University of Regina

ENEL/ENSE 865: Applied Machine Learning

Professor: Dr. Abdul Bais

Instructional Designer/TA: Dr. Muhammad Hamza Asad

Submitted by:

Name: Nandish Bakulkumar Bhatt

Student ID: 200441204

Writing Assignment:4

Ans: 1

Name: Nandish. Bakulkumar. Bhatt.
Student ID: 200441204

→ Before reaching to conclusion of question that with each iteration, the objective of K-Means clustering decreases. First, we will understand K-Means Clustering.

→ K-Means Clustering is a method which actually partition "n"-observations into "k"-clusters in which each ~~cluster~~ observation belongs to the cluster with nearest mean. This results in partion of data set into Vornoi cells.

→ Now, in K-Means clustering Algorithm, let -

$C_1, C_2, \ldots C_K$ → set containing indices of observations in each cluster.

where, $C_1 \cup C_2 \cup C_3 \cup \ldots \cup C_K = \{1, \ldots n\}$ and

$C_K \cap C_{K'} = \phi$ for $K \neq K'$

→ So, by forming such clusters, let's say if the i^{th} observation is in k^{th} cluster, then we can say, $i \in C_K$.

→ The main function of K-means clustering is that a best clustering is one for which the cluster variation between themselves is a small as possible.

→ Now,

The within-cluster variation for cluster C_K is a measure ~~variation~~ of $W(C_K)$.

So, if taking squared Euclidean distance, then -

$$W(C_K) = \frac{1}{|C_K|} \sum_{i,i' \in C_K} \sum_{j=1}^{P} (x_{ij} - x_{i'j})^2 \quad \text{---------- (1)}$$

where, $|C_k| \longrightarrow$ No. of observations in k^{th} cluster.

So, $W(C_k) =$ Sum of all pairwise squared Euclidean distances between the observations in k^{th} cluster / Total no. of observation in k^{th} cluster

→ Now, for K-Means clustering, we need to ~~minimize~~ partition the observations into K-clusters such that total within-cluster variation, summed over all K-clusters is as small as possible. i.e.,

$$\underset{C_1, C_2, \ldots C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\} \quad \text{---------- (2)}$$

→ Thus, by combining equations (1) and (2), we get —

Optimization problem that defines K-Means Clustering $=$

$$\underset{C_1, C_2, \ldots C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\} \quad \text{---(3)}$$

This is actually a measure of quality of given clustering. (lower is better)

→ Now, to minimize the above objective is very difficult problem, as there are almost K^n ways to partition n-observations into K-clusters. But, still it can be solved/minimized using the following algorithm:—

Steps:-

(1) Randomly / Arbitrarily assigns centers from all observations to each of cluster 1, 2, 3, ... k.

(2) Iterate until the clusters centers is not changing.

(a) For each of the K - clusters, calculate the cluster centroid.

(The k^{th} cluster centroid = Vector of p- feature means for the observations in the k^{th} cluster)

(b) Assign each observation to the cluster whose centroid is closest.

→ So, we can say that above algorithm is decreasing the value of objective at each step with high confidence.

→ Thus, $\sum_{j=1}^{K} \sum_{i: z_i = j} \| \mu_j - x_i \|_2^2$ where, $\mu_j = \frac{1}{n_j} \sum_{i: z_i = j} x_i$ ——— (4)

is decreasing.

→ Also, $\frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$ ——— (5)

where, $\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$ = Mean for feature -j in cluster C_k.

→ In Step-2(a), the cluster means for each feature are constants that minimize the sum-of square deviations, and in step-2(b), reallocating the observations can only improve eq^n(4). So, as the algorithm starting running, the clustering obtained will continually improve until the result no longer changes and objective of eq^n (3) will never increase.

→ So, when result is not changing, then a local optimum has been reached.

→ Also, K-means algorithm finds a local optimum rather than global optimum as the results obtained depends on initial (random) cluster assignment of each observation in step (1) of algorithm. So, for this reason, it is crucial to run algorithm several times from different initial configuration. Then, select the best solution, in which the objective is smallest.

