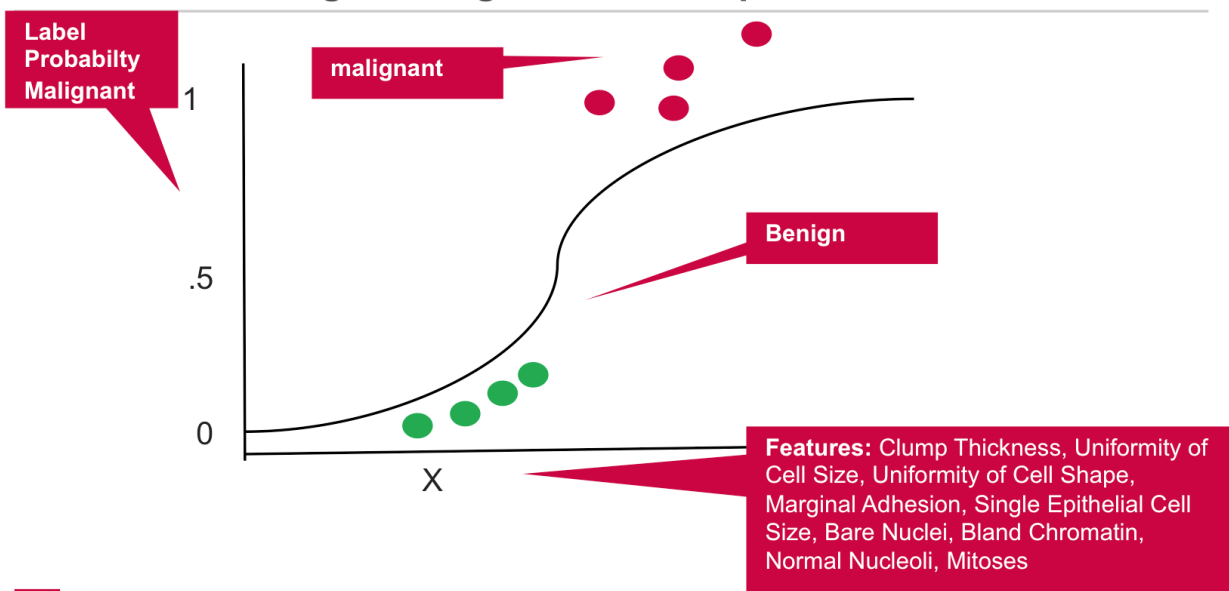**Q 1. What is Logistic Regression?**

Logistic Regression is a supervised learning classification algorithm. It is used when a dependent variable or target is categorical.

For example,
- To predict whether an email is spam (1) or not (0)
- Whether the tumor is malignant or benign

Breast Cancer Logistic Regression Example

Logistic regression can be binomial, ordinal or multinomial
.
- Binomial or binary logistic regression deals with situations in which the observed outcome for a dependent variable can have only two possible types, "0" and "1" (which may represent, for example, "dead" vs. "alive" or "win" vs. "loss").
- Multinomial Logistic Regression deals with situations where the outcome can have three or more possible types (e.g., "disease A" vs. "disease B" vs. "disease C") that are not ordered.
- Ordinal Logistic Regression deals with dependent variables that are ordered.

Q 2. What is a Decision Tree?

Decision Tree is a supervised learning algorithm that can be used for both classification and regression problems, but mostly it is preferred for solving classification problems.

In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

 A decision tree simply asks a question, and based on the answer (Yes/No), it further splits the tree into subtrees.

