DECISION TREES AND RANDOM FOREST Q&A

1.What is a Decision Tree?

A decision tree is a tree in which every node specifies a test of some attribute of the data and each branch descending from that node corresponds to one of the possible values for this attribute.

2.To which kind of problems are decision trees most suitable?

Decision trees are most suitable for tabular data.

The outputs are discrete.

Explanations for decisions are required.

The training data may contain errors.

The training data may contain missing attribute values.

3.On what basis is an attribute selected in the decision tree for choosing it as a node?

Attribute selection is done using Information Gain in decision trees. The attribute with maximum information gain is selected.

4.What is Information Gain? What are its disadvantages?

Information gain is the reduction in entropy due to the selection of an attribute. Information gain ratio biases the decision tree AGAINST considering attributes with a large number of distinct values which might lead to overfitting. In order to solve this problem, information gain ratio is used.

5.What is the inductive bias of decision trees?

Shorter trees are preferred over longer trees. Trees that place high information gain attributes close to the root are preferred over those that do not.

6.How does a decision tree handle continuous attributes?

By converting continuous attributes to a threshold-based boolean attribute. The threshold is decided by maximizing the information gain.

7.How does a decision tree handle missing attribute values?

One way to assign the most common value of that attribute to the missing attribute value. The other way is to assign a probability to each of the possible values of the attribute based on other samples.

8.Name some algorithms used for deriving decision trees?

ID3 and C4.5

9. What is the Decision Tree Algorithm?

A Decision Tree is a supervised machine learning algorithm that can be used for both Regression and Classification problem statements. It divides the complete dataset into smaller subsets while at the same time an associated Decision Tree is incrementally developed.

The final output of the Decision Trees is a Tree having Decision nodes and leaf nodes. A Decision Tree can operate on both categorical and numerical data.

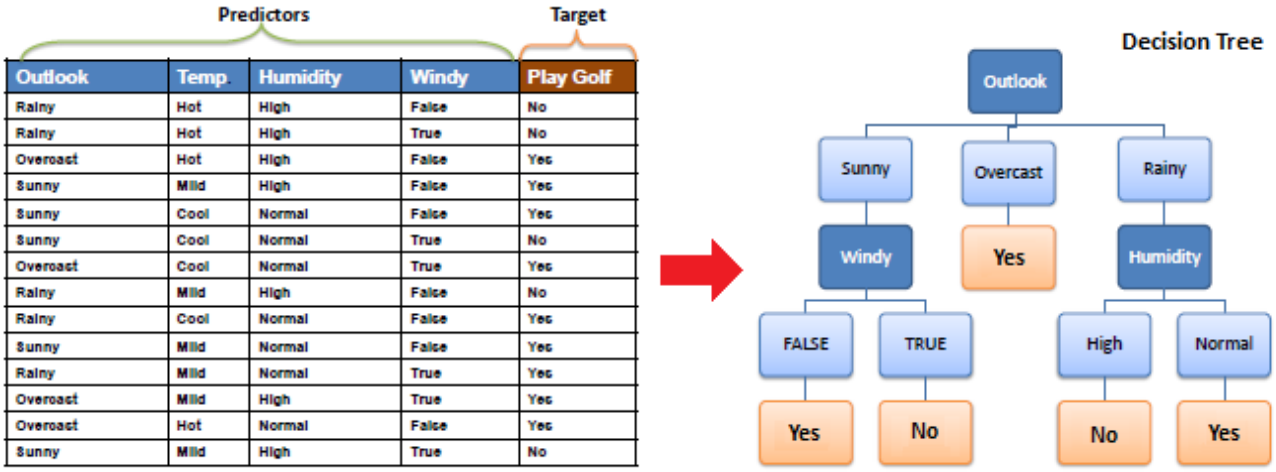


Image Source: Google Images

10. List down some popular algorithms used for deriving Decision Trees along with their attribute selection measures.

Some of the popular algorithms used for constructing decision trees are:

1. ID3 (Iterative Dichotomiser): Uses Information Gain as attribute selection measure.

2. C4.5 (Successor of ID3): Uses Gain Ratio as attribute selection measure.

3. CART (Classification and Regression Trees) – Uses Gini Index as attribute selection measure.

11. Explain the CART Algorithm for Decision Trees.

The CART stands for **Classification and Regression Trees** is a greedy algorithm that greedily searches for an optimum split at the top level, then repeats the same process at each of the subsequent levels.

Moreover, it does verify whether the split will lead to the lowest impurity or not as well as the solution provided by the greedy algorithm is not guaranteed to be optimal, it often produces a solution that's reasonably good since finding the optimal Tree is an **NP-Complete problem** that requires **exponential time complexity**.

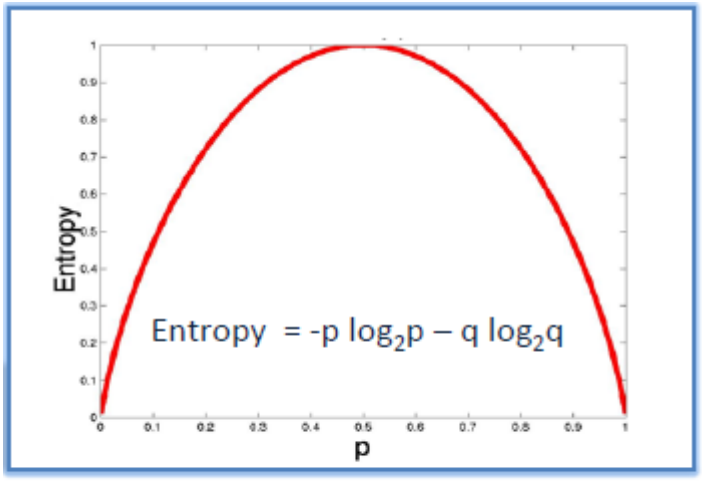
As a result, it makes the problem intractable even for small training sets. This is why we must go for a **"reasonably good"** solution instead of an optimal solution.

12. List down the attribute selection measures used by the ID3 algorithm to construct a Decision Tree.

The most widely used algorithm for building a Decision Tree is called ID3. ID3 uses Entropy and Information Gain as attribute selection measures to construct a Decision Tree.

1. Entropy: A Decision Tree is built top-down from a root node and involves the partitioning of data into homogeneous subsets. To check the homogeneity of a sample, ID3

uses entropy. Therefore, entropy is zero when the sample is completely homogeneous, and entropy of one when the sample is equally divided between different classes.



Entropy = $-0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$

2. Information Gain: Information Gain is based on the decrease in entropy after splitting a dataset based on an attribute. The meaning of constructing a Decision Tree is all about finding the attributes having the highest information gain.

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

13. Briefly explain the properties of Gini Impurity.

Let X (discrete random variable) takes values y_+ and y_- (two classes). Now, let's consider the different cases:

Case- 1: When 100% observations belong to y_+ . Then, the Gini impurity of the system would be: –

$$Ginx = 1 - (1^2 + 0^2) = 0$$

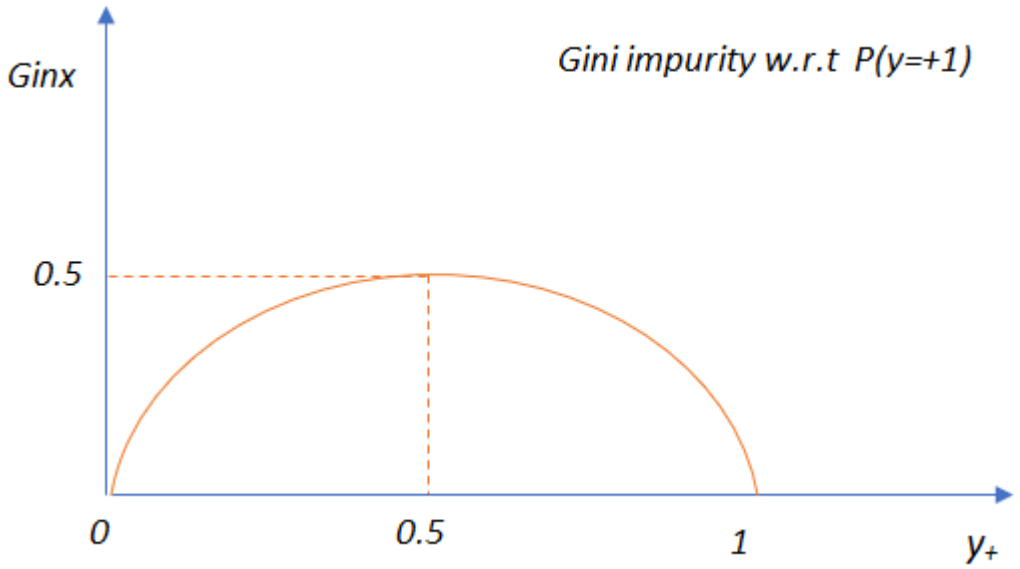
Case- 2: When 50% observations belong to y_+ . Then, the Gini impurity of the system would be: –

$$Ginx = 1 - (0.5^2 + 0.5^2) = 0.5$$

Case- 3: When 0% observations belong to y_+ . Then, the Gini impurity of the system would be: –

$$Ginx = 1 - (0^2 + 1^2) = 0$$

After observing all these cases, the graph of Gini impurity w.r.t to y_+ would come out to be:



14. Explain the difference between the CART and ID3 Algorithms.

The CART algorithm produces only binary Trees: non-leaf nodes always have two children (i.e., questions only have yes/no answers).