Reference: An Introduction to Statistical Learning with Applications in R by James, Witten, Hastie and Tibshirani.

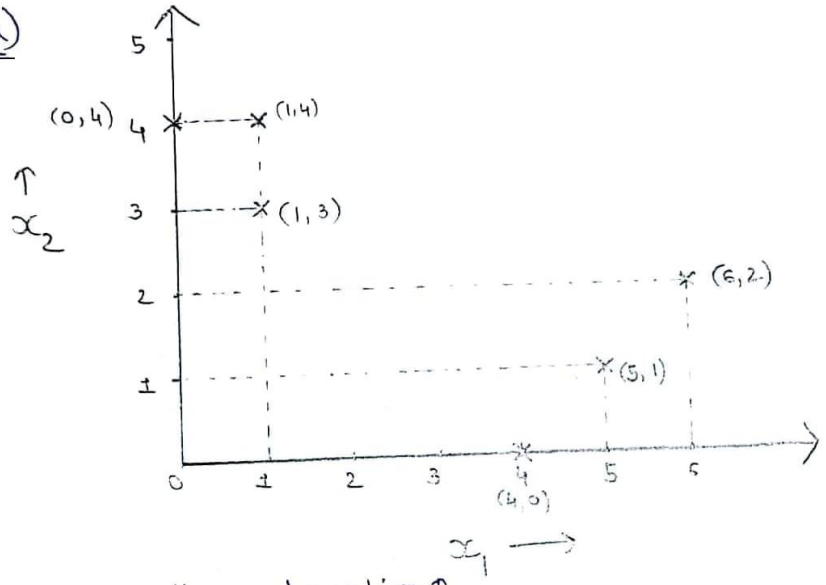
Ans: 2 Given that, $K = 2$. = Number of clusters

No. of observations, $n = 6$

No. of features, $p = 2$.

Observation	x_1	x_2	
1	1	4	$P_1(1,4)$
2	1	3	$P_2(1,3)$
3	0	4	$P_3(0,4)$
4	5	1	$P_4(5,1)$
5	6	2	$P_5(6,2)$
6	4	0	$P_6(4,0)$

2(a)



Plotting the observations

2(b) K-Means Clustering:-

→ Let the initial cluster centers are:-

$\mu_1 = (1,4)$ and $\mu_2 = (5,1)$.

(∵ It is observed from figure, that (1,4) (0,4) and (1,3) are in one group and (4,0)(5,1)(6,2) are forming another group, so taking (1,4) as center of one and (5,1) as center of another)

→ Thus, calculating the distance of all observations with respect to these two clusters.

→ The distance is calculated using —

$$d(x_i, x_q) = \sqrt{(x_i[1] - x_q[1])^2 + (x_i[2] - x_q[2])^2 + \ldots + (x_i[d] - x_q[d])^2}$$

→ Thus, Observations:

1st observation: (1, 4)

- Distance of (1,4) from $\mu_1(1, 4) = \sqrt{(1-1)^2 + (4-4)^2} = 0$.
- Distance of (1,4) from $\mu_2(5, 1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$

→ 2nd observation: (1, 3).

Distance of (1,3) from $\mu_1(1, 4) = \sqrt{(1-1)^2 + (4-3)^2} = 1$

Distance of (1,3) from $\mu_2(5, 1) = \sqrt{(1-5)^2 + (3-1)^2} = 4.47$

→ 3rd observation: (0, 4).

