In diagram assume that red spot is the real value and blue dots are predictions.



Normally, as the complexity of the model increases, you will see a reduction in error due to lower bias in the model. However, this only happens till a particular point.
As we continue to make the model more complex, we end up over-fitting the model and hence the model will start suffering from high variance.

A champion model should maintain a balance between these two types of errors. This is known as the **trade-off management** of bias-variance errors. Ensemble learning is one way to execute this trade off analysis.

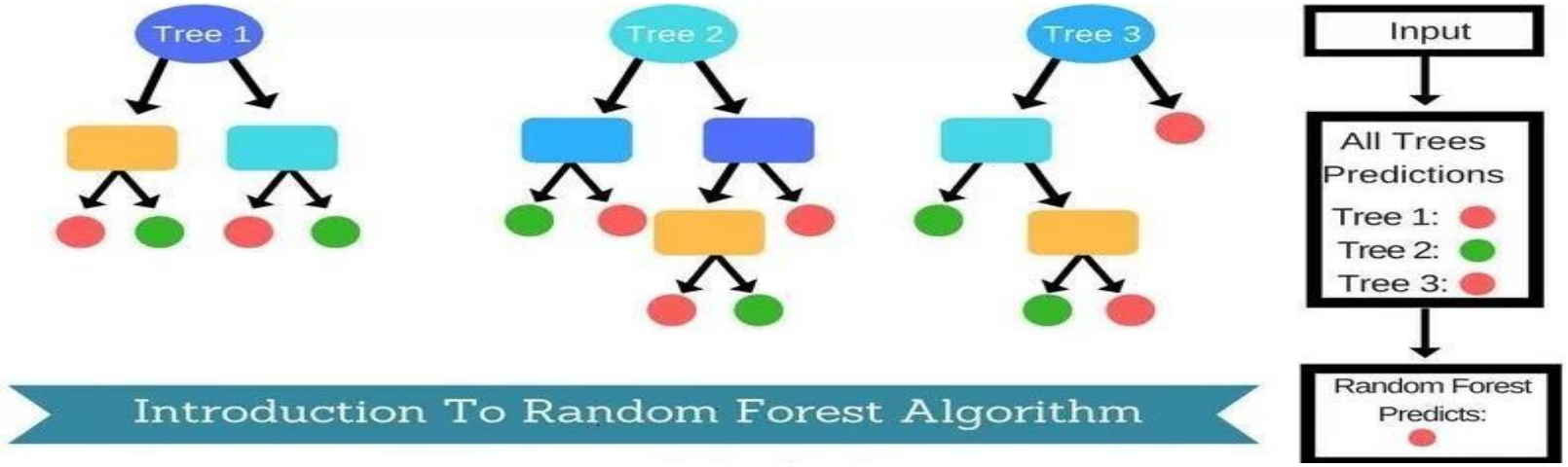


Ensemble Learning Use Case – Random Forest

A commonly used class of ensemble algorithms are forests of randomizedtrees.

Random forest algorithm is a supervised classification algorithm. As the name suggest, this algorithm creates the forest with a number of trees.

Lets understand it better with an example.