On the contrary, other Tree algorithms such as ID3 can produce Decision Trees with nodes having more than two children.

15. Which should be preferred among Gini impurity and Entropy?

In reality, most of the time it does not make a big difference: they lead to almost similar Trees. Gini impurity is a good default while implementing in sklearn since it is slightly faster to compute.However, when they work in a different way, then Gini impurity tends to isolate the most frequent class in its own branch of the Tree, while entropy tends to produce slightly more balanced Trees.

16. List down the different types of nodes in Decision Trees.

The Decision Tree consists of the following different types of nodes:

1. Root node: It is the top-most node of the Tree from where the Tree starts.

2. Decision nodes: One or more Decision nodes that result in the splitting of data into multiple data segments and our main goal is to have the children nodes with maximum homogeneity or purity.

3. Leaf nodes: These nodes represent the data section having the highest homogeneity.

17. What do you understand about Information Gain? Also, explain the mathematical formulation associated with it.

Information gain is the difference between the entropy of a data segment before the split and after the split i.e, reduction in impurity due to the selection of an attribute.

Some points keep in mind about information gain:

- The high difference represents high information gain.
- Higher the difference implies the lower entropy of all data segments resulting from the split.
- Thus, the higher the difference, the higher the information gain, and the better the feature used for the split.

Mathematically, the information gain can be computed by the equation as follows:

Information Gain = $E(S_1) – E(S_2)$

– **$E(S_1)$** denotes the entropy of data belonging to the node before the split.

– **$E(S_2)$** denotes the weighted summation of the entropy of children nodes by considering the weights as the proportion of data instances falling in specific children nodes.

18. Do we require Feature Scaling for Decision Trees? Explain.

Decision Trees are mainly intuitive, easy to interpret as well as require less data preparation. In fact, they don't require feature scaling or centering(standardization) at all. Such models are often called **white-box models**.Decision Trees provide simple classification rules based on if and else statements which can even be applied manually if need be.

For Example, Flower classification for the **Iris** dataset.

19. What are the disadvantages of Information Gain?

Information gain is defined as the reduction in entropy due to the selection of a particular attribute. Information gain biases the Decision Tree against considering attributes with a large number of distinct values which might lead to overfitting.

In order to solve this problem, the **Information Gain Ratio** is used.

20. List down the problem domains in which Decision Trees are most suitable.

Decision Trees are suitable for the following cases:

1. Decision Trees are most suitable for **tabular data**.

2. The outputs are **discrete**.

3. Explanations for Decisions are required.

4. The training data may contain errors, noisy data(outliers).

5. The training data may contain **missing feature** values.

21. Explain the time and space complexity of training and testing in the case of a Decision Tree.

Time and Space complexity for Training:

In the training stage for features (dimensions) in the dataset, we sort the data which takes **O(n log n)** time following which we traverse the data points to find the right threshold which takes **O(n)** time. Subsequently, for d dimensions, the total time complexity would be:

$$O(nlogn * d) + O(n * d)$$

$$which\ asymptotically\ is$$

$$O(nlogn * d)$$

Usually while training a decision tree we identify the nodes which are typically stored in the form of if-else statements due to which training space complexity is **O(nodes)**.

Time and Space Complexity for Testing:

Moreover, the testing time complexity is **O(depth)** as we have to traverse from the root to a leaf node of the decision tree i.e., testing space complexity is **O(nodes)**.

22. If it takes one hour to train a Decision Tree on a training set containing 1 million instances, roughly how much time will it take to train another Decision Tree on a training set containing 10 million instances?

As we know that the computational complexity of training a Decision Tree is given by **O(n × m log(m))**. So, when we multiplied the size of the training set by 10, then the training time will be multiplied by some factor, say K.

Now, we have to determine the value of K. To finds K, divide the complexity of both:

K = (n × 10m × log(10m)) / (n × m × log(m)) = 10 × log(10m) / log(m)

For 10 million instances i.e., m = 10^6, then we get the value of K ≈ 11.7.

Therefore, we can expect the training time to be roughly 11.7 hours.

23. How does a Decision Tree handle missing attribute values?

Decision Trees handle missing values in the following ways:

- Fill the missing attribute value by the most common value of that attribute.
- Fill the missing value by assigning a probability to each of the possible values of the attribute based on other samples.

24. How does a Decision Tree handle continuous(numerical) features?

Decision Trees handle continuous features by converting these continuous features to a **threshold-based boolean** feature.

To decide The threshold value, we use the concept of Information Gain, choosing that threshold that maximizes the information gain.

25. What is the Inductive Bias of Decision Trees?

The ID3 algorithm preferred Shorter Trees over longer Trees. In Decision Trees, attributes having high information gain are placed close to the root are preferred over those that do not.

26. Explain Feature Selection using the Information Gain/Entropy Technique.

The goal of the feature selection while building a Decision Tree is to select features or attributes (Decision nodes) which lead to a split in children nodes whose combined entropy adds up to lower entropy than the entropy value of the data segment before the split. This implies higher information gain.

27. Compare the different attribute selection measures.

The three measures, in general, returns good results, but:

1. Information Gain: It is biased towards multivalued attributes

2. Gain ratio: It prefers unbalanced splits in which one data segment is much smaller than the other segment.

3. Gini Index: It is biased to multivalued attributes, has difficulty when the number of classes is large, tends to favor tests that result in equal-sized partitions and purity in both partitions.

28. Does the Gini Impurity of a node lower or greater than that of its parent. Comment whether it is generally lower/greater, or always lower/greater?

A node's Gini impurity is generally lower than that of its parent as the CART training algorithm cost function splits each of the nodes in a way that minimizes the weighted sum of its children's Gini impurities. However, sometimes it is also possible for a node to have a higher Gini impurity than its parent but in such cases, the increase is more than compensated by a decrease in the other child's impurity.

For better understanding we consider the following Example:

Consider a node containing four samples of class A and one sample of class B.

Then, its Gini impurity is calculated as $\mathbf{1 - (1/5)^2 - (4/5)^2 = 0.32}$

Now suppose the dataset is one-dimensional and the instances are arranged in the manner: A, B, A, A, A. We can verify that the algorithm will split this node after the second instance, producing one child node with instances A, B, and the other child node with instances A, A, A.

Then, the first child node's Gini impurity is $\mathbf{1 - (1/2)^2 - (1/2)^2 = 0.5}$, which is higher than its parent's. This is compensated for by the fact that the other node is pure, so its overall weighted Gini impurity is $\mathbf{2/5 \times 0.5 + 3/5 \times 0 = 0.2}$, which is lower than the parent's Gini impurity.

29. Why do we require Pruning in Decision Trees? Explain.

After we create a Decision Tree we observe that most of the time the leaf nodes have very high homogeneity i.e., properly classified data. However, this also leads to overfitting. Moreover, if enough partitioning is not carried out then it would lead to underfitting.

Hence the major challenge that arises is to find the optimal trees which result in the appropriate classification having acceptable accuracy. So to cater to those problems we first make the decision tree and then use the error rates to appropriately prune the trees.

30. Are Decision Trees affected by the outliers? Explain.

Decision Trees are not sensitive to noisy data or outliers since, extreme values or outliers, never cause much reduction in **Residual Sum of Squares(RSS),** because they are never involved in the split.

31. What do you understand by Pruning in a Decision Tree?

When we remove sub-nodes of a Decision node, this process is called pruning or the opposite process of splitting. The two techniques which are widely used for pruning are- Post and Pre Pruning.

Post Pruning:

- This type of pruning is used after the construction of the Decision Tree.
- This technique is used when the Decision Tree will have a very large depth and will show the overfitting of the model.
- It is also known as backward pruning.
- This technique is used when we have an infinitely grown Decision Tree.

Pre Pruning:

- This technique is used before the construction of the Decision Tree.
- Pre-Pruning can be done using Hyperparameter tuning.
- Overcome the overfitting issue.

32. List down the advantages of the Decision Trees.

1. Clear Visualization: This algorithm is simple to understand, interpret and visualize as the idea is mostly used in our daily lives. The output of a Decision Tree can be easily interpreted by humans.

2. Simple and easy to understand: Decision Tree works in the same manner as simple if-else statements which are very easy to understand.

3. This can be used for both classification and regression problems.

4. Decision Trees can handle both continuous and categorical variables.

5. No feature scaling required: There is no requirement of feature scaling techniques such as standardization and normalization in the case of Decision Tree as it uses a rule-based approach instead of calculation of distances.

6. Handles nonlinear parameters efficiently: Unlike curve-based algorithms, the performance of decision trees can't be affected by the Non-linear parameters. So, if there is high non-linearity present between the independent variables, Decision Trees may outperform as compared to other curve-based algorithms.

7. Decision Tree can automatically handle missing values.

8. Decision Tree handles the outliers automatically, hence they are usually robust to outliers.

9. Less Training Period: The training period of decision trees is less as compared to ensemble techniques like Random Forest because it generates only one Tree unlike the forest of trees in the Random Forest.

33. List out the disadvantages of the Decision Trees.

1. Overfitting: This is the major problem associated with the Decision Trees. It generally leads to overfitting of the data which ultimately leads to wrong predictions for testing data points. it keeps generating new nodes in order to fit the data including even noisy data and ultimately the Tree becomes too complex to interpret. In this way, it loses its generalization capabilities. Therefore, it performs well on the training dataset but starts making a lot of mistakes on the test dataset.