Below diagram explains the general structure of a decision tree:
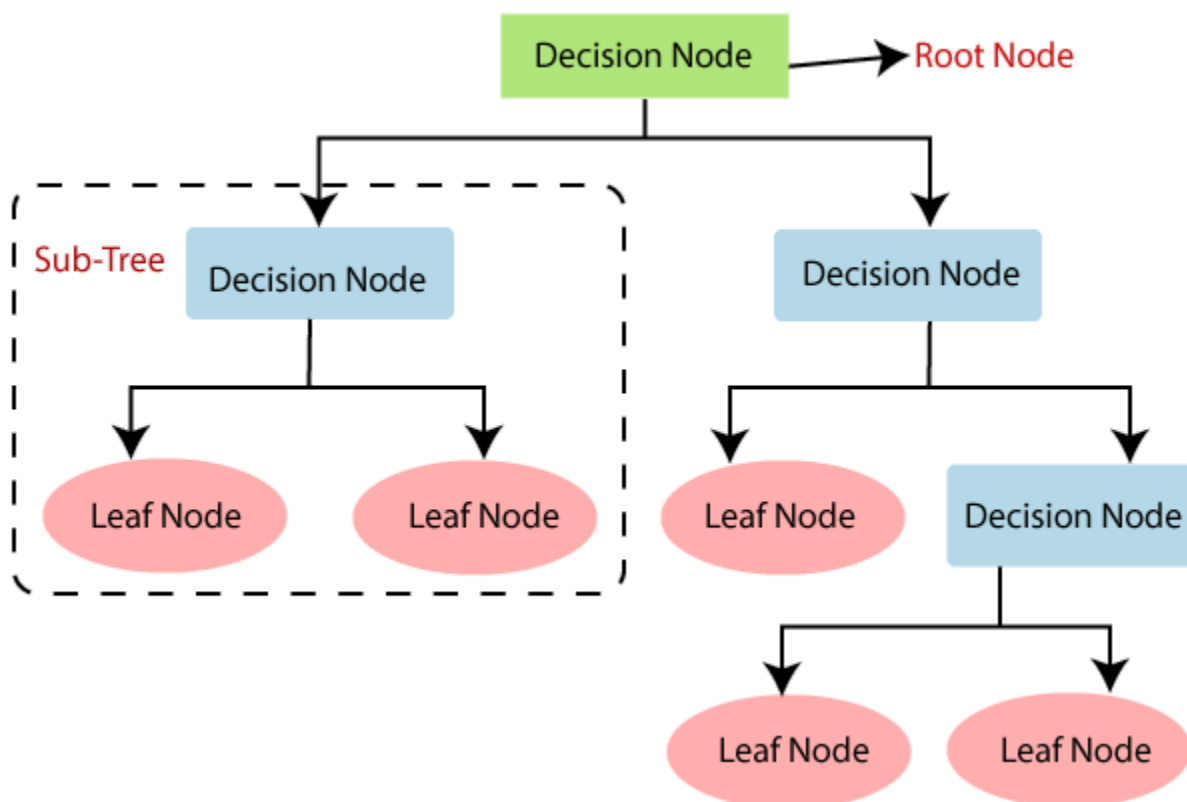
Q 3. How does the decision tree algorithm work ?

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of the root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.

- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**

- **Step-3:** Divide the S into subsets that contain possible values for the best attributes.

- **Step-4:** Generate the decision tree node, which contains the best attribute.

- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

Q 4. Give a real life example of a decision tree algorithm.

Example: Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer).

Consider the below diagram:
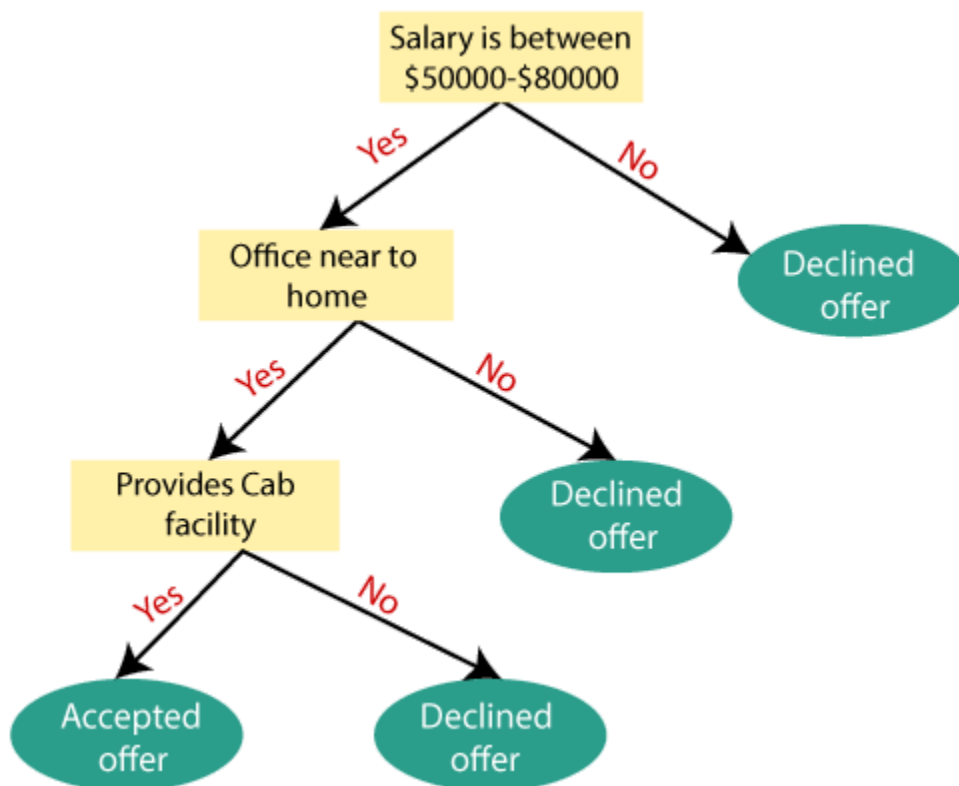
Q 5. What are the advantages and disadvantages of the decision tree?

