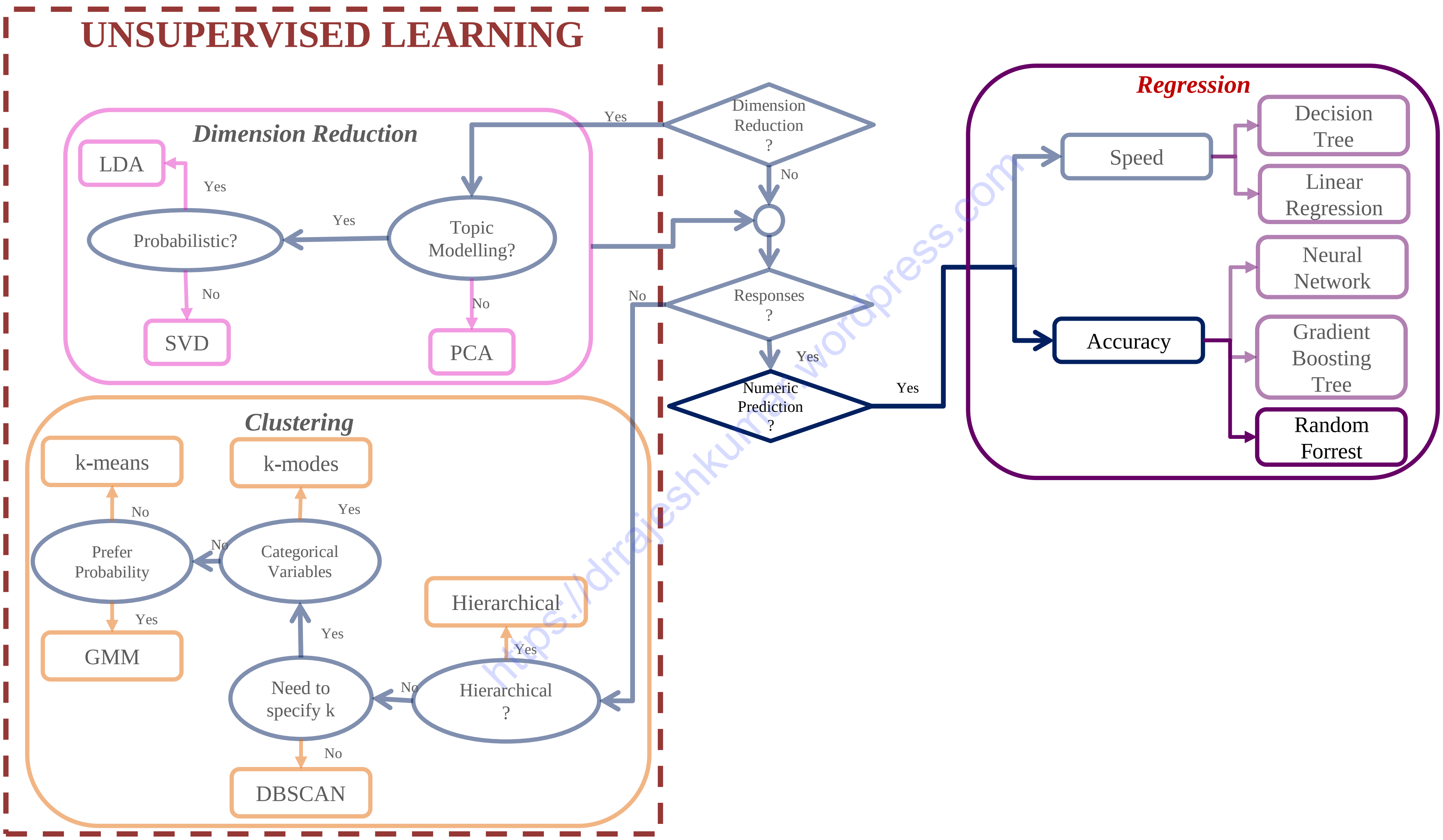
Machine Learning

Dr. Rajesh Kumar

PhD, PDF (NUS, Singapore)
SMIEEE, FIET (UK), FIETE, FIE (I),SMIACSIT, LMISTE, MIAENG
Professor, Department of Electrical Engineering
Professor, Centre of Energy and Environment
Malaviya National Institute of Technology, Jaipur, India, 302017
Tel: (91) 9549654481
http://drrajeshkumar.wordpress.com
rkumar.ee@mnit.ac.in, rkumar.ee@gmail.com