2. High variance: As mentioned, a Decision Tree generally leads to the overfitting of data. Due to the overfitting, there is more likely a chance of high variance in the output which leads to many errors in the final predictions and shows high inaccuracy in the results. So, in order to achieve zero bias (overfitting), it leads to high variance due to the bias-variance tradeoff.

3. Unstable: When we add new data points it can lead to regeneration of the overall Tree. Therefore, all nodes need to be recalculated and reconstructed.

4. Not suitable for large datasets: If the data size is large, then one single Tree may grow complex and lead to overfitting. So in this case, we should use Random Forest instead, an ensemble technique of a single Decision Tree.

34. What is a decision tree in data science?

A decision tree is a machine learning algorithm that is used for both classification and regression tasks. The algorithm works by breaking down a dataset into smaller and smaller subsets, until each subset contains only one data point. The algorithm then uses the data points in each subset to make a prediction about the target variable.

35. Can you explain how to create a decision tree model in Python?

In Python, you can create a decision tree model using the scikit-learn library. This library provides a DecisionTreeClassifier class that you can use to train your model. You will need to provide training data to the classifier, which it will use to create the tree. Once the tree is created, you can use it to make predictions on new data.

36. How are the values of decision nodes calculated in a decision tree?

The values of decision nodes are calculated by finding the entropy of the child nodes. The entropy is a measure of how impure or pure a node is. A node is pure if all of its children are of the same class. The entropy is calculated by taking the sum of the negative logarithms of the probabilities of the child nodes.

37. Can you explain what entropy is and how it's used in decision trees?

Entropy is a measure of how disordered or random a system is. In the context of decision trees, entropy is used to measure how pure a given node is. A node is pure if all of the examples in it belong to the same class. If a node is not pure, then it is said to be impure. The entropy of a node is calculated by taking the sum of the negative logarithms of the probabilities of the node being in each class. The entropy is used to help determine which attribute should be used to split the node.

38. What metrics do you use to evaluate a decision tree model?

The most common metric for evaluating a decision tree model is accuracy. This measures how often the model correctly predicts the target class. Other metrics you might use include precision, recall, and the F1 score.

39. Can you explain information gain and its usage in a decision tree?

Information gain is a measure of how much information is gained by making a particular decision. In a decision tree, information gain is used to determine which attribute should be used to split the data at each node. The attribute with the highest information gain is chosen, and the data is split accordingly. Information gain is thus a key part of the decision tree learning algorithm.

40. What are some examples of bias in machine learning?

Some examples of bias in machine learning include selection bias, survivorship bias, and self-fulfilling prophecies.

41. Why are decision trees preferred over other algorithms like linear regression or random forests?

Decision trees are preferred over other algorithms for a few reasons. First, decision trees are very easy to interpret and explain. This is because they can be visualized as a flowchart, which makes them easy to understand for people who are not familiar with complex mathematical models. Second, decision trees are very flexible and can be used for both regression and classification tasks. Third, decision trees are not sensitive to outliers, meaning

that they can still produce accurate predictions even if there are a few data points that are very different from the rest.

42. What is a node in a decision tree?

A node is a point in the decision tree where a decision is made. This decision can be based on a variety of factors, but is typically based on some value in the data that is being processed by the tree. Nodes can be either internal nodes, which make a decision and have branches leading to other nodes, or leaf nodes, which do not have any branches and simply represent a final decision.

43. What is meant by pruning in a decision tree?

Pruning is the process of removing unnecessary branches from a decision tree in order to improve its accuracy. This is done by first constructing the tree using a training dataset, and then testing the tree on a separate dataset. Branches that do not improve the accuracy of the tree are then removed.

44. What does it mean if a decision tree has low variance but high bias?

This means that the decision tree is not overfitting the data, but that it is not capturing all of the relevant information in the data either. This can be due to a number of factors, such as a small training set, or a simple model.

45. What is meant by bagging and boosting in context with decision trees?

Bagging and boosting are two methods used to improve the performance of decision trees. Bagging involves training multiple decision trees on different subsets of the data, and then averaging the predictions of the trees. Boosting involves training multiple decision trees on different subsets of the data, but each tree is trained on a subset that is weighted towards instances that the previous trees in the ensemble misclassified.

46. What techniques can be used to prevent overfitting in Decision Trees?

Some techniques that can be used to prevent overfitting in Decision Trees are pruning, setting a minimum number of samples required at a leaf node, and setting a maximum depth for the tree.

47. What are the advantages and disadvantages of using decision trees?

Decision trees are a type of machine learning algorithm that can be used for both classification and regression tasks. The advantages of using decision trees include that they are easy to interpret and explain, they can handle both numerical and categorical data, and they are relatively robust to outliers. The disadvantages of using decision trees include that they can be prone to overfitting, and they may not be the best choice for very high-dimensional data.

48. What are the various ways of splitting a node in a decision tree?

The various ways of splitting a node in a decision tree are known as splitting criteria. The most common splitting criteria are information gain, gini index, and chi-square. Information gain is the simplest to calculate and understand, and it usually works well in practice. Gini index is a slightly more sophisticated measure that takes into account the relative sizes of the classes in the node. Chi-square is a statistical measure that is used to test whether two variables are independent of each other.

49. What is "categorical" data and why is it important when working with decision trees?

Categorical data is data that can be divided into distinct groups or categories. This is important when working with decision trees because the algorithm used to create the tree relies on being able to identify a clear boundary between different groups of data. If the data is not clearly divided into groups, then the tree will not be able to accurately predict outcomes.

50. What are some real-world applications where decision trees are used?

Decision trees are used in a variety of settings, including but not limited to:
-Classifying emails as spam or not spam
-Predicting whether or not a customer will default on a loan
-Determining whether or not an insurance claim is fraudulent

51. What is your opinion on decision trees as a supervised algorithm? Do they work better for certain types of problems than others?

I think decision trees can be a very powerful supervised algorithm, particularly for classification problems. They tend to be very intuitive and easy to interpret, which can be helpful in understanding the data and the relationships between variables. However, they can also be prone to overfitting, so it is important to be careful when using them. I think they work best on problems with a relatively small number of features, where the relationships between variables are relatively simple.

52. What is meant by gini index when dealing with decision trees?

The gini index is a measure of how impure a given node is. A node is pure if all of the data points in that node belong to the same class. The gini index is calculated by taking the sum of the squared probabilities of each class and subtracting it from 1. The gini index can be used to help choose the best split point for a decision tree.

53. What is ID3?

ID3 is a decision tree algorithm that is used to generate a decision tree from a given dataset. It works by constructing a tree from the given data, and then using the ID3 algorithm to determine which attribute of the data should be used to split the data into different branches.

54. You will see two statements listed below. You will have to read both of them carefully and then choose one of the options from the two statements' options. The contextual question is, Choose the statements which are true about bagging trees.

The individual trees are not at all dependent on each other for a bagging tree.

To improve the overall performance of the model, the aggregate is taken from weak learners. This method is known as bagging trees.

Only statement number one is TRUE.

Only statement number two is TRUE.

Both statements one and two are TRUE.

None of the options which are mentioned above.

Ans. The correct answer to this question is C because, for a bagging tree, both of these statements are true. In bagging trees or bootstrap aggregation, the main goal of applying this algorithm is to reduce the amount of variance present in the decision tree. The mechanism of creating a bagging tree is that with replacement, a number of subsets are taken from the sample present for training the data.

Now, each of these smaller subsets of data is used to train a separate decision tree. Since the information which is fed into each tree comes out to be unique, the likelihood of any tree having any impact on the other becomes very low. The final result which all these trees give is collected and then processed to provide the output. Thus, the second statement also comes out to be true.

55. You will see two statements listed below. You will have to read both of them carefully and then choose one of the options from the two statements' options. The contextual question is, Choose the statements which are true about boosting trees.

The weak learners in a boosting tree are independent of each other.

The weak learners' performance is all collected and aggregated to improve the boosted tree's overall performance.

Only statement number one is TRUE.

Only statement number two is TRUE.

Both statements one and two are TRUE.

None of the options which are mentioned above.

Ans. If you were to understand how the boosting of trees is done, you will understand and will be able to differentiate the correct statement from the statement, which is false. So, a boosted tree is created when many weak learners are connected in series. Each tree present in this sequence has one sole aim: to reduce the error which its predecessor made.

If the trees are connected in such fashion, all the trees cannot be independent of each other, thus rendering the first statement false. When coming to the second statement, it is true mainly because, in a boosted tree, that is the method that is applied to improve the overall performance of the model. The correct option will be B, i.e., only the statement number two is TRUE, and the statement number one is FALSE.

56. You will see four statements listed below. You will have to read all of them carefully and then choose one of the options from the options which follows the four statements. The contextual question is, Choose the statements which are true about Radom forests and Gradient boosting ensemble method.

Both Random forest and Gradient boosting ensemble methods can be used to perform classification.

Random Forests can be used to perform classification tasks, whereas the gradient boosting method can only perform regression.

Gradient boosting can be used to perform classification tasks, whereas the Random Forest method can only perform regression.

Both Random forest and Gradient boosting ensemble methods can be used to perform regression.

Only statement number one is TRUE.

Only statement number two is TRUE.

Both statements one and two are TRUE.