Distance of (0,4) from $\mu_1(1, 4) = \sqrt{(0-1)^2 + (4-4)^2} = 1$

Distance of (0,4) from $\mu_2(5, 1) = \sqrt{(0-5)^2 + (4-1)^2} = 5.83$

→ 4th observation: (5, 1)

Distance of (5,1) from $\mu_1(1, 4) = \sqrt{(5-1)^2 + (1-4)^2} = 5$

Distance of (5,1) from $\mu_2(5, 1) = \sqrt{(5-5)^2 + (1-1)^2} = 0$

→ 5th observation: (6, 2)

Distance of (6,2) from $\mu_1(1, 4) = \sqrt{(6-1)^2 + (2-4)^2} = 5.38$

Distance of (6,2) from $\mu_2(5,1) = \sqrt{(6-5)^2+(2-1)^2} = 1.41$

→ 6th observation: (4,0)

Distance of (4,0) from $\mu_1(1,4) = \sqrt{(4-1)^2+(0-4)^2} = 5$

Distance of (4,0) from $\mu_2(5,1) = \sqrt{(4-5)^2+(0-1)^2} = 1.41$

Observation	(1,4) Distance of cluster-1's center w.r.t. folbwing obsn.	(5,1) Distance of cluster-2's center w.r.t. following obsn	Observation assigned to the cluster no.
(1,4)	0	5	1 (μ_1)
(1,3)	1	4.47	1 (μ_1)
(0,4)	1	5.83	1 (μ_1)
(5,1)	5	0	2 (μ_2)
(6,2)	5.38	1.41	2 (μ_2)
(4,0)	5	1.41	2 (μ_2)

→ So, the new clusters are :—

Cluster:-1	→ (1,4), (1,3), (0,4)
Cluster:-2	→ (5,1), (6,2), (4,0)

→ So, the cluster centers of new clusters are,

μ_1 = center of cluster-1 = Mean of assigned observations

$= \left(\frac{1+1+0}{3}, \frac{4+3+4}{3}\right)$

$\therefore \mu_1 = (0.67, 3.67)$

$\therefore \mu_2$ = center of cluster-2 $= \left(\frac{5+6+4}{3}, \frac{1+2+0}{3}\right) = (5, 1)$

→ Thus, calculating distance of all observations with respect to these two new clusters —

Observation: 1 (1, 4)

Distance of (1, 4) from μ_1 $\begin{matrix}(0.67,\\ 3.67\end{matrix}$) $= \sqrt{(1-0.67)^2+(4-3.67)^2} = 0.467$

Distance of (1, 4) from $\mu_2(5, 1) = \sqrt{(1-5)^2+(4-1)^2} = 5$

→ 2nd observation: (1, 3)

Distance of (1, 3) from $\mu_1(0.67, 3.67) = \sqrt{(1-0.67)^2+(3-3.67)^2} = 0.75$

Distance of (1, 3) from $\mu_2(5, 1) = \sqrt{(1-5)^2+(3-1)^2} = 4.47$

→ 3rd observation (0, 4)

Distance of (0, 4) from $\mu_1(0.67, 3.67) = \sqrt{(0-0.67)^2+(4-3.67)^2} = 0.74$

Distance of (0, 4) from $\mu_2(5, 1) = \sqrt{(0-5)^2+(4-1)^2} = 5.83$

→ 4th observation (5, 1)

Distance of (5, 1) from $\mu_1(0.67, 3.67) = \sqrt{(5-0.67)^2+(1-3.67)^2} = 5.08$

Distance of (5, 1) from $\mu_2(5, 1) = \sqrt{(5-5)^2+(1-1)^2} = 0$

→ 5th observation (6, 2)

Distance of (6, 2) from $\mu_1(0.67, 3.67) = \sqrt{(6-0.67)^2+(2-3.67)^2} = 5.58$

Distance of (6, 2) from $\mu_2(5, 1) = \sqrt{(6-5)^2+(2-1)^2} = 1.41$

→ 6th observation (4, 0)

Distance of (4, 0) from $\mu_1(0.67, 3.67) = \sqrt{(4-0.67)^2+(0-3.67)^2} = 4.95$

Distance of (4, 0) from $\mu_2(5, 1) = \sqrt{(4-5)^2+(0-1)^2} = 1.41$

→ So, combining all this data into table —

→ Now, calculating the distance w.r.t. the observations assigned to new cluster groups.