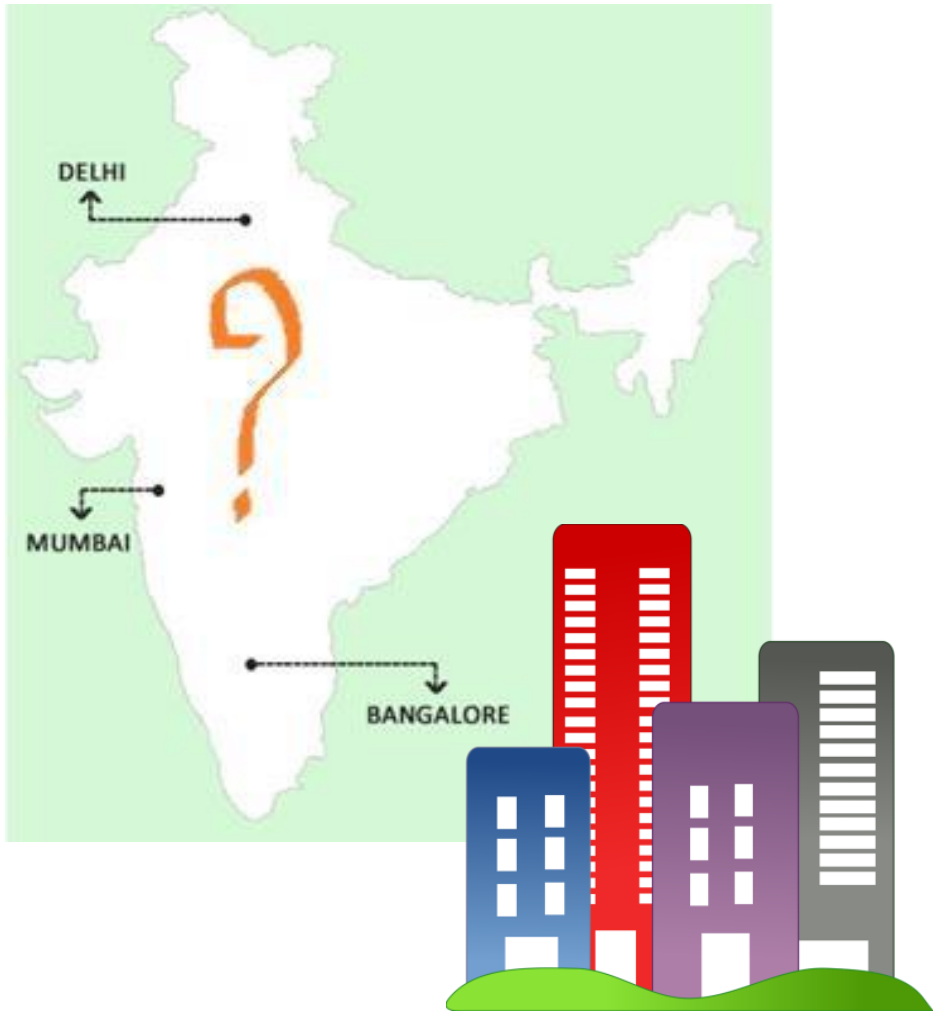


Random forest - Real life example

Suppose **Kiara** decides to buy a house and settle in one of these cities – Mumbai, Delhi or Bangalore. She wants to make a right and optimal choice considering all the perspectives since it is a very important decision for her.

So she decided to ask her **best friend** about the places she may like. Then her friend started asking about her opinions. It's just like
her best friend will ask, You have visited X city. Did you like it?, etc..

Based on the answers which are given by Kiara, her best friend will start recommending the place Kiara may like. Here her best friend forms the decision tree with the answer given by Kiara.



As Kiara’s best friend may recommend HER the best place by virtue of being her friend. The model will be **biased** with the closeness of their friendship. So she decided to ask few more friends to recommend the best place she may like.

Now her friends asked some **random questions** and each one recommended one place to Kiara. Now Kiara considered the place which has **highest votes** from her friends as the final place to settle down.

In the above decision process, two main interesting algorithms decision tree algorithm and random forest algorithm are used.

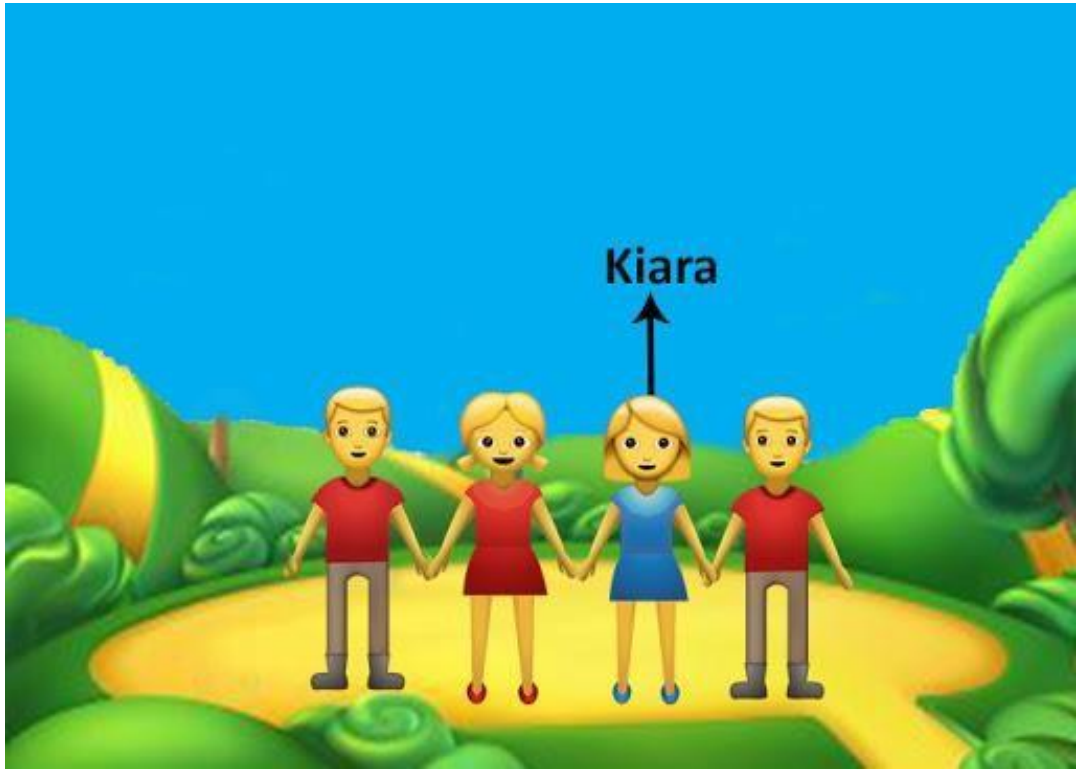
Lets see how…

To recommend the best place to Kiara, her best friend asked some questions. Based on the answers given by Kiara, she recommended a place. This is **decision tree algorithm** approach.