UNSUPERVISED LEARNING
Dimension Reduction
LDA
Yes
Probabilistic?
Yes
Topic Modelling?
No
SVD
No
PCA
Clustering
k-means
k-modes
No
Yes
Prefer Probability
No
Categorical Variables
Yes
GMM
Hierarchical
Yes
Yes
Need to specify k
No
Hierarchical ?
No
DBSCAN
Dimension Reduction ?
Yes
No
Responses ?
No
Yes
Numeric Prediction ?
Yes
Regression
Speed
Decision Tree
Linear Regression
Accuracy
Neural Network
Gradient Boosting Tree
Random Forrest

Random Forest

- Combination of decision trees used to predict the observation
- The decision trees are formed based on any of the following
  - Random selection of features / attributes
  - Random selection of data
- The prediction is based on ensemble method bagging

Random Forest

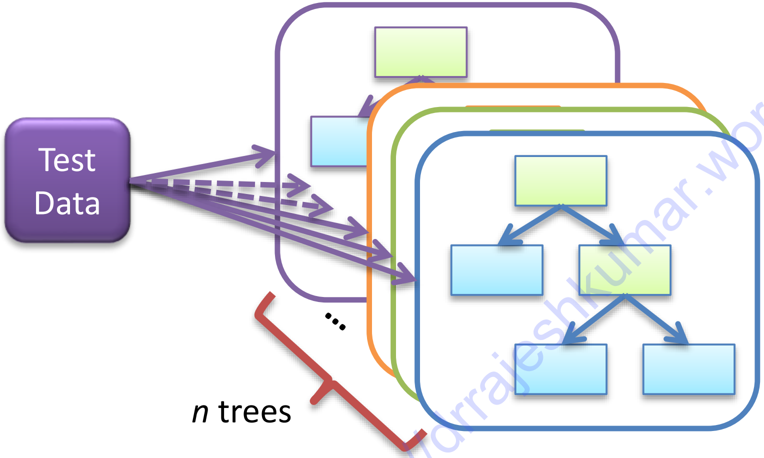
- Combination of decision trees used to predict the observation
- The decision trees are formed based on any of the following
  - Random selection of features / attributes
  - Random selection of data
- The prediction is based on ensemble method bagging

Ensemble methods is a machine learning technique that combines several base models in order to produce prediction

Random Forest

- Combination of decision trees used to predict the observation
- The decision trees are formed based on any of the following
  - Random selection of features / attributes
  - Random selection of data
- The prediction is based on ensemble method bagging
  - Bagging:
    Votes are taken from forest of decision trees and prediction is done based on majority

Random Forest



Step 1: Test data feed to the *n* decision trees

Original Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	Yes	167	Yes

Bootstrapped Dataset

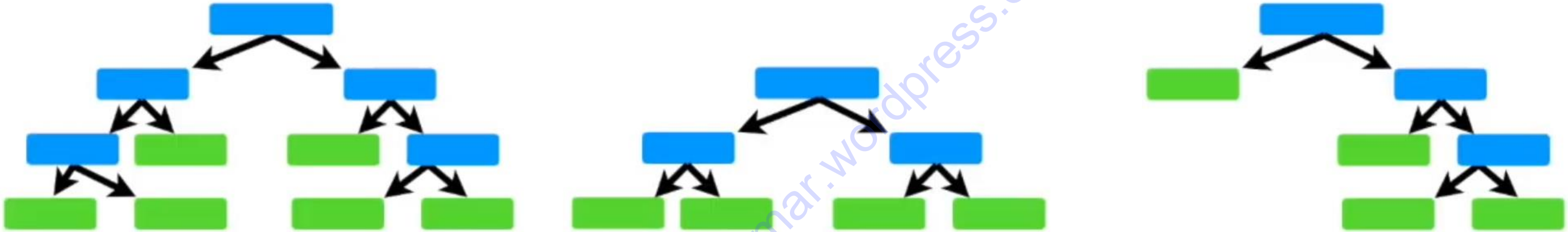
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

Step 2: Create a decision tree using the bootstrapped dataset, but only use a random subset of variables (or columns) at each step.