Observation	Distance of cluster-1's center (0.67, 3.67) w.r.t. following obsn.	Distance of cluster-2's center (5, 1) w.r.t. following obsn.	Observation assigned to the closest cluster center no.
(1, 4)	0.467	5	1 (μ_1)
(1, 3)	0.75	4.47	1 (μ_1)
(0, 4)	0.74	5.83	1 (μ_1)
(5, 1)	5.08	0	2 (μ_2)
(6, 2)	5.58	1.41	2 (μ_2)
(4, 0)	4.95	1.41	2 (μ_2)

→ So, new clusters groups are:—

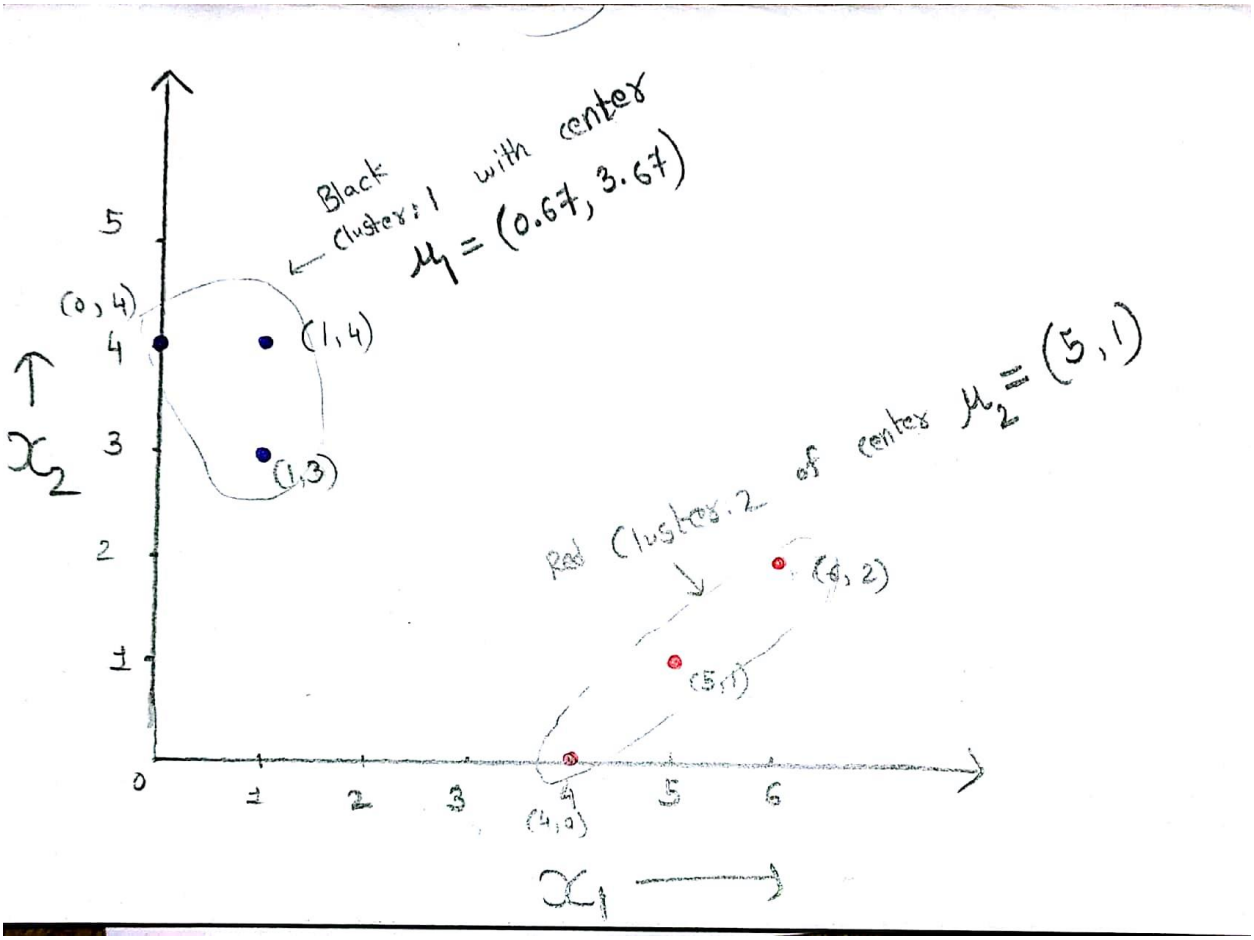
Cluster:-1	(1,4), (1,3), (0,4)
Cluster-2	(5,1), (6,2), (4,0)

