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?**
   R-squared is generally a better measure of the goodness of fit for a regression model than the residual sum of squares (RSS).

   R-squared is a statistical measure that represents the proportion of the variance for the dependent variable that's explained by the independent variables in the model.

   The Residual Sum of Squares (RSS) measures the total squared difference between the observed values of the dependent variable and the values predicted by the model. While RSS gives an indication of how well the model fits the data in terms of reducing errors (residuals), it does not provide a standardized measure like R-squared that accounts for the overall variance explained.

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.**

   A.Total Sum of Squares (TSS): TSS measures the total variance in the observed data. It is the sum of the squared differences between each observed value and the overall mean of the observed values.

   $TSS = \sum(yi - \bar{y})2$

   where yi is the i-th observed value and $\bar{y}$ is the mean of the observed values.

   B.Explained Sum of Squares (ESS): ESS measures the portion of the total variance that is explained by the regression model. It is the sum of the squared differences between the predicted values (from the regression model) and the mean of the observed values.

   $ESS = i = \sum(zi - \bar{y})2$

   where zi is the i-th predicted value from the regression model and $\bar{y}$ is the mean of the observed values.

   C.Residual Sum of Squares (RSS): RSS measures the portion of the total variance that is not explained by the regression model. It is the sum of the squared differences between the observed values and the predicted values.

   $RSS = i = \sum(yi - zi)2$

   yi is the i-th observed value and zi is the i-th predicted value from the regression model

   Relationship Between TSS, ESS, and RSS

   TSS=ESS+RSS
   This equation indicates that the total variance in the observed data (TSS) is equal to the sum of the variance explained by the model (ESS) and the variance not explained by the model (RSS).

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

While training a machine learning model, the model can easily be overfitted or underfitted. To avoid this, we use regularization in machine learning to properly fit a model onto our test set. Regularization techniques help reduce the chance of overfitting and underfitting and help us get an optimal model

4.What is Gini–impurity index?

Gini Impurity is a measurement used to build Decision Trees to determine how the features of a dataset should split nodes to form the tree. It's a tool that helps decision trees decide how to split up the information they're

given. It is a number between 0-1 where 0 represents purity of the classification and 1 denotes random distribution of elements among various classes.

A Gini Index of 0.5 shows that there is equal distribution of elements across some classes.
A Gini Impurity of 0 is the lowest and the best possible impurity for any data set.
Gini Impurity is calculated by subtracting the sum of the squared probabilities of each class from one.

5. Are unregularized decision-trees prone to overfitting? If yes, why?
Unregularized decision trees tend to overfit because they can create highly complex models that capture the noise in the training data, leading to poor generalization on new data. Applying regularization techniques such as limiting the tree's depth, setting minimum samples for splits and leaves, limiting the number of leaf nodes and pruning can help control the complexity of the tree and improve its performance on unseen data.

6. What is an ensemble technique in machine learning?

Ensemble techniques in machine learning involve combining the predictions of multiple models to produce a more robust and accurate prediction than any single model could achieve on its own.
In this the multiple models are trained to solve the same problem and combined to get better results.
Types of Ensemble techniques:
1.Bagging-It considers homogeneous weak learners ,learns them independently from each other in parallel and combines them following some kind of deterministic averaging process i.e. it involves training multiple instances of the same model on different subsets of the training data
Eg. RandomForest
2,Boosting-it considers homogeneous weak learners , learn them sequentially in a very adaptive way(a base model depends upon the previous one) and combines them following a deterministic strategy i.e. it involves training models sequentially, where each model attempts to correct the errors of the previous one. Eg AdaBoost, GradientBoost
3.Voting:- Voting involves training multiple models (heterogeneous models)and combining their predictions through a majority vote (for classification) or averaging (for regression).Eg Hard voting, Soft Voting

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

Bagging-It considers homogeneous weak learners ,learns them independently from each other in parallel and combines them following some kind of deterministic averaging process i.e. it involves training multiple instances of the same model on different subsets of the training data
Eg. RandomForest
Boosting-it considers homogeneous weak learners , learn them sequentially in a very adaptive way(a base model depends upon the previous one) and combines them following a deterministic strategy i.e. it involves training models sequentially, where each model attempts to correct the errors of the previous one. Eg AdaBoost, GradientBoost

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

Out-of-bag error in Random Forest is a practical method to estimate how well the model is performing without requiring additional data for validation, making it a valuable tool in the evaluation of ensemble models. Out-of-bag (OOB) error is an estimate of the prediction error of a Random Forest model. It is calculated by averaging the prediction error on each training sample, where the prediction is made by the trees that do not include the sample in their respective bootstrap training sets.