Only statement number three is TRUE

Only statement number four is TRUE

Only statement number one and four is TRUE

Ans. The answer to this question is straightforward. Both of these ensemble methods are actually very capable of doing both classification and regression tasks. So, the answer to this question would be F because only statements number one and four are TRUE.

57. You will see four statements listed below. You will have to read all of them carefully and then choose one of the options from the options which follows the four statements. The contextual question is, consider a random forest of trees. So what will be true about each or any of the trees in the random forest?

Each tree which constitutes the random forest is based on the subset of all the features.

Each of the in a random forest is built on all the features.

Each of the trees in a random forest is built on a subset of all the observations present.

Each of the trees in a random forest is built on the full observation set.

Only statement number one is TRUE.

Only statement number two is TRUE.

Both statements one and two are TRUE.

Only statement number three is TRUE

Only statement number four is TRUE

Both statements number one and four are TRUE

Both the statements number one and three are TRUE

Both the statements number two and three are TRUE

Both the statements number two and four are TRUE

Ans. The generation of random forests is based on the concept of bagging. To build a random forest, a small subset is taken from both the observations and the features. The values which are obtained after taking out the subsets are then fed into singular decision trees. Then all the values from all such decision trees are collected to make the final decision. That means the only statements which are correct would be one and three. So, the right option would be G.

58. You will see four statements listed below. You will have to read all of them carefully and then choose one of the options from the options which follows the four statements.

The contextual question is, select the correct statements about the hyperparameter known as "max_depth" of the gradient boosting algorithm.

Choosing a lower value of this hyperparameter is better if the validation set's accuracy is similar.

Choosing a higher value of this hyperparameter is better if the validation set's accuracy is similar.

If we are to increase this hyperparameter's value, then the chances of this model actually overfitting the data increases.

If we are to increase this hyperparameter's value, then the chances of this model actually underfitting the data increases.

Only statement number one is TRUE.

Only statement number two is TRUE.

Both statements one and two are TRUE.

Only statement number three is TRUE

Only statement number four is TRUE

Both statements number one and four are TRUE

Both the statements number one and three are TRUE

Both the statements number two and three are TRUE

Both the statements number two and four are TRUE

Ans. The hyperparameter max_depth controls the depth until the gradient boosting will model the presented data in front of it. If you keep on increasing the value of this hyperparameter, then the model is bound to overfit. So, statement number three is correct. If we have the same scores on the validation data, we generally prefer the model with a lower depth. So, statements number one and three are correct, and thus the answer to this decision tree interview questions is g.

59. You will see four statements listed below. You will have to read all of them carefully and then choose one of the options from the options which follows the four statements. The contextual question is which of the following methods does not have a learning rate as one of their tunable hyperparameters.

Extra Trees.

AdaBoost

Random Forest

Gradient boosting.

Only statement number one is TRUE.

Only statement number two is TRUE.

Both statements one and two are TRUE.

Only statement number three is TRUE

Only statement number four is TRUE

Both statements number one and four are TRUE

Both the statements number one and three are TRUE

Both the statements number two and three are TRUE

Both the statements number two and four are TRUE

Ans. Only Extra Trees and Random forest does not have a learning rate as one of their tunable hyperparameters. So, the answer would be g because the statement number one and three are TRUE.

60. Choose the option, which is true.

Only in the algorithm of random forest, real values can be handled by making them discrete.

Only in the algorithm of gradient boosting, real values can be handled by making them discrete.

In both random forest and gradient boosting, real values can be handled by making them discrete.

None of the options which are mentioned above.

Ans. Both of the algorithms are capable ones. They both can easily handle the features which have real values in them. So, the answer to this decision tree interview questions and answers is C.

61. Choose one option from the list below. The question is, choose the algorithm which is not an ensemble learning algorithm.

Gradient boosting

AdaBoost

Extra Trees

Random Forest

Decision Trees

Ans. This question is straightforward. Only one of these algorithms is not an ensemble learning algorithm. One thumb rule to keep in mind will be that any ensemble learning method would involve the use of more than one decision tree. Since in option E, there is just the singular decision tree, then that is not an ensemble learning algorithm. So, the answer to this question would be E (decision trees).

62. You will see two statements listed below. You will have to read both of them carefully and then choose one of the options from the two statements' options. The contextual question is, which of the following would be true in the paradigm of ensemble learning.

The tree count in the ensemble should be as high as possible.

1. You will still be able to interpret what is happening even after you implement the algorithm of Random Forest.
2. Only statement number one is TRUE.
3. Only statement number two is TRUE.
4. Both statements one and two are TRUE.
5. None of the options which are mentioned above.

Ans. Since any ensemble learning method is based on coupling a colossal number of decision trees (which on its own is a very weak learner) together so it will always be beneficial to have more number of trees to make your ensemble method. However, the algorithm of random forest is like a black box. You will not know what is happening inside the model. So, you are bound to lose all the interpretability after you apply the random forest algorithm. So, the correct answer to this question would be A because only the statement that is true is the statement number one.

63. What are the advantages and disadvantages of using Random Forest?

The advantages of using Random Forest are that it is a very accurate and versatile machine learning algorithm. It can be used for both regression and classification tasks, and it is not prone to overfitting. The disadvantages of using Random Forest are that it is a black box algorithm, meaning that it is difficult to interpret the results, and it is also computationally expensive.

64. What are the main differences between a decision tree and a random forest?

The main difference between a decision tree and a random forest is that a decision tree is built using a single tree, while a random forest is built using a collection of trees. A random

forest is more accurate than a decision tree because it can reduce the variance of the predictions by averaging the results of the individual trees.

65. What do you need to specify before building a random forest model?

In order to build a random forest model, you need to specify the number of trees in the forest, the number of features to consider when looking for the best split, the minimum number of samples required to split a node, and the minimum number of samples required to be at a leaf node.

66. What are the parameters that affect the behavior of a random forest?

The parameters that affect the behavior of a random forest are the number of trees in the forest, the number of features considered at each split, the minimum number of samples required to split a node, the maximum depth of the tree, and the minimum number of samples required to be at a leaf node.

67. What are the steps involved in creating a random forest model in python?

The steps involved in creating a random forest model in python are as follows:

1. Import the required libraries
2. Load the dataset
3. Split the dataset into training and test sets
4. Train the model on the training set
5. Make predictions on the test set
6. Evaluate the model

68. What are the different types of node split methods in Random Forest?

The different types of node split methods in Random Forest are:

– Gini impurity
– Information gain
– Chi-squared

Each of these methods has its own advantages and disadvantages, so it is important to choose the one that is best suited for your data and your task.

69. What are the various metrics used for evaluating the performance of a random forest?

The most common metric used to evaluate the performance of a random forest is the accuracy score. However, other metrics, such as the F1 score, can also be used.

70. Can you give me some examples of how you would use a random forest model in data science?

There are a few different ways that you could use a random forest model in data science. One way would be to use it for classification tasks, such as identifying which customers are likely to churn or predicting whether or not a loan will default. Another way you could use it would be for regression tasks, such as predicting housing prices or stock returns. Finally, you could also use a random Forest model to help you understand which features are most important in predicting a particular outcome.

71. Can you explain what overfitting and underfitting mean in the context of random forests?

Overfitting occurs in a machine learning model when the model has been trained too closely to the training data, and as a result, the model does not generalize well to new data. This means that the model is not able to accurately predict the output for new data points. Underfitting occurs when the model has not been trained enough, and as a result, it does not accurately capture the patterns in the training data. This results in a model that does not perform well on either the training data or new data.

72. What are the best practices you should follow when using random forest models?

There are a few best practices to follow when using random forest models:

1. Make sure that your data is properly prepared and cleaned before training the model. This includes dealing with missing values, outliers, and other issues.

2. Train your model on a variety of different data sets to ensure that it is generalizing well and not overfitting.

3. Tune the hyperparameters of your model to find the best possible performance.

4. Evaluate your model on a hold-out set of data to get an accurate estimate of its performance.

73. What do you understand about bias and variance in the context of machine learning?

Bias and variance are two important concepts when it comes to machine learning. Bias refers to the error that is introduced by the simplifying assumptions that are made when a model is created. Variance, on the other hand, is the error that is introduced by the fact that the data used to train the model is not representative of the entire population.

74. What do you understand about homoscedastic and heteroscedastic distributions?

A homoscedastic distribution is one where the variance is constant across all values of the random variable. A heteroscedastic distribution is one where the variance is not constant, and instead varies depending on the value of the random variable.

75. Can you explain what ensemble learning is?

Ensemble learning is a machine learning technique that combines the predictions of multiple models to create a more accurate prediction. This is often done by training multiple models on different subsets of the data and then averaging the predictions of the models.

76. Can you explain what covariance matrices and correlation coefficents are?

A covariance matrix is a matrix that shows the covariance between two or more variables. A correlation coefficient is a number that represents how two variables are correlated.

77. Can you explain the difference between Mean Absolute Error and Root Mean Square Error? Which one do you think is a better measure of accuracy?

Mean Absolute Error is simply the average of the absolute values of the differences between the predicted values and the actual values. Root Mean Square Error is the square root of the average of the squared differences between the predicted values and the actual values. I think that Root Mean Square Error is a better measure of accuracy because it punishes large errors more than Mean Absolute Error does.

