1. R-squared or Residual Sum of Squares (RSS) which one of these two is a better measure of goodness of fit model in regression and why?

Ans = R-squared and Residual Sum of Squares (RSS) are both commonly used measures of goodness of fit for regression models. However, they measure different aspects of the model performance.

Overall, R-squared is a more comprehensive measure of model performance, while RSS is a more specific measure of the amount of unexplained variance in the dependent variable.

2. What are TSS (Total Sum of Squares), ESS (Explained Sum of Squares) and RSS (Residual Sum of Squares) in regression. Also mention the equation relating these three metrics with each other.

Ans = In regression analysis, Total Sum of Squares (TSS), Explained Sum of Squares (ESS), and Residual Sum of Squares (RSS) are three important metrics used to evaluate the goodness of fit of a model.

TSS is the total variation in the dependent variable. It represents the total sum of squared differences between the observed values of the dependent variable and the mean of the dependent variable. TSS is calculated using the following equation:

$$TSS = \Sigma(y - \bar{y})^2$$

where y is the observed value of the dependent variable, $\bar{y}$ is the mean of the dependent variable, and Σ represents the sum over all observations.

ESS is the explained variation in the dependent variable. It represents the sum of squared differences between the predicted values of the dependent variable and the mean of the dependent variable. ESS is calculated using the following equation:

$$ESS = \Sigma(y\_pred - \bar{y})^2$$

where y_pred is the predicted value of the dependent variable based on the model, and $\bar{y}$ is the mean of the dependent variable.

RSS is the residual variation in the dependent variable. It represents the sum of squared differences between the observed values of the dependent variable and the predicted values of the dependent variable. RSS is calculated using the following equation:

$$RSS = \Sigma(y - y\_pred)^2$$

where y is the observed value of the dependent variable, y_pred is the predicted value of the dependent variable based on the model.

These three metrics are related to each other through the following equation:

TSS = ESS + RSS

3. What is the need of regularization in machine learning?

Ans = Regularization is a technique in machine learning that is used to prevent overfitting of a model. Overfitting occurs when a model is trained to fit the training data so closely that it becomes too specific to that data, and it does not generalize well to new, unseen data. Regularization techniques add a penalty term to the loss function of the model that discourages the model from learning complex relationships in the training data that may not generalize well to new data. regularization is an important technique in machine learning that can improve the performance and generalization of a model, and it is particularly useful when dealing with complex models with many parameters.

4. What is Gini–impurity index?

Ans = Gini-impurity index is a measure of impurity or diversity used in decision tree algorithms and other machine learning models that use decision trees. It is a measure of how often a randomly chosen element from a dataset would be incorrectly classified if it were randomly labeled according to the distribution of labels in the subset. The Gini-impurity index is often used in decision tree algorithms to select the best feature for splitting the dataset at each node. The feature that results in the greatest reduction in the Gini-impurity index is chosen as the splitting criterion, as it maximizes the homogeneity of the resulting subsets.

Gini-impurity index = 1 - ∑(pi)^2

5. Are unregularized decision-trees prone to overfitting? If yes, why?

Ans = Yes, unregularized decision trees are prone to overfitting because they have the capacity to create very complex trees that perfectly fit the training data, even if the underlying patterns are noisy or irrelevant. As a result, an unregularized decision tree can become overly specific to the training data and not generalize well to new, unseen data

unregularized decision trees have the potential to overfit the data, and regularization techniques are important to prevent overfitting and improve the performance and generalization of the model.

6. What is an ensemble technique in machine learning?

Ans = In machine learning, an ensemble technique is a method of combining multiple individual models to improve the overall performance and accuracy of a machine learning system. Ensembles are used to reduce overfitting, increase stability, and improve generalization.

There are several types of ensemble techniques, including:

1.Bagging

2.Boosting

3.Stacking

4.Random Forest

7. What is the difference between Bagging and Boosting techniques?

Ans = The main difference between Bagging and Boosting is in how the individual models are trained and combined:

1.Bagging (Bootstrap Aggregating) involves training multiple models on different subsets of the training data, using a technique called bootstrapping. Each model is trained independently of the other models, and the final output is obtained by averaging the outputs of all models. Bagging is particularly effective at reducing the variance of the model and improving the stability and generalization of the model.