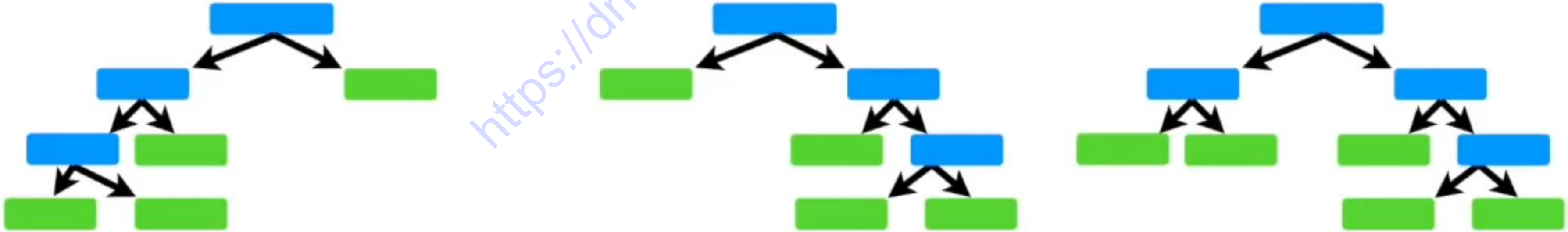
Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

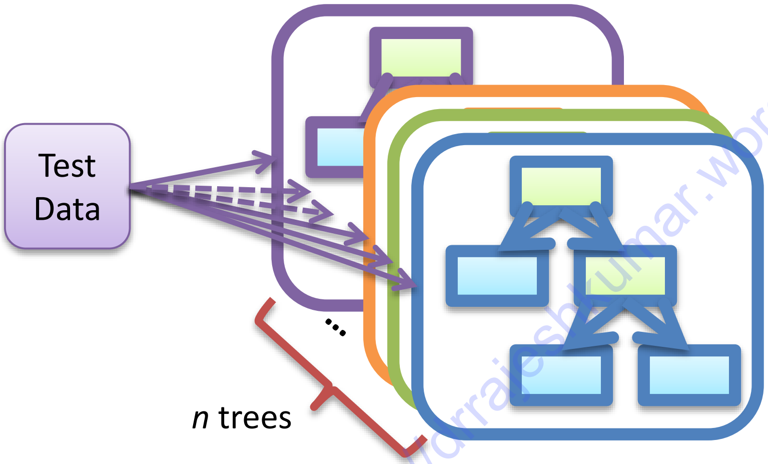
Using a bootstrapped sample and considering only a subset of the variables at each step results in a wide variety of trees.



The variety is what makes random forests more effective than individual decision trees.