78. Can you explain what information gain is?

Information gain is a measure of how much information is gained by splitting a dataset on a given attribute. The higher the information gain, the more “useful” the attribute is for splitting the dataset.

79. What is a classification and regression forest?

A classification and regression forest is a machine learning algorithm that can be used for both classification and regression tasks. It is a type of ensemble learning algorithm, which means that it combines the predictions of multiple individual models to produce a more accurate overall prediction.

80. What are the advantages of using a Regression Forest?

There are several advantages of using a regression forest, including:

– Increased accuracy: By averaging the results of multiple trees, a regression forest can provide more accurate predictions than a single tree.

– Reduced overfitting: By averaging the results of multiple trees, a regression Forest can help to reduce overfitting.

– Increased interpretability: By looking at the results of multiple trees, it can be easier to understand how a regression Forest is making predictions.

81. How does a random forest differ from a bagging algorithm?

A random forest is a type of bagging algorithm, but with a few key differences. First, a random Forest uses a random subset of features when building each decision tree, rather than using all features as a bagging algorithm would. Second, a random Forest uses a random subset of data points when building each decision tree, rather than using all data points as a bagging algorithm would. Finally, a random Forest uses a voting system to make predictions, rather than relying on a single decision tree.

82. What is a bootstrap sample? How can it be generated?

A bootstrap sample is a randomly generated sample of data that is used to estimate the population parameters. This is done by randomly selecting a unit from the population and then repeating this process a number of times. The bootstrap sample can be generated in R by using the bootstrap() function.

83. What do you mean by Random Forest Algorithm?

Random forest is an ensemble machine learning technique that averages several decision trees on different parts of the same training set, with the objective of overcoming the overfitting problem of the individual decision trees. In other words, a random forest algorithm is used for both classification and regression problem statements that operate by constructing a lot of decision trees at training time.

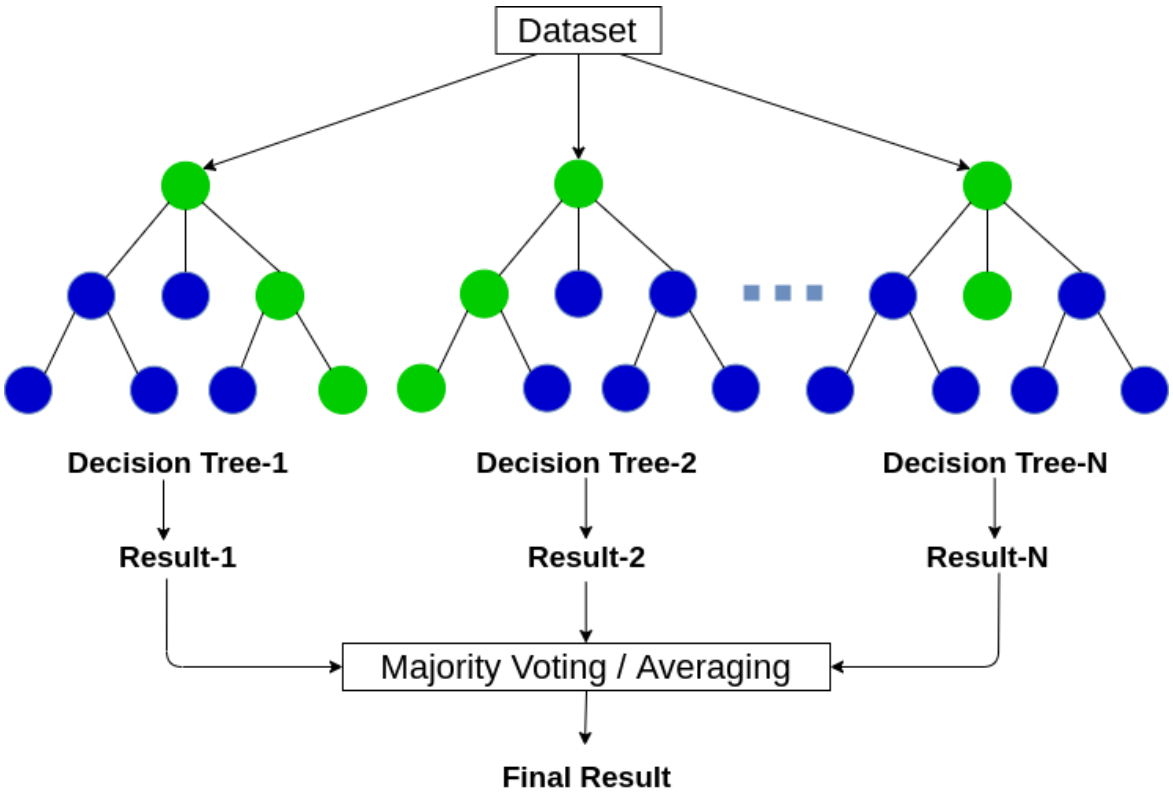


Image Source: Google Images

84. Why is Random Forest Algorithm popular?

Random Forest is one of the most popular and widely used machine learning algorithms for classification problems. It can also be used for the regression problem statements but it mainly performs well on the classification model.

It has become a lethal weapon for modern data scientists to refine the predictive model. The best part of the algorithm is that there are very few assumptions attached to it so data preparation is less challenging which results in time-saving. It's listed as a top algorithm (with ensembling) that is popular among the Kaggle Competitions.

85. Can Random Forest Algorithm be used both for Continuous and Categorical Target Variables?

Yes, Random Forest can be used for both continuous and categorical target (dependent) variables.

In a random forest i.e, the combination of decision trees, the classification model refers to the categorical dependent variable, and the regression model refers to the numeric or continuous dependent variable.

86. What do you mean by Bagging?

Bagging, also known as Bootstrap-Aggregating, involves generating K's new training data sets. Each new training data set picks a sample of data points with replacement (known as bootstrap samples) from the original data set.

By sampling with replacement, means some of the data points may be repeated in each new training data set. The K models are fitted using the K bootstrap samples formed and then for predictions we combined the result of all trees by averaging the output (for regression) or voting (for classification).

87. Explain the working of the Random Forest Algorithm.

The steps that are included while performing the random forest algorithm are as follows:

Step-1: Pick K random records from the dataset having a total of N records.

Step-2: Build and train a decision tree model on these K records.

Step-3: Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

Step-4: In the case of a regression problem, for an unseen data point, each tree in the forest predicts a value for output. The final value can be calculated by taking the mean or average of all the values predicted by all the trees in the forest.

and, in the case of a classification problem, each tree in the forest predicts the class to which the new data point belongs. Finally, the new data point is assigned to the class that has the maximum votes among them i.e, wins the majority vote.

88. Why do we prefer a Forest (collection of Trees) rather than a single Tree?

While building a machine learning model, our aim is to generalize the model properly for giving predictions on unseen data.

The problem of overfitting takes place when we have a flexible model. A flexible model is having high variance because the learned parameters like the structure of the decision tree, etc will vary with the training data. On the contrary, an inflexible model is said to have a high bias as it makes assumptions about the training data and an inflexible model may not have the capacity to fit even the training data and in both situations, the model has high variance, and high bias implies the model is not able to generalize new and unseen data points properly.

So, we have to build a model carefully by keeping the bias-variance tradeoff in mind.

The main reason for the overfitting of the decision tree due to not put the limit on the maximum depth of the tree is because it has unlimited flexibility, which means it keeps growing unless, for every single observation, there is one leaf node present.

Moreover, instead of limiting the depth of the tree which results in reduced variance and an increase in bias, we can combine many decision trees that eventually convert into a forest, known as a single ensemble model (known as the random forest).

89. What do you mean by Bootstrap Sample?

It is basically random with the replacement sampling method.

For Example, Suppose we have a box of lottery tickets in which there are 100 unique numbers from 0 to 99. We want to select a random sample of tickets from the box. If we put the ticket back in the box, it may be selected more than once. Therefore, in this process, we are picking the samples randomly from the box with replacement.

90. What is Out-of-Bag Error?

Out-of-Bag is equivalent to validation or test data. In random forests, there is no need for a separate testing dataset to validate the result. It is calculated internally, during the algorithm run, in the following manner –

As the forest is built on training data, each tree is tested on 1/3rd of the samples (36.8%) that are not used in building that tree (similar to the validation data set).

This is known as the out-of-bag error estimate which in short is an internal error estimate of a random forest as it is being constructed.

91. What does random refer to in 'Random Forest'?

'Random' in Random Forest refers to mainly two processes –

Random observations to grow each tree.

Random variables selected for splitting at each node.

Random Record Selection: Each tree in the forest is trained on roughly 2/3rd of the total training data (exactly 63.2%) and here the data points are drawn at random with replacement from the original training dataset. This sample will act as the training set for growing the tree.

Random Variable Selection: Some independent variables(predictors) say, m are selected at random out of all the predictor variables, and the best split on this m is used to split the node.

NOTE:

By default, m is taken as the square root of the total number of predictors for classification whereas m is the total number of all predictors divided by 3 for regression problems.

The value of m remains constant during the algorithm run i.e, forest growing.

92. Why does the Random Forest algorithm not require split sampling methods?

Random Forest does not require a split sampling method to assess the accuracy of the model.

This is because it performs internal testing on 2/3rd of the available training data that is used to grow each tree and the remaining one-third portion of training data is always used to calculate out-of-bag error to compute the model performance.

