MACHINE LEARNING

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 is a goodness-of-fit measure for linear regression models. This statistic indicates the percentage of the variance in the dependent variable that the independent variables explain collectively. R-squared is a commonly used measure of goodness of fit in regression, indicating the proportion of variance explained by the model. However, Adjusted R-squared is often better for multiple regression as it accounts for the number of predictors. RSS provides an absolute measure of fit but is not as interpretable or comparable across different models.

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 **TSS-** the total sum of squares (TSS) is the sum of squared differences between the observed dependent variables and the overall mean. Think of it as the dispersion of the observed variables around the mean—similar to the variance in descriptive statistics. But TSS measures the total variability of a dataset, commonly used in regression analysis and ANOVA.

The difference between variance and SST is that we adjust for the degree of freedom by dividing by n–1 in the variance formula.

$TSS=n\sum i=1(yi-\bar{}y)2$

ESS- The explained sum of squares (ESS) is the sum of the squares of the deviations of the predicted values from the mean value of a response variable, in a standard regression model.

The Explained SS tells you how much of the variation in the dependent variable your model explained.
Explained SS = Σ(Y-Hat – mean of Y)^2.

RSS- The residual sum of squares (RSS) is a statistical technique used to measure the amount of variance in a data set that is not explained by a regression model itself. Instead, it estimates the variance in the residuals, or error term. The RSS measures the amount of error remaining between the regression function and the data set after the model has been run. A smaller RSS figure represents a regression function that is well-fit to the data.The RSS, also known as the sum of squared residuals, essentially determines how well a regression model explains or represents the data in the model.

$RSS = \sum^{n}_{i=1} (y^i - f(x_i))^2$

The equation relating these three metrics with each other.

TSS=ESS+RSS

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

ANS While training a machine learning model, the model can easily be overfitted or under fitted. 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 help us get an optimal model . Regularization is a technique used to reduce errors by fitting the function appropriately on the given training set and avoiding overfitting.

4. What is Gini–impurity index?

ANS The Gini index measures the extent to which the distribution of income or consumption among individuals or households within an economy deviates from a perfectly equal distribution. A Gini index of 0 represents perfect equality, while an index of 100 implies perfect inequality.

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. More precisely, the Gini Impurity of a dataset is a number between 0-0.5, which indicates the likelihood of new, random data being misclassified if it were given a random class label according to the class distribution in the dataset.

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

ANS Decision trees have a tendency to overfit to the training set because they can keep growing deeper and more complex until they perfectly classify the training data. This can lead to the tree capturing noise in the data, rather than the underlying relationships, and thus performing poorly on new, unseen data. Regularization techniques such as pruning, setting a minimum number of samples required to split a node, or limiting the maximum depth of the tree can help mitigate overfitting in decision trees. Regularization techniques aim to simplify the tree and prevent it from becoming overly complex, thus improving its ability to generalize to new data.

1. High Variance: Decision trees have high variance, meaning they are sensitive to small variations in the training data.Without constraints, decision trees can become overly specific to the training data, capturing noise and outliers rather than generalizable patterns.

2. Unlimited Growth: Unregularized decision trees have no constraints on their growth, meaning they can continue to split nodes until each leaf node is pure (contains instances of only one class). This can result in excessively deep trees with many branches, which are more likely to overfit the training data.

3. Memorization of Noise: Decision trees are capable of memorizing the training data, especially when it contains noise or irrelevant features. Without regularization, decision trees may capture these spurious patterns, leading to poor generalization performance on unseen data.

4. Lack of Pruning: Unregularized decision trees do not incorporate pruning techniques to remove unnecessary branches and simplify the tree structure. Pruning is essential for preventing overfitting by reducing the complexity of the tree while maintaining its predictive power.

6. What is an ensemble technique in machine learning?

ANS Ensemble learning is a machine learning technique that enhances accuracy and resilience in forecasting by merging predictions from multiple models. It aims to mitigate errors or biases that may exist in individual models by leveraging the collective intelligence of the ensemble.