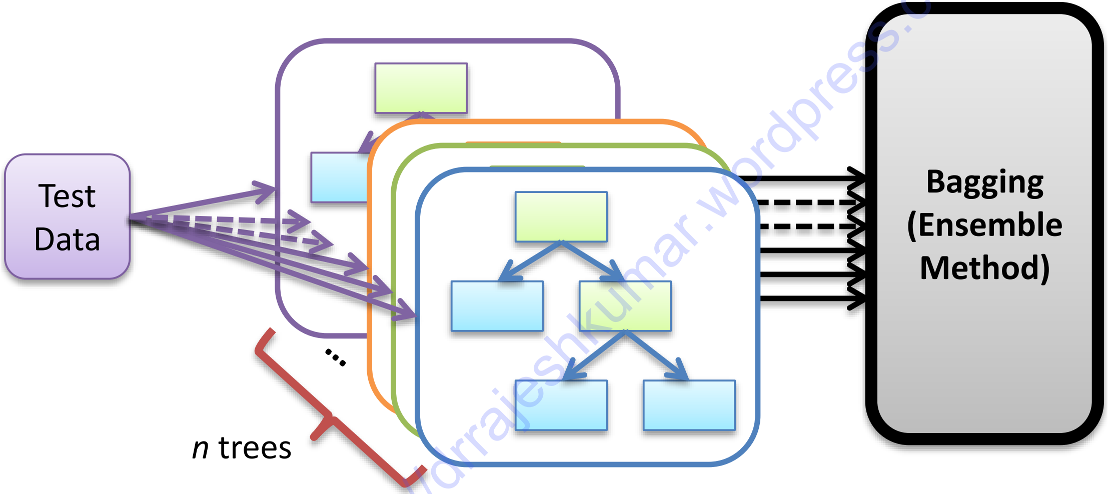


Random Forest



Step 2: Evaluation of individual decision tree starts

Random Forest



Step 3: Individual prediction of decision trees feed to Bagging Block

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	No	168	**YES**

In this case, “**Yes**” received the most votes, so we will conclude that this patient has heart disease.

Heart Disease	
Yes	No
5	1

Random Forests consider 2 types of missing data...

1) Missing data in the original dataset used to create the random forest.

2) Missing data in a new sample that you want to categorize.

Original Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	???	???	No

New Sample

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	???	

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	???	???	No

The general idea for dealing with missing data in this context is to make an initial guess that could be bad, then gradually refine the guess until it is (hopefully) a good guess.

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**No**	???	No

So "**No**" is our initial guess.

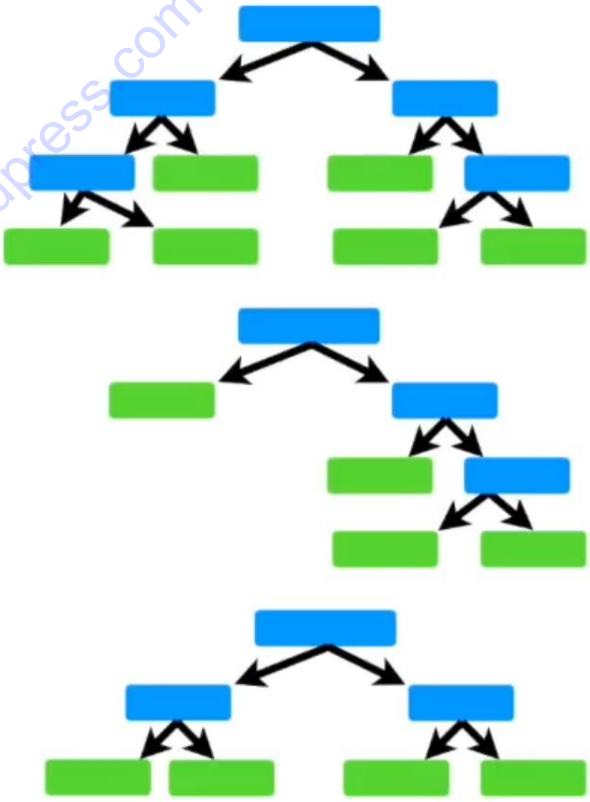
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	No	180	No

In this case, the median value is 180

Filled-in Missing Values

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**No**	**180**	No

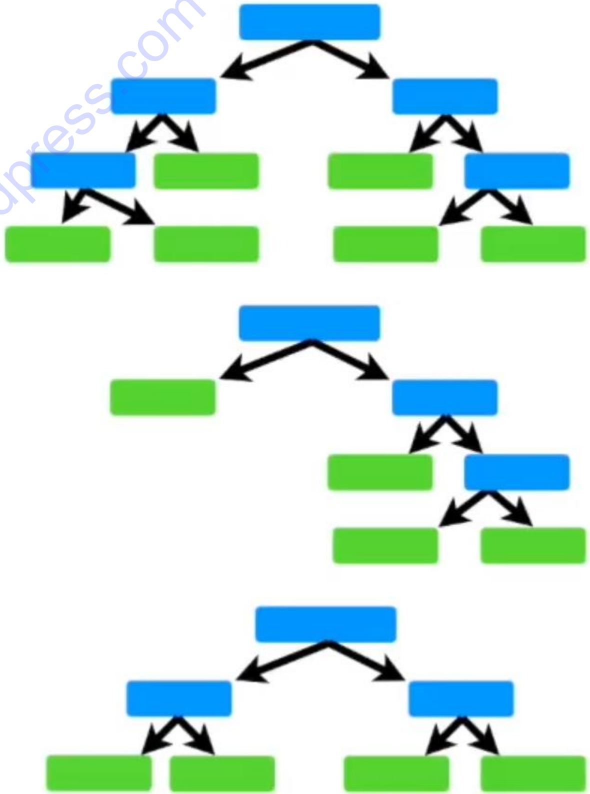
Step 1: Build a random forest...