93. List down the features of Bagged Trees.

The main features of Bagged Trees are as follows:

1. Reduces variance by averaging the ensemble's results.

2. The resulting model uses the entire feature space when considering node splits.

3. It allows the trees to grow without pruning, reducing the tree-depth sizes which result in high variance but lower bias, which can help improve the prediction power.

94. What are the Limitations of Bagging Trees?

The major limitation of bagging trees is that it uses the entire feature space when creating splits in the trees.

Suppose from all the variables within the feature space, some are indicating certain predictions, so there is a risk of having a forest of correlated trees, which actually increases bias and reduces variance. So, our objective is not achieved due to these issues.

95. List down the factors on which the forest error rate depends upon.

The forest error rate in Random forest depends on the following two factors:

1. How correlated the two trees in the forest are i.e,

The correlation between any two different trees in the forest. Increasing the correlation increases the forest error rate.

2. How strong each individual tree in the forest is i.e,

The strength of each individual tree in the forest. In a forest, a tree having a low error rate is considered a strong classifier. Increasing the strength of the individual trees eventually leads to a decrement in the forest error rate.

Moreover, reducing the value of mtry i.e, the number of random variables used in each tree reduces both the correlation and the strength. Increasing it increases both. So, in between, there exists an "optimal" range of mtry which is usually quite a wide range.

Using the OOB error rate, a value of mtry can quickly be found in the range. This parameter is only adjustable from which random forests are somewhat sensitive.

96. How does a Random Forest Algorithm give predictions on an unseen dataset?

After training the algorithm, each tree in the forest gives a classification on leftover data (OOB), and we say the tree "votes" for that class. Then finally, the forest chooses the classification having the most votes over all the trees in the forest.

For a binary dependent variable, the vote will be either YES or NO, and finally, it will count up the YES votes. This is the Random Forest (RF) score and the percent YES votes received is the predicted probability. In the regression case, it is the average of the dependent variable.

For example, suppose we fit 500 trees in a forest, and a case is out-of-bag in 200 of them:

160 trees vote class 1

40 trees vote class 2

In this case, the RF score is class1 since the probability for that case would be 0.8 which is 160/200. Similarly, it would be an average of the target variable for the regression problem.

97. Prove that in the Bagging method only about 63% of the total original examples (total training set) appear in any of sampled bootstrap datasets. Provide proper justification.

The detailed explanation of the proof is as follows:

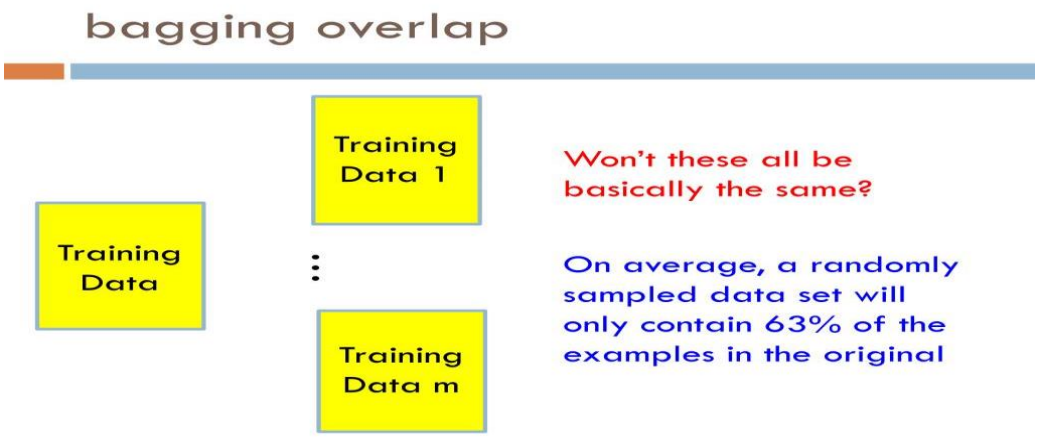
Input: n labelled training examples $S = \{(x_i, y_i)\}, i = 1,..,n$

Suppose we select n samples out of n with replacement to get a training set S_i still different from working with the entire training set.

$Pr(S_i = S) = n!/n^n$ (very small number, exponentially small in n)

$Pr((x_i,y_i)$ not in $S_i) = (1-1/n)^n = e^{-1} \sim 0.37$

Hence for large data sets, about 37% of the data set is left out!



98. How to determine the overall OOB score for the classification problem statements in a Random Forest Algorithm?

For each tree, by using the leftover (**36.8%**) data, compute the misclassification rate, which is known as **out of bag (OOB) error rate**. Finally, we aggregate all the errors from all trees and we will determine the overall OOB error rate for the classification.

For Example, If we grow 300 trees then on average a record will be OOB for about **37*3 =111** trees.

99. How does random forest define the Proximity (Similarity) between observations?

Random Forest defines proximity between two data points in the following way:

- Initialize proximities to zeroes.
- For any given tree, apply all the cases to the tree.
- If case i and case j both end up in the same node, then proximity prox(ij) between i and j increases by one.
- Accumulate over all trees in Random Forest and normalize by twice the number of trees in Random forest.

Finally, it creates a proximity matrix i.e, a square matrix with entry as 1 on the diagonal and values between 0 and 1 in the off-diagonal positions. Proximities are close to 1 when the observations are “alike” and conversely the closer proximity to 0, implies the more dissimilar cases are.

100. What is the use of proximity matrix in the random forest algorithm?

A proximity matrix is used for the following cases :

- Missing value imputation
- Detection of outliers

101. List down the parameters used to fine-tune the Random Forest.

Two parameters that have to fine-tune to improve the predictions that are important in the random forest algorithm are as follows:

- Number of trees used in the forest (**n_tree**)
- Number of random variables used in each of the trees in the forest (**mtry**)

102. How to find an optimal value of the hyperparameter " n_tree"?

To find an optimal value of n_tree, we first fix the value of mtry to the default value (sqrt of the total number of all predictors) and search for the optimal n_tree value.

To find the value of n_tree (number of trees) that corresponds to a stable classifier, we train random forest models with different values of n_tree such as (100, 200, 300….,1,000).

As a result, we have 10 Random Forest classifiers in our hand for each value of n_tree, record the OOB error rate and see that value of n_tree where the out-of-bag error rate stabilizes and reaches its minimum value.

103. How to find an optimal value of the hyperparameter "mtry"?

The following two ways can be used to find the optimal value of mtry :

1. Apply a similar procedure as in finding the optimal n_tree such that random forest is run 10 times. The optimal number of predictors selected for split is selected for which the out-of-bag error rate stabilizes and reaches the minimum.

2. In this method, we are doing the experiment by including the values such as the square root of the total number of all predictors, half of this square root value, and twice of the square root value, etc and at the same time check which value of mtry gives the maximum area under the curve.

For Example, Suppose we have 1000 predictors, then the number of predictors to select for each node would be 16, 32, and 64.

104. How do Random Forests select the Important Features?

Sometimes random forests can also be used to determine the importance of variables i.e, rank in a regression or classification problem.

The factors that are used to find the rank of the variables are as follows:

- **Mean Decrease Accuracy:** If we drop that variable, how much the model accuracy decreases.
- **Mean Decrease Gini:** This measure of variable importance is used for the calculation of splits in trees based on the Gini impurity index.

Conclusion: The higher the value of mean decrease accuracy or mean decrease Gini score, the higher the importance of the variable in the model.

105. Explain the steps of calculating Variable Importance in Random Forest.

The steps for calculating variable importance in Random Forest Algorithm are as follows:

1. For each tree grown in a random forest, find the number of votes for the correct class in out-of-bag data.

2. Now perform random permutation of a predictor's values (let's say variable-k) in the OOB data and then check the number of votes for the correct class. By "random permutation of a predictor's values", it means changing the order of values (shuffling).

3. At this step, we subtract the number of votes for the correct class in the variable-k-permuted data from the number of votes for the correct class in the original OOB data.

4. Now, the raw importance score for variable k is the average of this number over all trees in the forest. Then, we normalized the score by taking the standard deviation.

5. Variables having large values for this score are ranked as more important as building a current model without original values of a variable gives a worse prediction, which means the variable is important.

106. List down some of the shortcomings of the Random Forest Algorithm.

The shortcomings of the Random Forest algorithm are as follows:

1. Random Forests aren't good at generalizing cases with completely new data.

For Example, If we know that the cost of one ice cream is $1, 2 ice-creams cost $2, and 3 ice-creams cost $3, then how much do 10 ice-creams cost? In such cases, Linear regression models can easily figure this out, while a Random Forest has no way of finding the answer.

2. Random forests are biased towards the categorical variable having multiple levels or categories. It is because the feature selection technique is based on the reduction in impurity and is biased towards preferring variables with more categories so the variable selection is not accurate for this type of data.

107. List down the advantages and disadvantages of the Random Forest Algorithm.

Advantages:

- Random Forest is unbiased as we train multiple decision trees and each tree is trained on a subset of the same training data.
- It is very stable since if we introduce the new data points in the dataset, then it does not affect much as the new data point impacts one tree, and is pretty hard to impact all the trees.
- Also, it works well when you have both categorical and numerical features in the problem statement.
- It performs very well, with missing values in the dataset.

Disadvantages:

- Complexity is the major disadvantage of this algorithm. More computational resources are required and also results in a large number of decision trees combined together.
- Due to their complexity, training time is more compared to other algorithms.