→ So, the cluster of new clusters are same only. i.e.,

μ_1 = center of cluster-1 = (0.67, 3.67)

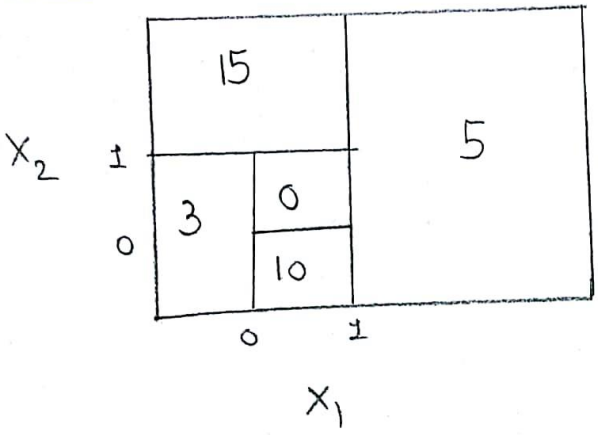
μ_2 = center of cluster-2 = (5, 1).

→ Thus, the cluster centers stop changing, so we can say that (1,4), (1,3), (0,4) forms one cluster and (5,1), (6,2), (4,0) forms another cluster.


Black
← cluster: 1 with center
$\mu_1 = (0.67, 3.67)$
(0,4)
(1,4)
(1,3)
x_2
5
4
3
2
1
0
1
2
3
4
5
6
(4,0)
x_1
Red (cluster 2 of center $\mu_2 = (5,1)$
(6,2)
(5,1)

