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 (coefficient of determination) is generally considered a better measure of goodness of fit in regression compared to Residual Sum of Squares (RSS). R-squared represents the proportion of the variance in the dependent variable that is explained by the independent variables in the model. It ranges from 0 to 1, with 1 indicating a perfect fit. RSS, on the other hand, is the sum of the squared residuals (the differences between the observed and predicted values). While RSS provides information about the overall magnitude of the residuals, it doesn't give a clear indication of the proportion of variability explained by the model. In summary, R-squared is preferred because it provides a normalized measure of goodness of fit and helps assess the percentage of variance in the dependent variable that the model explains. Higher R-squared values indicate a better fit, whereas RSS alone doesn't provide a clear indication of the goodness of fit without additional normalization.

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 important terms used to assess the goodness of fit of a regression model.

Total Sum of Square (TSS): TSS represents the total variability in the dependent variable (Y). It is the sum of the squared differences between each observed dependent variable value and the overall mean of the dependent variable. Equation: $TSS=\sum (Y_i - \bar{Y})^2$ where Yi is each observed value of the dependent variable, and Y is the mean of the dependent variable.

Explained Sum of Squares (ESS): ESS represents the variability in the dependent variable that is explained by the independent variables (the predictors) in the model. It is the sum of the squared differences between the predicted values (fitted values) and the overall mean of the dependent variable. Equation: $ESS=\sum(\hat{Y}_i-\bar{Y})^2$, where Yi is each predicted value of the dependent variable.

Residual Sum of Square (RSS): RSS represents the unexplained variability in the dependent variable, which is attributed to the residuals (the differences between observed and predicted values). It is the sum of the squared differences between each observed value and its corresponding predicted value. Equation $RSS=\sum(Yi-Yi)2$, where Yi is each observed value, and Yi is each predicted value.

The relationship between these metrics is given by the following equation:

$TSS=ESS+RSS$

This equation signifies that the total variability in the dependent variable can be decomposed into the part explained by the model (ESS) and the part unexplained or attributed to residuals (RSS).

3. What is the need of regularization in machine learning?
   ANS Regularization in machine learning is a technique used to prevent overfitting and improve the generalization of a model. The need for regularization arises from the trade-off between fitting the training data well and avoiding the model to become too complex or sensitive to noise. Here are some reasons why regularization is important: Overfitting occurs when a model learns the training data too well, including its noise and outliers, and fails to generalize to new, unseen data. Regularization adds a penalty term to the model's objective function, discouraging overly complex models that fit the training data too closely. In linear regression, when predictor variables are highly correlated (multicollinearity), the model can become sensitive to small changes in the data. Regularization techniques like Ridge Regression can handle multicollinearity by shrinking the coefficients of correlated variables. Regularization methods can drive the coefficients of irrelevant or less important features to zero, effectively performing feature selection. This is particularly useful in high-dimensional datasets where some features may not contribute significantly to the predictive power. Regularization can help improve the numerical stability of the optimization process, especially when dealing with ill-conditioned matrices or near-singular matrices. Regularization introduces a trade-off between fitting the training data and keeping the model simple. It helps control the complexity of the model by penalizing large coefficients.

4. What is Gini–impurity index?

ANS The Gini impurity is a measure used in decision tree algorithms for classification tasks. It quantifies the impurity or disorder in a set of data points. In the context of decision trees, the Gini impurity is used to evaluate the "purity" of a node, which means how often a randomly chosen element would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the node.

The Gini impurity for a node is calculated using the following formula:

$Gini(t)=1-\sum i=1 cpi2$

where: $Gini(t)$ is the Gini impurity for node t,

c is the number of classes or labels,

pi is the proportion of data points in node t that belong to class i

The Gini impurity takes on values between 0 and 1. A Gini impurity of 0 indicates perfect purity, meaning all the data points in the node belong to the same class. A Gini impurity of 1 indicates maximum impurity, meaning the classes are equally distributed in the node.

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

ANS Yes, unregularized decision trees are prone to overfitting. Decision trees have a tendency to be highly flexible and adapt themselves closely to the training data, capturing noise and outliers. This flexibility can lead to the tree memorizing the training data rather than learning the underlying patterns, which results in poor generalization to new, unseen data.

The key reasons why unregularized decision trees are prone to overfitting include:

HIGH VARIENCE: Unregularized decision trees can have high variance, meaning they are sensitive to variations in the training data

MEMORIZAION OF NOISE: Decision trees are capable of capturing noise and outliers in the training data, especially if they are allowed to grow deep.

COMPLEXITY: Unregularized decision trees tend to be complex and can represent very detailed decision boundaries.

LACK OF SMOOTHING: Decision trees do not inherently smooth the decision boundaries, and an unregularized tree may create disjointed, irregular regions in the feature space.