108. What is the Random Forest Algorithm?
A random forest algorithm is an ensemble learning technique, which means it combines numerous classifiers to enhance a model's performance. In order to determine the output

depending on the input data, a random forest uses several decision tree (Classification and Regression Tree) models. It uses bagging to generate the decision trees it uses by using samples and randomly chosen features (variables).
Now, the modal (most frequent occuring) output among all the decision trees is picked as the one to move forward with when a random forest is used to forecast the outcome of a classification task. But if the goal is to forecast a regression problem's outcome, the mean (average) of all the decision trees' outputs is obtained before moving on.

109. For what applications are random forests used?
Numerous sectors have used the random forest algorithm to help them make better business decisions. Examples of use cases are:

1. Finance: This method is favoured over others since it takes less time to manage and pre-process data. It can be used to assess high-risk consumers for fraud and issues with option pricing.
2. Healthcare: The random forest approach is used in computational biology to solve issues including classifying gene expression, finding biomarkers, and annotating sequences (link resides outside IBM; PDF, 737 KB). Doctors can therefore estimate pharmacological reactions to certain drugs.
3. E-commerce: Cross-selling can be accomplished by using recommendation engines.

110. How does a Random Forest Work?
There are three key hyperparameters for random forest algorithms that must be set prior to training. Node size, tree count, and sampled feature count are a few of them. From there, classification or regression issues can be resolved using the random forest classifier. Each decision tree in the ensemble that makes up the random forest method is built of a data sample taken from a training set with replacement known as the bootstrap sample.

One-third of the training sample—also referred to as the out-of-bag (oob) sample—is set aside as test data; we'll return to this sample later. The dataset is subsequently given a second randomization injection by feature bagging, increasing dataset diversity and decreasing decision tree correlation.

The prediction will be determined differently depending on the type of issue.
The individual decision trees will be averaged for the regression job, and for the classification task, the predicted class will be determined by a majority vote, or the most common categorical variable.

The prediction is then finalized by cross-validation using the oob sample.

111. Why do we need the random forest algorithm?

It's a great question, and the response is simple. A random forest uses cross-validation to provide results that are more accurate and can be utilized for either classification or regression problems. Even missing values are handled, and a larger data collection with many features can be handled (dimensions). Most crucially, because it selects random features and generalizes well across the available data, it accomplishes all of this with minimal over-fitting. This makes it a lot more accurate algorithm than the traditional decision tree method, which is deficient in the majority of the aforementioned characteristics.

112. What are the advantages of the random forest methodology?

The random forest approach has a variety of significant benefits when applied to classification or regression tasks.

Some of them consist of:

Less chance of overfitting: Decision trees have a propensity to closely fit all the samples contained in training data, which increases the possibility of overfitting.

The classifier won't, however, overfit the model when there are a large number of decision trees in a random forest since the averaging of uncorrelated trees reduces the total variance and prediction error.

Flexibility: Random forest is a well-liked approach among data scientists since it can accurately handle both classification and regression jobs.

The random forest classifier benefits from feature bagging by maintaining accuracy even when some of the data is missing, which makes it a useful tool for guessing missing values.

Simple evaluation of feature contribution: Random forest makes it simple to evaluate variable contribution. There are various methods for determining feature relevance.

To gauge how much the model's accuracy declines when a particular variable is removed, the gini importance and mean drop in impurity (MDI) are frequently utilized. A different importance metric is permutation importance, often known as mean decrease accuracy (MDA). By randomly permuting the feature values in oob samples, MDA can determine the average reduction in accuracy.

113. What are the disadvantages of the random forest methodology?

The random forest approach has a variety of significant drawbacks when applied to classification or regression tasks.

Process that takes a long time: Because random forest methods can handle big data sets, they can make predictions that are more accurate. However, because they must compute data for each individual decision tree, they can take a long time to process data.

More resources are needed: Because random forests analyze bigger data sets, more resources are needed to store that data.

More complex: When compared to a forest of decision trees, a single one's prediction is simpler to understand.

114. How do random forests maintain accuracy when data is missing?
They often employ two techniques to accomplish this:

1. Since the available data is not being used, dropping or eliminating the data points with no values is not a desirable option.
2. Using the median to complete the missing values if they are numerical or the mode to complete the missing values if they are categorical. Even Nevertheless, there are limitations since occasionally there is insufficient data to paint an accurate picture.

115. What are ensemble methods?
Multiple models (commonly referred to as "weak learners") are taught to tackle the same problem using the ensemble learning paradigm, which then combines the findings to produce better ones. The basic claim is that by properly combining weak models, we can produce more precise and/or reliable models.

116. How do you improve random forest accuracy?
Depending on your aim and data. One of the numerous things you must take into account is the hyperparameter. You hardly ever obtain good accuracy if the parameter you are using is incorrect or irrelevant.
It may be feasible to identify a decent mix of parameters by manually adjusting them repeatedly, but I advise using methods like GridSearch or Bayesian optimization to do so.

117. Why is Random Forest Algorithm popular?
One of the most well-liked and frequently applied machine learning techniques for classification issues is Random Forest. Although it also works well with regression problem statements, the classification model is where it shines the most.
Modern data scientists now use it as a deadly weapon to improve the forecasting model. The algorithm's best feature is that it relies on a minimal number of assumptions, making data preparation simpler and faster.

118. Why Random Forest is not affected by outliers?
Tree methods, on the other hand, are "contributory" methods in which only local points, or those in the same leaf node, influence an estimate for a particular point. Outliers in the output have a "quarantined" effect. As a result, outliers that might drastically alter the precision of some algorithms have less of an impact on a Random Forest's forecast.

119. What are assumptions of random forest?

Some decision trees may forecast the correct output while others may not since the random forest combines numerous trees to predict the dataset class. But when all the trees are combined, they forecast the right result. Consequently, the following two presumptions for an improved Random forest classifier:

1. For the dataset's feature variable to predict true outcomes rather than a speculated result, there should be some actual values in the dataset.
2. Each tree's predictions must have extremely low correlations.

120. Difference Between Random Forest And Decision Tree

Here are some of the most significant distinctions between Random Forest and Decision Tree:

1. Data processing: The decision trees use an algorithm to decide on nodes and sub-nodes; a node can be split into two or more sub-nodes, and by creating sub-nodes it gives another homogeneous sub-node, so we can say that the nodes have been split. The random forest is the combination of multiple decision trees, which is the class of dataset. Some decision trees out of it may give the correct output and others may not give it correctly, but all trees together predict

   The split is carried out using the best data that can be used initially, and the operation has been repeated until all of the child nodes have reliable data.
2. Complexity: The decision tree, which is used for both classification and regression, is a straightforward series of choices that are taken to obtain the desired outcomes. The advantage of the simple decision tree is that this model is simple to interpret, and when building decision trees, we are aware of which variable and what is the value of the variable is using to split the data, and as a result, the output will be predicted quickly. In contrast, the random forest is more complex because it combines decision trees, and when building a random forest, we have to define the number of trees we want to build and how many variables we need.
3. Accuracy: Compared to decision trees, random forest forecasts outcomes with more accuracy. We can also say that random forests build up many decision trees and that combines together to give a stable and accurate result. When we are using an algorithm to solve the regression problem in a random forest, there is a formula to get an accurate result for each node, whereas the accuracy of results is increased by using a combination of learning models.

4. Overfitting: When using algorithms, there is a risk of overfitting, which can be viewed as a general bottleneck in machine learning. Overfitting is the important issue in machine learning.

   When machine learning models are unable to perform well on unknown datasets, this is a sign of overfitting. This is especially true if the error is found on the testing or validation datasets and is significantly larger than the error on the training dataset. Overfitting occurs when models learn fluctuation data in the training data, which has a negative impact on the performance on the new data model.

   Due to the employment of several decision trees in the random forest, the danger of overfitting is lower than that of the decision tree.

   The accuracy increases when we employ a decision tree model on a given dataset since it contains more splits, which makes it easier to overfit the data and validate it.

121. How feature importance is calculated in random forest?

There are two techniques to calculate the feature importance integrated into the Random Forest algorithm:

The mean decrease impurity, also known as Gini significance, is calculated from the Random Forest structure.

Let's examine the Random Forest's architecture.

It consists of many Decision Trees.

There are internal nodes and leaves in every decision tree.

The internal node uses the chosen characteristic to decide how to split the data set into two distinct sets with related responses.

The criteria used to choose the features for internal nodes might range from gini impurity or information gain for classification tasks to variance reduction for regression activities. We can gauge how each feature reduces the split's impurity (the feature with highest decrease is selected for internal node).

We can track the average reduction in impurity for each feature.

The average of all the trees in the forest serves as a proxy for the significance of a feature.

This technique is offered in the Scikit-Learn Random Forest implementation (for both classifier and regressor).

The relative values of the computed importances should be considered when using this method, it is important to note.

The major benefit of this approach is computation speed; all necessary values are computed while the trainees are in Radom Forest. The method's shortcomings include its propensity to favor (select as significant) numerical features and categorical features with high cardinality. Additionally, when associated features are present, it may choose one trait while ignoring the significance of the other (which can lead to wrong conclusions).

Based on the mean drop in accuracy, the Mean Decrease Accuracy approach computes the feature importance on permuted out-of-bag (OOB) data. The scikit-learn package does not include this method's implementation.