Ans: 3

3(a) Now, the data given is :-



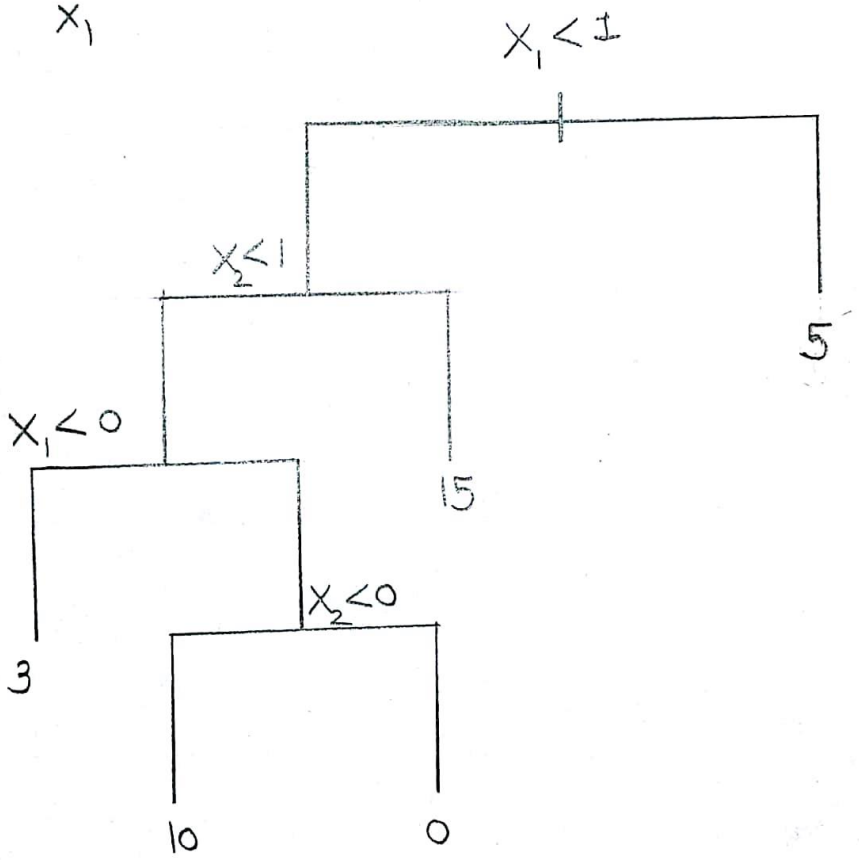
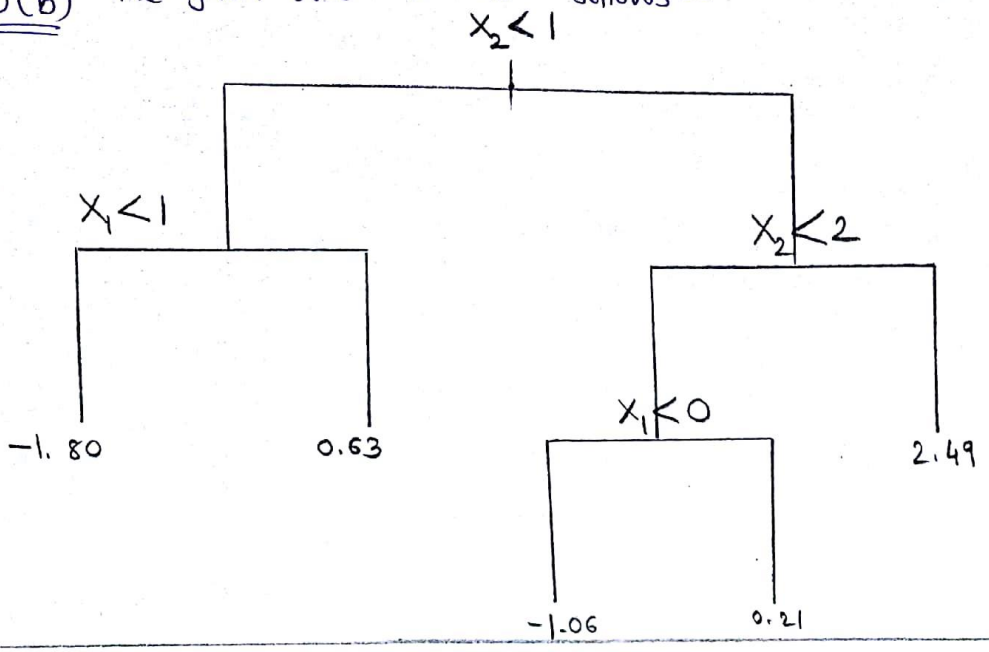
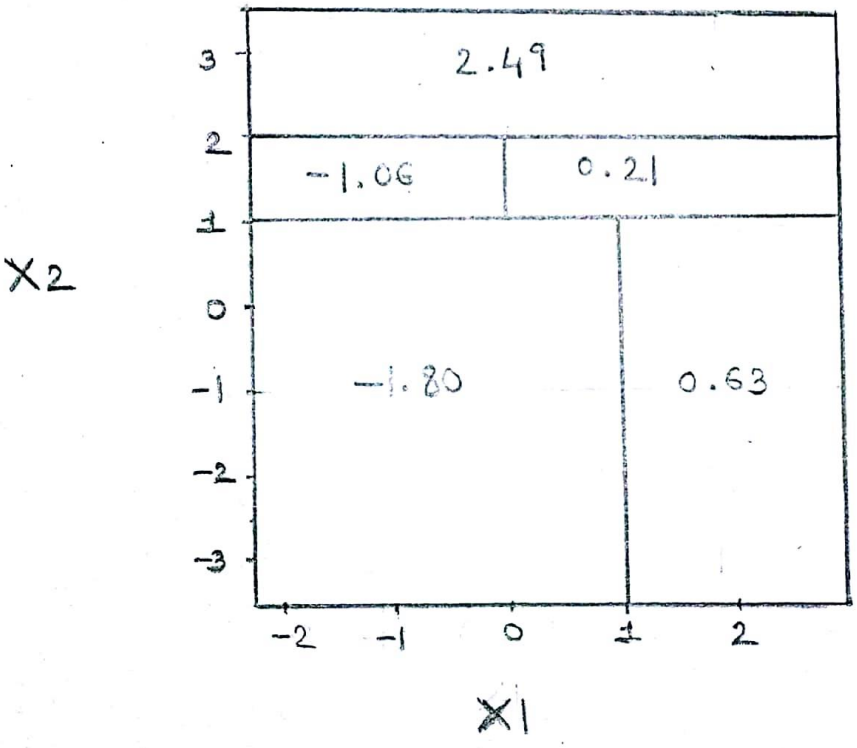