Here Kiara's best friend is the decision tree. The vote (recommended place) is the leaf of the decision tree (Target class). The target is finalized by a single person, In a technical way of saying, using an only single decision tree.

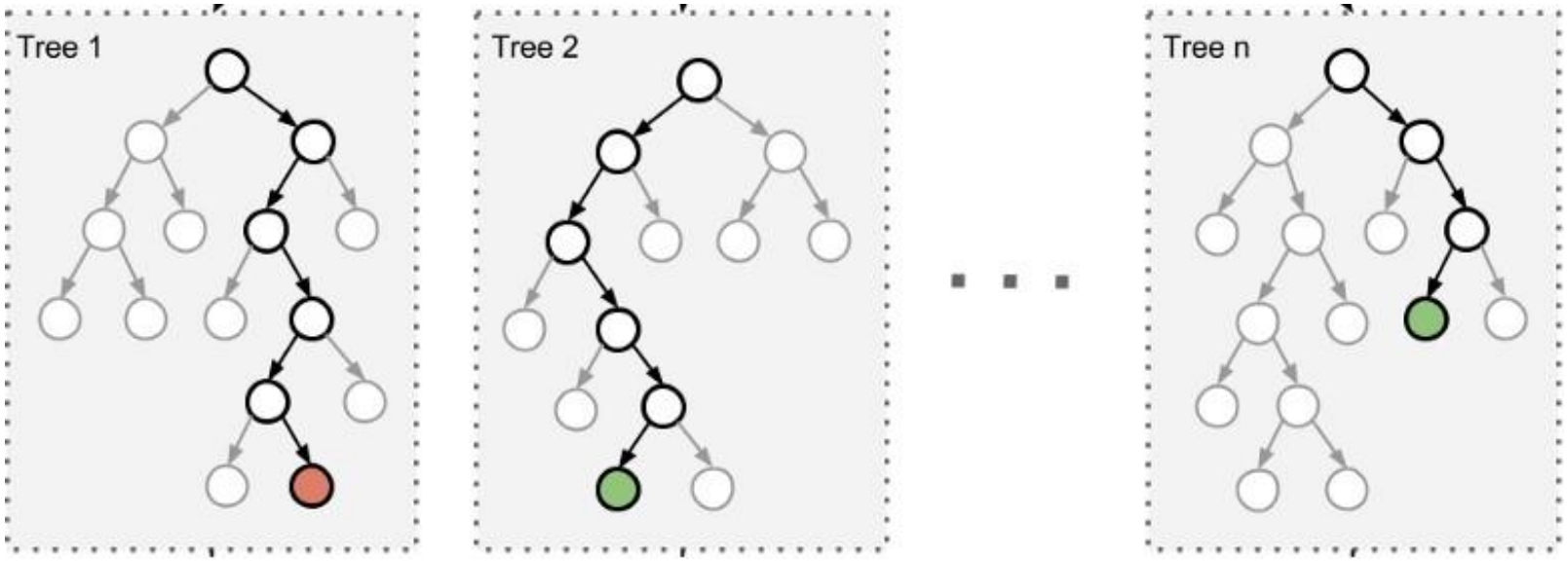
In this case when Kiara asked her friends to recommend the best place to settle among the three. Each friend asked her different questions and came up with their recommendation of a place.
Later Kiara considered all the recommendations and calculated the votes. Votes basically, to pick the popular place from the recommended places from all her friends.
Here, each friend is the tree and the combination of all friends (trees) will form the forest.

This forest is the **random forest**, as each friend asked random questions to recommend the best place to settle down among the three.



In random forests, each tree in the ensemble is built from a sample drawn with replacement (i.e. a bootstrap sample) from the training set. Hence, instead of using all the features, a random subset of features is selected, further randomizing the tree.

As a result, the bias of the forest increases slightly, but due to the averaging of less correlated trees, its variance decreases, resulting in an overall better model.



Random Forest Algorithm - Working

•Assume number of cases in the training set is N. Then, sample of these N cases is taken at random but *with replacement*. This sample will be the training set for growing the tree.