122. Does Random Forest need Pruning? Why or why not?
Very deep or fully-depth decision trees have a tendency to pick up on the data noise. They overfit the data, resulting in large variation but low bias. Pruning is an appropriate method for reducing overfitting in decision trees.

However, in general, full-depth random forests would do well. The correlation between the trees (or weak learners) would be poor since random forests training uses bootstrap aggregation (or sampling with replacement) together with random selection of features for a split. Because the trees are not coupled, even though each individual tree would have a high variance, the ensemble output would be suitable (lower variance and reduced bias).

123. Explain how the Random Forests give output for Classification, and Regression problems?
Classification: The Random Forest's output is the one that the majority of trees have chosen.
Regression: The mean or average forecast of each individual tree is the Random Forest's output.

124. How is it possible to perform Unsupervised Learning with Random Forest?
The goal of a decision tree model is to arrive at one of the predetermined output values by starting with some input data and proceeding through a sequence of if-then stages.
In contrast, a random forest model combines a number of decision trees that have been trained using different subsets of the starting data.
For instance, you might gather information on how frequently consumers have visited the bank in the past and what services they have used if you wanted to utilize a single decision tree to predict how frequently specific customers would use a specific service offered by the bank. You would identify particular qualities that affect the customer's decision. The decision tree would produce rules that would enable you to forecast whether a consumer will use the services or not.

If you input the same data into a random forest, the algorithm will produce multiple trees from a set of customers that are chosen at random. The sum of the individual outcomes of each of those trees would be the forest's output.

125. How would you improve the performance of Random Forest?
Attempting the following measures may help Random Forest function better:

1. Utilizing feature engineering and a dataset of greater quality. It is not desirable for the model to use too many features and data, hence it is occasionally necessary to perform some feature reduction.

2. Adjusting the algorithm's hyperparameters.
3. Trying several algorithms

126.What are proximities in Random Forests?

The word "proximity" refers to the similarity or closeness of two cases.
For each pair of cases, observations, or sample points, proximity is determined. Two cases are closer together if they share the same terminal node in a single tree. The proximities are normalized by dividing by the total number of trees at the conclusion of the run of all trees. In order to replace missing data, find outliers, and create enlightening low-dimensional perspectives of the data, proximity is used.

127. What does Random refer to in Random Forest?

In the process of choosing the features for each tree, randomness is introduced.
For each tree formed, the algorithm chooses a random set of features to split a node from rather than looking for the most crucial trait.
As a result, the model can be more flexible, factor chaos, and have a less deterministic structure, which improves out-of-sample performance while lowering the danger of overfitting.

128. What is Out-of-Bag Error?

OOB (out-of-bag) error
A method of calculating the prediction error that enables the Random Forest to be fitted and validated while being trained is known as out-of-bag (OOB) error/out-of-bag estimate.
Let's use the following illustration to better comprehend that:
Assume that each of the 20 students is given a different subset of the data (i.e. as seen in Stages 1, and 2). An observation may be contained within the subsets of more than one learner because we are sampling with replacement. Therefore, we'll calculate the Out-of-bag in the following way:

1. Repeat after each observation
2. Determine the students who did not previously have this observation in their subset for each observation.
3. Use the students you specified above to carry out the prediction for this observation.
4. The prediction error for each observation is averaged

   This allows us to assess the predictions made based on the observations that weren't used to train the basic learners. This also eliminates the requirement for any validation set.

129. What is Entropy?
Entropy, often known as Shannon Entropy, is a measure of how random or uncertain the data is for a finite collection S. It is denoted by H(S).

$$H(S) = \sum_{x \in X} p(x) log_2 \frac{1}{p(x)}$$

Simply put, it gauges purity to foretell a specific event. The root node serves as the foundation for the top-down construction of the decision tree. This root node's data is further divided or categorized into subsets with homogeneous instances.

130. Why Random Forest models are considered not interpretable?
Decision trees can readily be transformed into rules that improve the results' human interpretability and provide an explanation for the choices taken.
The usual guideline is to employ as many trees as you can for Random Forest. Most of the time, it would be difficult to comprehend why a group of hundreds of trees decided as they did.

131. Why is the training efficiency of Random Forest better than Bagging?
For Random Forest, just a subset of all features are randomly chosen, and the best split feature from the subset is used to divide each node in a tree. This is how Random Forest differs from Bagging. All features are taken into account when separating a node while bagging.

Because bagging takes into account all of the attributes, random forest has a superior training efficiency.

132. Explain how it is possible to get feature importance in Random Forest using Out Of Bag Error
The Out of Bag Error is used in the following technique to determine feature importance:
Without making any changes, generate the out of bag error for the base tree.
The standard out of bag error is this. Determine the range of values that each characteristic in the data used to construct the tree can be. Specifically, what are the data's maximum and minimum values? Randomly permute the values of each feature in all the data points between its maximum and minimum for each feature, one at a time. Find the out of bag mistake next after only permuting one feature.
The difference between the current out-of-bag error and the previous out-of-bag error should be calculated. It is almost certain that the new out of bag error will be higher than the mean value. Restart the procedure by changing a new feature after restoring the permuted features to their original values. The most crucial features are those whose inaccuracy increases the

most when they are permuted. The least significant features are those with the smallest increase in inaccuracy.

This approach can be compared to withholding individual machine pieces from someone but allowing them to assemble the entire machine. The part must be crucial if the assembly fails catastrophically. The parts must not be very crucial if you can't tell the difference after the machine has been created.

133.Give some reasons to choose Random Forests over Neural Networks

In terms of processing cost, Random Forest is less expensive than neural networks. A GPU might be required to finish training neural networks.
In comparison to random forest, neural networks require a lot more information.
Neural networks operate like a "black box," where the creator is unable to see how the output is determined. A vast number of trees in Random Forest might make it challenging to understand the ensemble even though each tree is distinct and easily understood. The most representative trees in an ensemble can be found in a number of different methods. The Random Forest is therefore superior in this sense.
When given unusual inputs, neural networks frequently have a higher risk of deviating from the expected behavior.

134. How can you tell the importance of features using Random Forest?

The ability to determine the relative value of various features in the data is one of the most intriguing things that can be done using Random Forest.
To see how different features affect the result, the various features and the output can be displayed and evaluated for outliers.
Using the information gain at each stage of the tree to calculate the relative information gain between various features is one way to gauge the significance of a feature.
The algorithm's steps are provided in the list below:

1. Start with an array that is zero-filled and the same size as the number of features in the model.
2. Using the information that was used to create the tree, traverse it.
   When you reach a branch in the tree, identify the feature that branch relied on.
3. Every branch divides on a single, unique trait. Instead of going down a separate branch of the tree, count the number of data points that reach each one.
4. Instead of before the branch, the knowledge gain is calculated after the fork.
   This is true regardless of how information gain is measured (e.g. Gini, Entropy).
5. The information gain is multiplied by the quantity of data points that reach the branch and is added to the array at the feature where the data points are being separated.
6. The array is normalized after the information gain for each branch has been added.

7. The calculation of the feature importance for each tree in the Random Forest is then repeated for each tree in turn. The feature importance of the entire Random Forest is then calculated by averaging it.

135. How to use Isolation Forest for Anomalies detection?

Similar to Random Forests, Isolation Forests (IF) are constructed using decision trees. Additionally, this model is unsupervised because there are no predefined labels present. The foundation of IsolationForests is the idea that anomalies are the "few and different" data items. Randomly subsampled data is processed in an isolation forest using a tree structure based on randomly chosen attributes. Since it took more cuts to isolate the samples that traveled further into the tree, they are less likely to be abnormalities.

The following steps are involved in creating or training an isolation tree given a dataset:

1. Pick a random subset of the information
2. Prior to isolating each point in the dataset:
3. Choosing a feature one at a time
4. Divide the feature at a chance location within its range.
5. An outlier point can be separated in a small number of divisions, as seen in the image below, whereas an interior point necessitates more partitions.

   Given a fresh point, the process of prediction entails:
6. Perform a binary search across each itree in the forest, traveling until you find a leaf.
7. Using the depth of the path to the leaf as the basis, calculate the anomaly score.
8. Create an overall anomaly score for the point by adding the anomaly scores from each of the separate itrees.

   In general, interior points will have a substantially longer trip to the leaf than anamolous points, making them more difficult to isolate.

136. What do you mean by Bagging?

The ensemble learning technique known as bagging, often referred to as bootstrap aggregation, is frequently used to lessen variance within a noisy dataset. In bagging, a training set's data is randomly sampled and replaced, allowing for multiple selections of the same data points. Following the generation of several data samples, these weak models are independently trained, and depending on the task—for example, classification or regression—the average or majority of those predictions result in a more accurate estimate. The random forest algorithm, which makes use of both feature randomness and bagging to build an uncorrelated forest of decision trees, is thought of as an extension of the bagging method.

137. When should you use another algorithm over a random forest?

Despite being extremely quick to train, ensembles of decision trees (such as Random Forests, a name that has been registered for one specific implementation) take a long time to provide predictions after training.

The model must be used with more trees because more accurate ensembles require more trees. This method is quick in the majority of real circumstances, but there may be some where run-time performance is crucial and other methods are favored.

It is crucial to understand that this tool is for predictive modelling and not for descriptive purposes; if you need a description of the relationships in your data, you should explore other solutions.

Ensembles of decision trees are perhaps the most practical tool now available for quick, easy, adaptable predictive modeling, but like any method, they are not without flaws.