Fig - 3(a)

3(b) The given tree is as follows –





Ans: 4 Before going to understand the similarity and differences of Bagging, Boosting and Random Forest, let's first understand the Ensemble Method.

Ensemble Methods :-

→ Ensemble Method is actually a machine learning model where multiple models (or weak learners) are trained to solve the same problem and combined together to get best results.

→ So, by doing this, we get robust model by combining all of the weak learning models.

Bagging :-

→ Bagging (Bootstrap aggregating) often considers homogenous weak learners, learns independently from each other in parallel and combines them following. with some kind of deterministic averaging process.

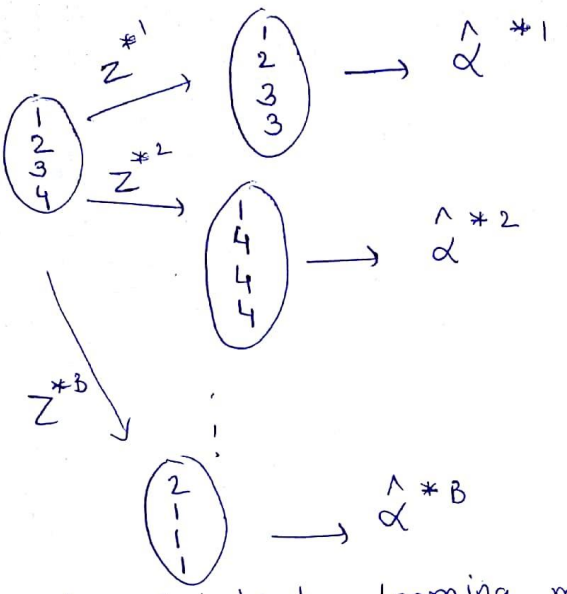
→ Bagging will getting an ensemble model with less variance than its component.

Algorithm:

→ In this, we create first multiple bootstrap samples, so that each new bootstrap sample will act as another almost independent dataset drawn from true observation/distribution.

→ Then, we will fit a weak learner for each of these samples and finally aggregate them such

that we "average" of outputs, and thus we obtain an ensemble ~~method~~ model with less variance than its components



→ Thus, train the statistical learning method on each of the B-bootstrapped training datasets and obtain B-predictions.

For prediction, if problem is of
- Regression: Avg. all B-predictions from all B-trees
- Classification: Majority vote among all B-trees.

→ Advantages:
(a) we get robust model with low variance.

→ Disadvantages:
- Individual learners are highly co-related among themselves.
- Highly computation is req. i.e, computationally expensive.

Boosting :-

→ Boosting Algorithm also works similar to bagging, but the difference is it learns them sequentially in a very adaptive way (a base model depends on the previous ones) and combines them following a deterministic strategy.