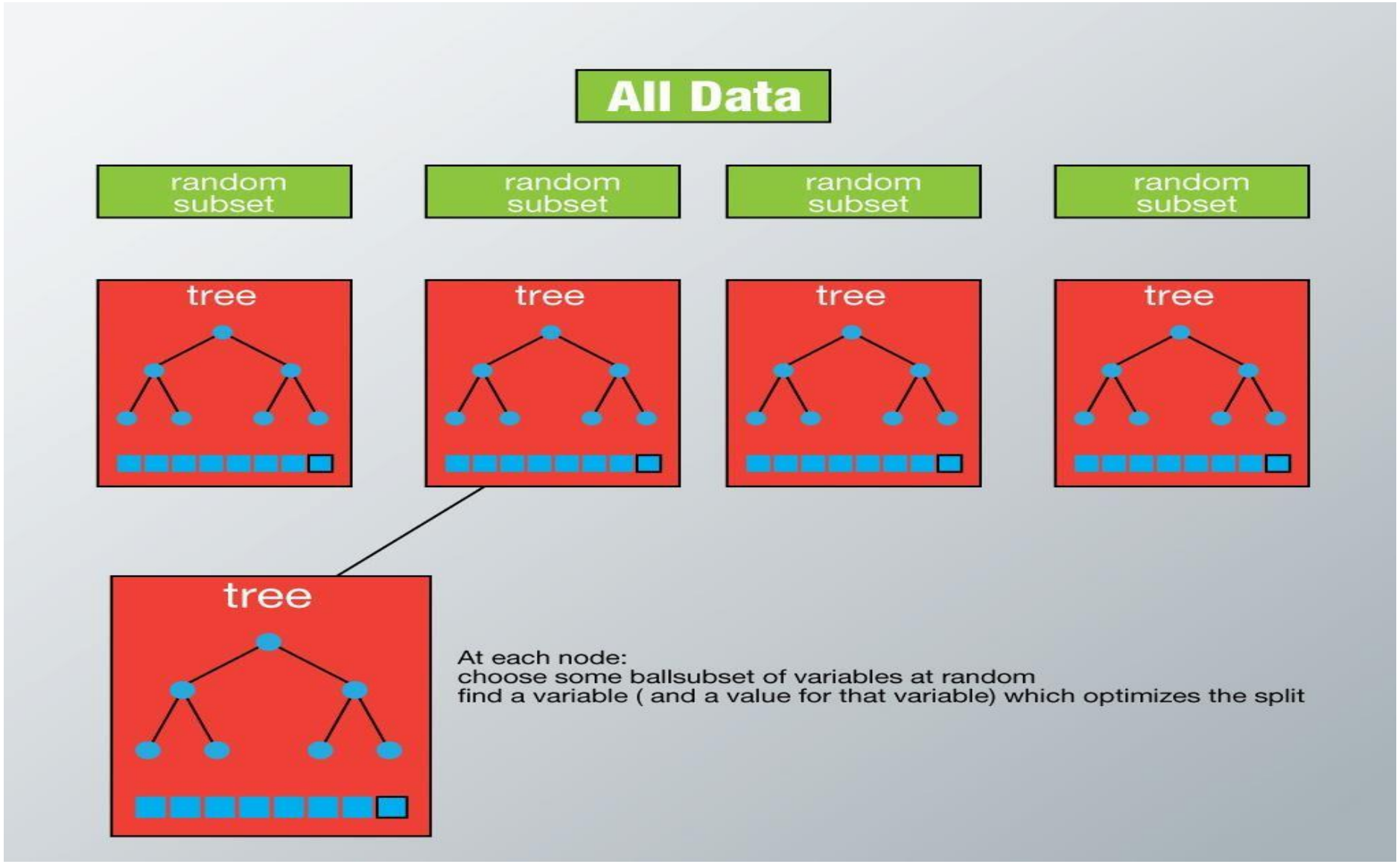
•If there are M input variables, a number m<M is specified such that at each node, m variables are selected at random out of the M. The best split on these m is used to split the node. The value of m is held constant while we grow the forest.

•Each tree is grown to the largest extent possible and there is no pruning.
Predict new data by aggregating the predictions of the ntree trees (i.e., majority votes for classification, average for regression).

Depending upon the value of *m*, there are three slightly different systems:

- ✓Random splitter selection: $m = 1$
- ✓Breiman's bagger: m = total number of predictor variables
- ✓Random forest: m << number of predictor variables. Brieman suggests three possible values for m: $\frac{1}{2}\sqrt{m}$, $\sqrt{m}$, and $2\sqrt{m}$

Visual representation of the working of Random Forest Algorithm:



Advantages of Random Forest

- The same **random forest algorithm** or the random forest classifier can use for both classification and the regression task.

- Random forest classifier will **handle the missing** values.

- It doesn't **overfit** the model.

- Can model the random forest classifier for **categorical values** also.