2.Boosting involves training multiple models sequentially, with each model attempting to correct the errors of the previous model. Each subsequent model is trained on the same training set as the previous models, but with a different weighting assigned to each training example. Boosting is particularly effective at reducing the bias of the model and improving the accuracy and performance of the model.

8. What is out-of-bag error in random forests?

Ans = Out-of-bag (OOB) error is a way of estimating the performance of a random forest model without the need for a separate validation set. In a random forest, each decision tree is trained on a bootstrapped sample of the training data, which means that some data points are left out of the sample and not

used in the training of the tree. These left-out data points are referred to as the out-of-bag (OOB) data.

9. What is K-fold cross-validation?

Ans = K-fold cross-validation is a technique used to evaluate the performance of a machine learning model by partitioning the data into K subsets (or folds) of roughly equal size. The model is trained on K-1 of the folds and tested on the remaining fold, and this process is repeated K times such that each fold is used as the test set exactly once.

10. What is hyper parameter tuning in machine learning and why it is done?

Ans = In machine learning, hyperparameter tuning is the process of finding the optimal combination of hyperparameters that results in the best performance of a machine learning algorithm on a given task.

Hyperparameter tuning can be done using various techniques, such as grid search, random search, and Bayesian optimization. Grid search involves systematically testing all possible combinations of hyperparameters within a predefined range, while random search involves randomly sampling hyperparameters from a predefined distribution. Bayesian optimization uses probabilistic models to guide the search for the best hyperparameters.

11. What issues can occur if we have a large learning rate in Gradient Descent?

Ans = If we use a large learning rate in gradient descent, it can lead to several issues, including:

1. Divergence.

2. Slow Convergence.

3. Overshooting the minimum.

4. Unstable gradients.

12. Can we use Logistic Regression for classification of Non-Linear Data? If not, why?

Ans = Logistic regression is a linear classifier, which means it can only classify data that is linearly separable. Linearly separable data is data that can be separated into different classes by a straight line or a hyperplane in the feature space.

Alternatively, non-linear classifiers such as decision trees, random forests, or support vector machines (SVMs) can be used for non-linear classification tasks.

In summary, logistic regression is a linear classifier that may not be suitable for classification of non-linear data. However, we can use techniques such as feature engineering or kernel methods to transform the data into a higher-dimensional feature space, or we can use non-linear classifiers such as decision trees, random forests, or SVMs to handle non-linear classification tasks.

13. Differentiate between Adaboost and Gradient Boosting.

Ans = Adaboost and Gradient Boosting are two popular boosting techniques used for ensemble learning in machine learning. The main differences between these two techniques are as follows:

1. Weighting of Samples
2. Training of Models
3. Gradient Calculation
4. Weighting of Models
5. Handling of Outliers

Adaboost and Gradient Boosting are both boosting techniques used for ensemble learning, but they differ in the way they assign weights to samples, train models, calculate gradients, weight models, and handle outliers. Adaboost assigns weights to samples and models based on their performance, trains weak classifiers sequentially, and uses the first derivative of the loss function to calculate the gradient. Gradient Boosting assigns equal weights to samples, trains a sequence of models, uses the second derivative of the loss function to calculate the gradient, and uses a robust loss function to handle outliers.

14. What is bias-variance trade off in machine learning?

Ans = The bias-variance tradeoff is a fundamental concept in machine learning that describes the relationship between the complexity of a model and its ability to generalize to new data.

Bias refers to the error that arises due to the model's simplifying assumptions about the underlying data. A model with high bias tends to underfit the data and may not capture the underlying patterns, resulting in poor performance on both the training and test data.

Variance refers to the error that arises due to the model's sensitivity to the noise in the training data. A model with high variance tends to overfit the training data and may not generalize well to new data.

15. Give short description each of Linear, RBF, Polynomial kernels used in SVM.

Ans = Support Vector Machines (SVMs) are a popular class of supervised machine learning algorithms used for classification and regression tasks. SVMs use a kernel function to map the input data into a higher-dimensional feature space, where the data can be more easily separated.

The three commonly used kernels in SVMs are:

1.Linear Kernel

2. RBF Kernel

3. Polynomial Kernel

the linear kernel is suitable for linearly separable data, the RBF kernel is suitable for non-linearly separable data, and the polynomial kernel is suitable for data that can be separated

using a polynomial boundary. The choice of kernel depends on the nature of the data and the problem at hand.