9. What is K-fold cross-validation?

In this the original dataset is equally partitioned into k subparts or folds.Out of the k folds,for each iteration,one group is selected as validation data and the remaining (k-1) groups are selected for training data.The process is repeated for k times until each group is treated as validation and remaining as training data.The final accuracy of model is calculated by taking the mean accuracy of the k-models validation data.

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

Hyperparameter tuning in machine learning refers to the process of selecting the optimal set of hyperparameters for a machine learning algorithm. Hyperparameters are the configuration settings used to structure and train the

model, which are set before the learning process begins and are not learned from the data. They differ from model parameters, which are learned during training.
It is done in order to achieve:
1.Improving model performance
2.Balancing Bias-Variance trade off
3.Ensuring robustness
4.Optimizing computational efficiency
Different algorithms have different hyperparameters.Eg:

1. Kernel Parameters (for SVMs)
2. Criterion (for DecisionTreesClassifier)
3. Alpha value(For Lasso and Ridge)

There are 2 methods to do so:
1.GradientSearchCV()-It checks the value for each parameter sent in a dictionary
2.RandomizedSearchCV()-It randomly chooses the value from the parameter passed in dictionary and gives the result.Therefore, it is faster than GradientSearchCV().

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

A large learning rate can cause significant issues in gradient descent optimization, including overshooting, divergence, oscillations, instability, and poor model performance. It is crucial to choose an appropriate learning rate or employ strategies like learning rate scheduling or adaptive learning rates to ensure effective training and convergence of the model.

1. Overshooting:

With a large learning rate, the updates to the model parameters can be too large, causing the optimization process to overshoot the minimum of the cost function. Therefore, instead of converging to the minimum, the algorithm may oscillate around it or even diverge, never settling down to an optimal value.

2. Divergence:

If the learning rate is excessively large, the updates might push the parameters far away from the optimal solution, potentially increasing the cost function value.This can lead to the gradient descent process diverging, where the cost function continues to increase without bound, and the model fails to learn anything meaningful.

3. Oscillations:

A large learning rate can cause the parameters to jump back and forth across the slopes of the cost function without making steady progress toward the minimum.This results in oscillatory behavior, preventing the algorithm from converging to a stable solution.

4. Instability:

The optimization process becomes unstable with a large learning rate because the updates can vary wildly in magnitude and direction.This instability can make it difficult to track the progress of training and understand how close the model is to an optimal solution.

5. Poor Model Performance:

Due to the inability to converge to an optimal set of parameters, the model trained with a large learning rate might end up with suboptimal weights.This results in poor performance on the training data, and consequently, the model will generalize poorly to new, unseen data.

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

Logistic Regression is inherently a linear classifier, meaning it assumes that the relationship between the input features and the output (binary) class is linear. The Logistic Regression is typically not suitable for classifying non-linear data:

1.Linear Decision Boundary:

Logistic Regression models the probability of the binary outcome using a sigmoid function, which results in a linear decision boundary in the input feature space.This linear boundary can only separate classes that are linearly separable. If the relationship between the features and the classes is non-linear, Logistic Regression cannot capture this complexity.

2.Limited Flexibility:

The model's linearity restricts its ability to model complex interactions and non-linear relationships between features.It can underfit when trying to fit data that exhibits non-linear patterns, leading to poor classification performance.

3.Assumption Violation:
Logistic Regression assumes that the log-odds of the output variable are a linear combination of the input features. If this assumption is violated (i.e., if the relationship is non-linear), the model's predictions may be inaccurate.
4.Suboptimal Performance on Non-Linear Data:
When applied to non-linear data, Logistic Regression may fail to learn and generalize well, resulting in low accuracy and poor performance metrics such as precision, recall, and F1-score.