→ In this, Ada Boost (Adaptive Boost) Algorithm is used which is as follows —

- (i) Start with same weight for all data points, $\alpha_i = \frac{1}{N}$
- (ii) For $t = 1, \ldots T$.
  - ↳ Learn $f_t(x)$ with data weights α_i.
  - ↳ Compute coefficient $\hat{w}_t$ where, $\hat{w}_t = \frac{1}{2} \ln \left(\frac{1 - \text{weighted\_error}[f_t]}{\text{weighted\_error}[f_t]} \right)$
  - ↳ Recompute weights α_i by $\alpha_i \leftarrow \begin{cases} \alpha_i e^{-\hat{w}_t} & \text{, if } \hat{y} = y \\ \alpha_i e^{\hat{w}_t} & \text{, if } \hat{y} \neq y \end{cases}$
  - ↳ Normalize weights α_i by $\alpha_i \leftarrow \frac{\alpha_i}{\sum_{j=1}^{N} \alpha_i}$

(iii) Final model prediction by —

$$\hat{y} = \text{sign} \left(\sum_{t=1}^{T} \hat{w}_t f_t(x) \right)$$

Advantages:

- Computationally less costly as compared to bagging.
- Prediction capability is high.

Disadvantages:

- Hard to scale up, as it depends on previous predictors
- Too dependent on outliers, so highly sensitive to outliers.

Random Forest :-

→ Random Forest is bagging method only, but has deep trees, fitted on bootstrap samples, are combined to produce an output with lower variance.

~~→ In this, the individual learners are~~

→ In addition to taking the random subset of data, it also takes the random selection of features rather than using all the features to grow trees.

→ Algorithm of random forest :-

(i) If there are N = observation and M features in training data set, then first a sample from training data is taken randomly with replacement.

(ii) A subset of M features are selected randomly and whichever feature gives the best split is used to split the node iteratively.

(iii) Tree is grown to largest.

(iv) Above steps are repeated & prediction is given based on aggregation of predictions from n number of trees.

→ Advantages :

- Handles missing values & maintains accuracy.
- Handles higher dimensional data.

→ Disadvantages :-

- Not giving precise values. for regression model., as final prediction is based on mean predictions.

* Similarity between Bagging and Boosting :-

(1) Both are ensemble methods to obtain N learners from a single learner.

(2) Bagging and boosting make random sampling and generate several training datasets.

(3) Bagging and Boosting arrive/make final decisions by - (a) Taking mean of N ~~observations~~/ learners
(b) Taking majority or voting.

(4) Both are good at reducing variance and give higher stability with minimizing errors.

* Difference b/w Bagging and Random Forest :-

Bagging	Random Forest
(1) Bagging is one of simplest ensemble based algorithms to enhance predictive accuracy.	(1) Random Forest on other hand is improved version of bagging, which is essentially an ensemble of decision trees training with bagging mechanism.
(2) Here, it is to train a bunch of unpruned decision trees built on different random subsets of training data, sampling with replacement.	(2) To build multiple decision trees and aggregate them to get an accurate result.
(3) Reduces overfitting.	(3) Reduces overfitting and good with imbalanced and missing data.

Boosting	Bagging	Random Forest
(1) Boosting considers homogenous weak learners, learns them sequentially in very adaptive way and combines them following a deterministic strategy.	(1) Bagging considers homogenous weak learners, learns them independently from each other in parallel and combines them following some kind of deterministic averaging process.	(1) Random Forest is same as bagging, but in addition to taking random subset of data, it also takes random selection of features rather than using all features to grow trees
(2) Individual trees have high bias and low variance.	(2) Individual trees have low bias and high variance.	(2) Individual trees have low bias and high variance.
(3) Individual trees are dependent on each other.	(3) Individual trees are independent on each other.	(3) Individual trees are independent on each other.
(4) Computationally less costly as compared to bagging.	(4) Computationally more costly as compared to boosting.	(4) Computationally more costly.
(5) Boosting tries to reduce bias. and not variance.	(5) Bagging tries to solve over-fitting problem. and reduces variance; not bias.	(5) Random Forest tries to solve problems of unbalanced and missing data.
(6) Models are weighted according to their performance.	(6) Each model receives an equal weight.	(6) It is same as bagging, but selection of features occurs.
(7) If classifier is stable and simpler (i.e. high bias), then apply boosting.	(7) If classifier is unstable (high variance), then apply bagging.	(7) If classifier is unstable and feature selection is needed, then apply random forest.
e.g, Gradient Boosting	e.g, Random Forest	