Filled-in Missing Values

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**No**	**180**	No

Step 2: Run all of the data down all of the trees.



Filled-in Missing Values

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**No**	**180**	No

Because sample 3…

	1	2	3	4
1				
2				
3				
4			1	

…and sample 4 ended up in the same leaf node…

…we put a 1 here.

Filled-in Missing Values

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**No**	**180**	No

	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

Then we divide each proximity value by the total number of trees. In this example, assume we had 10 trees.

Filled-in Missing Values

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**NO**	???	No

The weighted frequency for **"No"** is...

$$\text{Yes} = \frac{1}{3} \times 0.1 = 0.03$$

$$\text{No} = \frac{2}{3} \times 0.9 = 0.6$$

Yes = 1/3

No = 2/3

	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

"No" has a way higher weighted frequency, so we'll go with it.

$$\text{Weighted average} = (125 \times 0.1) + (180 \times 0.1) + (210 \times 0.8)$$

$$= 198.5$$

Filled-in Missing Values

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	NO	???	No

	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

That means…
1 - the proximity values
= distance

	1	2	3	4
1		2	1	1
2	2		1	1
3	1	1		10
4	1	1	10	

	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		1
4	0.1	0.1	1	

	1	2	3	4
1		0.8	0.9	0.9
2	0.8		0.9	0.9
3	0.9	0.9		0
4	0.9	0.9	0	

Gradient Boosting Tree

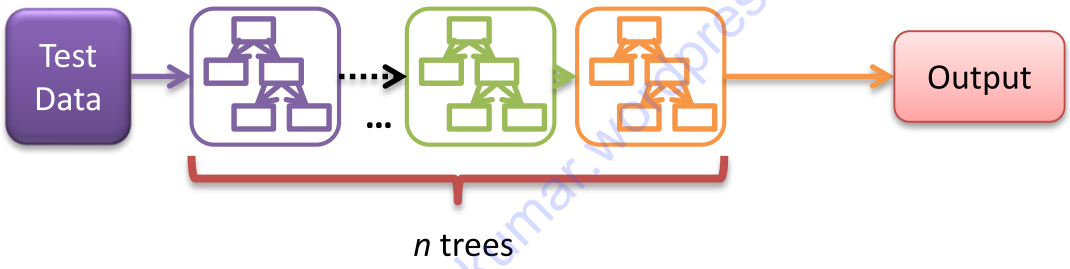
- Type of sequential ensemble method – Boosting
- Group of decision trees connected sequentially
- Boosting algorithms converts weak learners to strong learner
  - Objective is to reduce error in prediction (MSE generally used)