Advantages:

- It is simple to understand as it follows the same process which a human follows while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Disadvantages:

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the Random Forest algorithm.
- For more class labels, the computational complexity of the decision tree may increase.

Q 6. What is Random forest Algorithm?

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Q 7. How does the Random forest algorithm work ?

Random Forest works in two-phase first is to create the random forest by combining N decision trees, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.
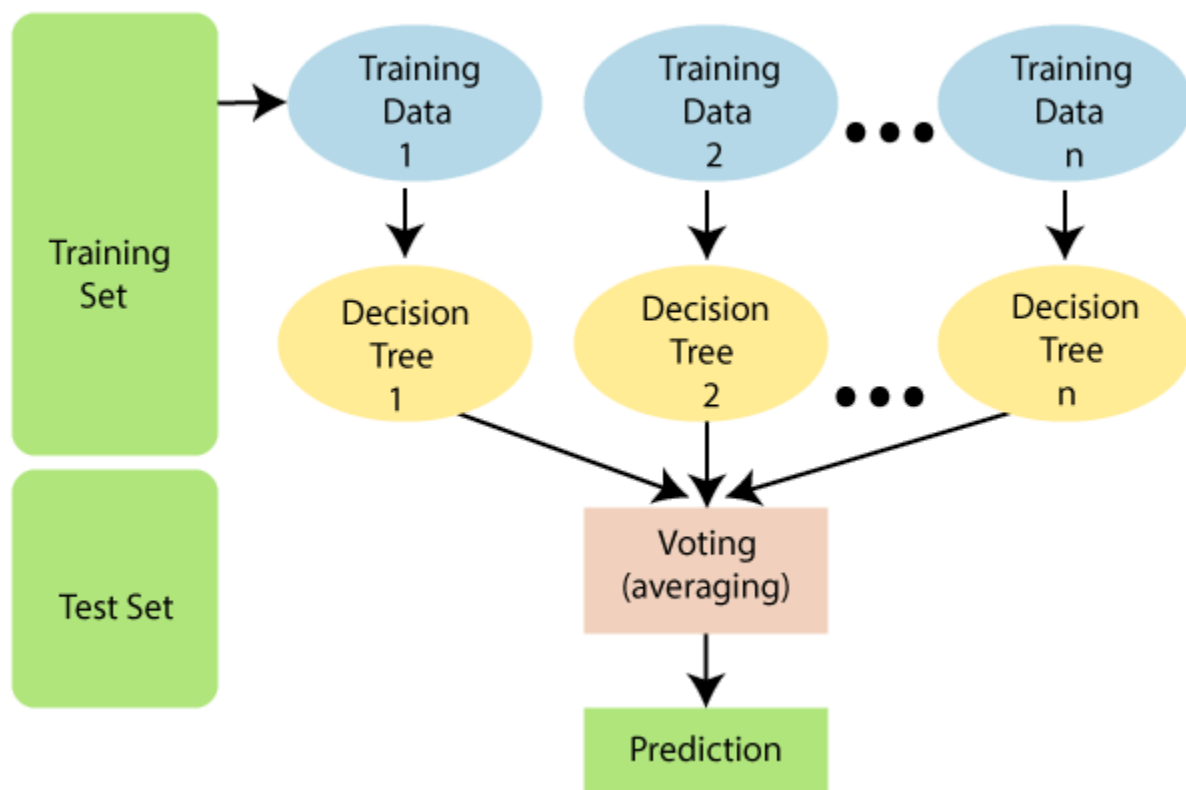
Q 8. Give an example of a random forest algorithm for better understanding.