13. Differentiate between Adaboost and Gradient Boosting

Feature	AdaBoost (Adaptive Boosting)	Gradient Boosting
1.Main Idea	Focuses on improving weak learners by adjusting weights of misclassified instances.	Sequentially builds models to correct residual errors from previous models.
2.Boosting Method	Adjusts weights of training instances based on misclassification.	Fits new models to the gradient (residual errors) of the loss function.
3.Weight Adjustments	Increases weights of misclassified instances; decreases weights of correctly classified ones.	No explicit weight adjustment; instead, the new model targets residuals.
4.Weak Learners	Typically uses decision stumps (one-level decision trees).	Typically uses deeper decision trees.
5.Combination of Learners	Weighted sum of weak learners based on their accuracy.	Weighted sum of weak learners, scaled by a learning rate.
6.Error Handling	Emphasizes correcting errors from the previous model by focusing on hard-to-classify instances.	Directly minimizes the residual error in each iteration using gradient descent.
7.Learning Rate	Not typically a parameter, but the influence of each learner is implicitly managed by instance weights.	Explicitly involves a learning rate to scale the contribution of each learner.
8.Iteration Approach	Sequential; each learner is trained with adjusted instance weights.	Sequential; each learner is trained on the residuals of the previous ensemble.
9.Sensitivity to Noise	Higher sensitivity due to focus on misclassified instances.	Generally lower sensitivity with proper regularization.
10.Computational Cost	Lower computational cost per iteration, simpler learners.	Higher computational cost per iteration, more complex learners.
11.Flexibility	Less flexible, usually used for binary classification.	More flexible, can handle various loss functions (e.g., regression, classification).
12.Regularization	Implicit through weighting scheme.	Explicit regularization methods such as tree constraints (depth, subsampling).
13.Risk of Overfitting	Lower, but can still overfit on noisy data.	Higher risk, mitigated by regularization techniques.
14.Typical Applications	Simple binary classification problems.	A wide range of applications, including regression, classification, and ranking.
15.Algorithm Complexity	Simpler, straightforward implementation.	More complex, requires tuning of multiple hyperparameters.

14. What is bias-variance trade off in machine learning?
The bias-variance tradeoff in machine learning describes the tradeoff between two sources of error that affect the performance of predictive models: bias and variance.
Bias:
Bias refers to the error introduced by approximating a real-world problem, which may be complex, by a simplified model. High bias means the model is too simple and cannot capture the underlying patterns of the data, leading to underfitting.
Variance:
Variance refers to the error introduced by the model's sensitivity to fluctuations in the training data. High variance means the model is too complex and captures noise in the training data as if it were a true pattern, leading to overfitting.
The Tradeoff:
The bias-variance tradeoff is the balance between these two sources of error i.e.underfitting and overfitting.

An ideal model finds a balance where it is complex enough to capture the underlying patterns but simple enough to avoid capturing noise, achieving low bias and low variance.

Several techniques can help manage the bias-variance tradeoff:

1.Regularization: Techniques like L1 (Lasso) and L2 (Ridge) regularization add penalties to the loss function for more complex models, discouraging overfitting.

2.Cross-Validation: Use cross-validation to evaluate model performance on different subsets of the data, providing a more robust estimate of generalization error.eg K-fold validation, Leave one out validation

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

In Support Vector Machines (SVM), kernels are used to transform the input data into a higher-dimensional space where it becomes easier to find a hyperplane that can separate the data points of different classes.
1.Linear Kernel:
The linear kernel is the simplest type of kernel. It computes the dot product of two vectors in the original feature space.It is used when the data is linearly separable i.e. it can be separated by a straight line (or hyperplane in higher dimensions).It requires fewer hyperparameters to tune.

2.RBF (Radial Basis Function) Kernel:
The RBF kernel (also known as the Gaussian kernel) maps the input data into an infinite-dimensional space.It is a popular choice because it can handle the non-linear relationship between the features and the labels.It is Powerful and flexible, can model complex relationships.

3.Polynomial Kernel:
The polynomial kernel represents the similarity of vectors in a feature space over polynomials of the original variables.It can fit data with non-linear relationships by considering polynomial combinations of the features.It is used where the relationship between features is polynomial, such as certain types of regression problems and pattern recognition.