$$MSE = \sum_{i=1}^{n} \left(y_i - y_i^p \right)^2$$

  where, y_i is i^{th} target value, $y^p{}_i$ is i^{th} prediction

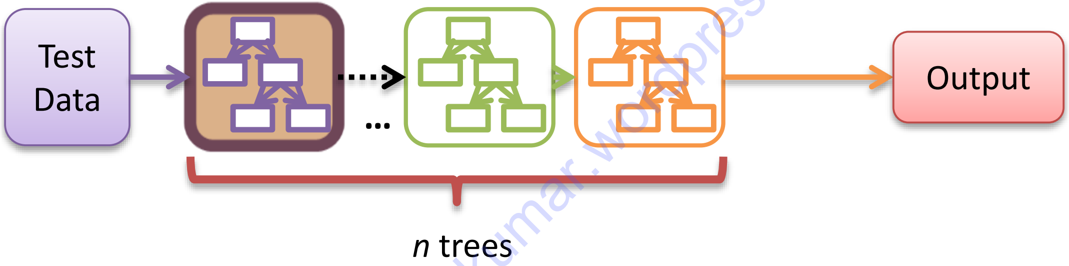
  - The base learners (decision trees) are generated sequentially
  - The base learner uses gradient descent and update prediction based on a learning rate
  - The next learner is feed with weighted previously mislabelled instances to enhance overall performance

Gradient Boosting Tree



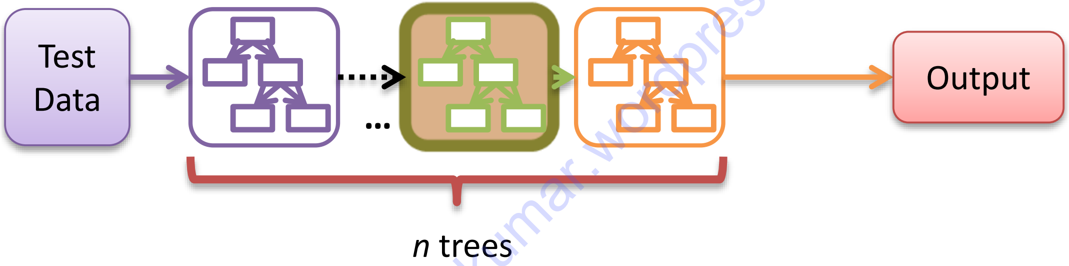
Step 1: Instance of test data is feed to the first decision tree

Gradient Boosting Tree



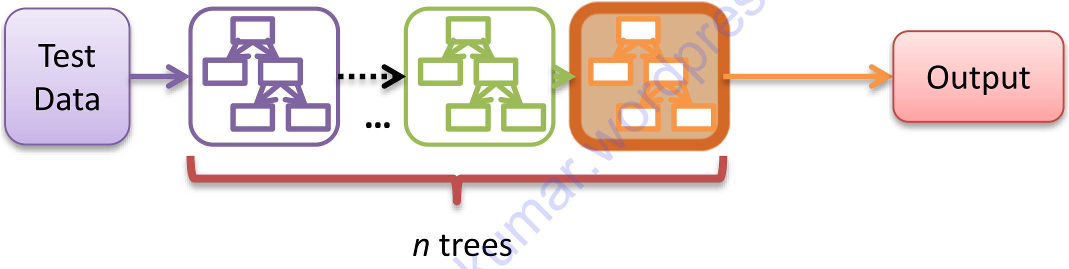
Step 2a: Evaluation of first decision trees starts and feed prediction to second decision tree

Gradient Boosting Tree



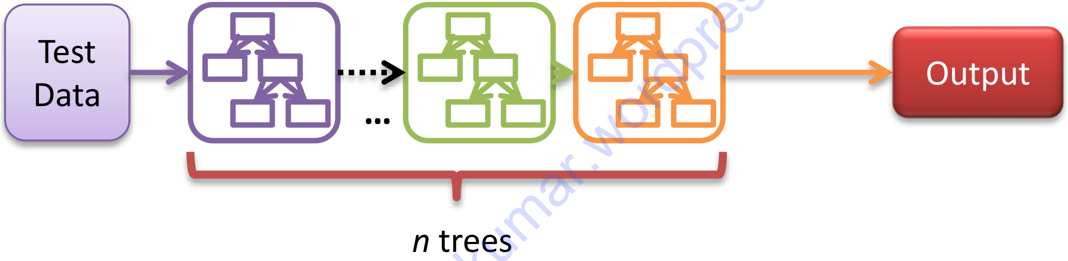
Step 2.(n-1): Sequentially the prediction of the previous decision tree feed to the next decision tree

Gradient Boosting Tree



Step 2.n: Sequentially evaluation reaches to the last decision tree

Gradient Boosting Tree



Step 3: Prediction is obtained as output from the last decision tree