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:
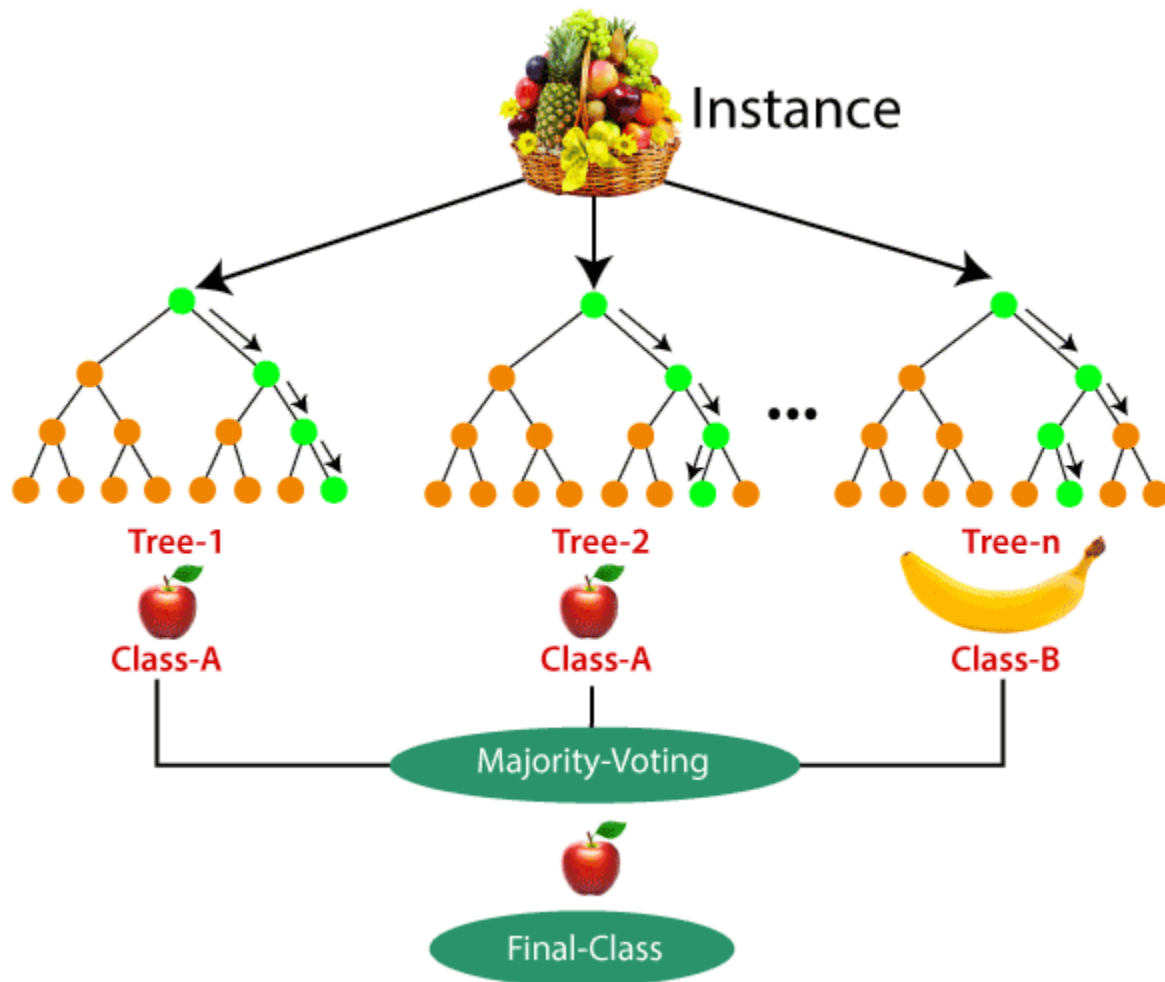
Q 9. What is data preprocessing?

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put it in a formatted way. So for this, we use data preprocessing tasks.

It involves below steps:

- **Getting the dataset**

- **Importing libraries**

- **Importing datasets**

- **Finding Missing Data**

- **Encoding Categorical Data**

- **Splitting dataset into training and test set**

- **Feature scaling**

Q 10. What is a confusion matrix ?

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an error matrix. Some features of Confusion matrix are given below:

- For the 2 prediction classes of classifiers, the matrix is of 2*2 table, for 3 classes, it is 3*3 tables, and so on.

- The matrix is divided into two dimensions, that are predicted values and actual values along with the total number of predictions.

- Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.

- It looks like the below table:

n = total predictions	Actual: No	Actual: Yes
Predicted: No	True Negative	False Positive
Predicted: Yes	False Negative	True Positive

The above table has the following cases:

- True Negative: Model has given prediction No, and the real or actual value was also No.

- True Positive: The model has predicted yes, and the actual value was also true.

- False Negative: The model has predicted no, but the actual value was Yes, it is also called Type-II error.

- False Positive: The model has predicted Yes, but the actual value was No. It is also called a Type-I error.

Q 11. What are the needs of a confusion matrix in machine learning?

Needs of a confusion matrix are as follows:

- It evaluates the performance of the classification models, when they make predictions on test data, and tells how good our classification model is.
- It not only tells the error made by the classifiers but also the type of errors such as it is either type-I or type-II error.
- With the help of the confusion matrix, we can calculate the different parameters for the model, such as accuracy, precision, etc.