To address the overfitting issue, regularization techniques or hyperparameter tuning can be applied. Regularization methods include limiting the maximum depth of the tree, setting a minimum number of samples required to split a node, or pruning the tree after it has been built. These techniques help prevent the tree from becoming too complex and overfitting the training data, leading to improved performance on unseen data.

6. What is an ensemble technique in machine learning?

ANS An ensemble technique in machine learning is a method that combines multiple models or algorithms to improve the overall performance and robustness of a predictive model. Ensemble methods typically involve training several models independently and then combining their predictions in a way that reduces the variance or bias of the individual models, leading to more accurate and reliable predictions. Common ensemble techniques include bagging, boosting, and stacking

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

Ans The main difference between Bagging and Boosting techniques lies in their approach to combining multiple models:

Bagging (Bootstrap Aggregating) involves training each model in the ensemble independently with a random sample of the training data, and then averaging the predictions to make the final prediction. It aims to reduce variance and prevent overfitting.

Boosting, on the other hand, works by training multiple models sequentially, where each model corrects the errors made by its predecessors. It focuses on reducing bias and improving accuracy by giving more weight to misclassified data points.

In summary, Bagging aims to reduce variance, while Boosting focuses on reducing bias and improving accuracy.

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

Ans The out-of-bag (OOB) error in random forests refers to the mean prediction error on each training sample, using only the trees that did not have the sample in their bootstrap sample during training. In other words, for each tree in the random forest, the OOB error is computed using the data that were not used in training that particular tree. This provides a built-in validation measure for random forests, allowing for an estimation of the model's performance without the need for an additional validation set.

9. What is K-fold cross-validation?

Ans K-fold cross-validation is a technique used to assess the performance of a predictive model. The dataset is divided into K equal-sized subsets, and the model is trained and tested K times. In each iteration, one of the K subsets is used as the test set, and the remaining K-1 subsets are used as the training data. This process allows for a more robust estimation of the model's performance by reducing the impact of data partitioning on model evaluation. Finally, the performance results from the K iterations are averaged to provide a single estimation of model performance.

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

Ans Hyperparameter tuning in machine learning involves the process of selecting the optimal hyperparameters for a learning algorithm. Hyperparameters are parameters set prior to the training process and are not learned from the data. This tuning is done to find the best combination of hyperparameters that result in the most accurate and robust model. It helps to improve the performance of the model by optimizing its ability to generalize to new, unseen data. Common methods for hyperparameter tuning include grid search, random search, and more advanced optimization techniques like Bayesian optimization or genetic algorithms.

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

Ans When using a large learning rate in Gradient Descent, several issues can arise:

1. Divergence: A large learning rate can cause the optimization process to overshoot the minimum of the loss function, leading to divergence instead of convergence.

2. Oscillations: High learning rates can cause the optimization process to oscillate around the minimum of the loss function rather than converging to it.

3. Unstable Training: Large learning rates may result in unstable training, making it difficult for the model to learn the underlying patterns in the data.

4. Inaccurate Results: The model may fail to converge to an optimal solution, leading to inaccurate predictions and poor model performance.

To mitigate these issues, it's crucial to choose an appropriate learning rate that allows for stable convergence during the training process.

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

Ans While Logistic Regression is inherently designed for linear classification, it can be extended to handle non-linear data by using techniques such as feature engineering, adding non-linear features (polynomial features), or by applying kernel tricks in the context of kernelized Logistic Regression. However, this makes the model more complex and may not always be the most efficient or effective method for handling non-linear data. In such cases, other models like decision trees, support vector machines (SVM), or neural networks are often preferred for non-linear classification tasks due to their inherent capability to capture non-linear relationships in the data more effectively.

13. Differentiate between Ada boost and Gradient Boosting.

Ans AdaBoost focuses on sequentially training a series of weak learners (usually decision trees). Each weak learner is trained on a subset of the training data, with misclassified instances receiving higher weights in subsequent iterations. The final model is an ensemble of these weak learners, with each contributing to the overall prediction based on their individual performance. Gradient Boosting also builds an ensemble of weak learners, typically decision trees, but the training process is slightly different. It sequentially trains weak learners to correct the errors (residuals) of the previous models. Each new learner fits to the negative gradient of the loss function with respect to the predictions of the ensemble The final model is an additive combination of these weak learners

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

Ans The bias-variance tradeoff is a fundamental concept in machine learning that involves finding the right balance between two sources of error: bias and variance. Both bias and variance contribute to the overall error of a machine learning model, and understanding the tradeoff between them is crucial for building models that generalize well to new, unseen data.

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

Ans Linear kernel is suitable for linearly separable data or when the relationship between features and classes is approximately linear RBF kernel is versatile and effective for capturing complex, non-linear relationships. It is a common choice when the nature of the data is not well known. Polynomial kernel introduces non-linearity through polynomial functions and is useful when the decision boundary has a polynomial shape. The choice of the kernel depends on the characteristics of the data, and tuning kernel parameters, such as gamma or degree, can significantly impact the performance of the SVM model.