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

ANS The bagging technique combines multiple models trained on different subsets of data, whereas boosting trains models sequentially, focusing on the error made by the previous model.

Bagging is best for high variance and low bias models while boosting is effective when the model must be adaptive to errors, suitable for bias and variance errors.

Generally, boosting techniques are not prone to overfitting. Still, it can be if the number of models or iterations is high, whereas the Bagging technique is less prone to overfitting.

Bagging improves accuracy by reducing variance, whereas boosting achieves accuracy by reducing bias and variance.

Boosting is suitable for bias and variance, while bagging is suitable for high-variance and low-bias models.

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

ANS The out-of-bag (OOB) error is the average error for each calculated using predictions from the trees that do not contain in their respective bootstrap sample. This allows the RandomForestClassifier to be fit and validated whilst being trained. The OOB Error provides an unbiased estimate of the model's performance without the need for a separate validation set. It serves as an internal validation mechanism within the random forest algorithm. Here's how the out-of-bag error works in Random Forests:

1.Bootstrap Sampling: For each decision tree in the Random Forest, a bootstrap sample is drawn from the original training data. This means that some instances are sampled with replacement, while others are not included in the sample.

2. Out-of-Bag Instances: The instances that are not included in the bootstrap sample for a particular tree are referred to as out-of-bag instances for that tree.

3. Prediction for Out-of-Bag Instances: After training each decision tree, the out-of-bag instances (those not included in the bootstrap sample) are passed down the tree, and their predictions are recorded. These predictions serve as estimates of how well the tree generalizes to unseen data

9. What is K-fold cross-validation?

ANS K-fold cross-validation is a technique for evaluating predictive models. The dataset is divided into k subsets or folds. The model is trained and evaluated k times, using a different fold as the validation set each time. Performance metrics from each fold are averaged to estimate the model's generalization performance. In K-Fold Cross Validation, we split the dataset into k number of subsets (known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purpose each time.

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

ANS Hyperparameter tuning is the process of selecting the optimal values for a machine learning model's hyperparameters. Hyperparameters are settings that control the learning process of the model, such as the learning rate, the number of neurons in a neural network, or the kernel size in a support vector machine. The goal of hyperparameter tuning is to find the values that lead to the best performance on a given task

These are used to specify the learning capacity and complexity of the model. Some of the hyperparameters are used for the optimization of the models, such as Batch size, learning rate, etc., and some are specific to the models, such as Number of Hidden layers, etc.

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

ANS Learning rate (λ) is one such hyper-parameter that defines the adjustment in the weights of our network with respect to the loss gradient descent. Learning rate is used to scale the magnitude of parameter updates during gradient descent.

When the learning rate is too large, gradient descent can suffer from divergence. If the learning rate is too high, the algorithm may overshoot the minimum, and if it is too low, the algorithm may take too long to converge.

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

ANS Logistic Regression is a linear classification algorithm, meaning it assumes a linear relationship between the independent variables (features) and the log-odds of the binary outcome. Therefore, it is inherently limited in its ability to model complex, nonlinear relationships between the features and the target variable. If the data is nonlinearly separable, meaning the decision boundary that separates the classes is nonlinear, Logistic Regression may not perform well. In such cases, Logistic Regression may struggle to capture the underlying patterns in the data, leading to poor classification performance.

However, it's important to note that Logistic Regression can still be effective in certain situations where the decision boundary is approximately linear or where nonlinear relationships can be transformed into linear relationships through feature engineering or dimensionality reduction techniques

13. **Differentiate between Adaboost and Gradient Boosting**.

ANS AdaBoost is the first designed boosting algorithm with a particular loss function. On the other hand, Gradient Boosting is a generic algorithm that assists in searching the approximate

solutions to the additive modelling problem. This makes Gradient Boosting more flexible than AdaBoost. Here are the main differences between AdaBoost and Gradient Boosting:

1. lgorithm:

AdaBoost: AdaBoost is a boosting algorithm that focuses on reducing bias. It sequentially trains a series of weak learners (e.g., decision stumps) by adjusting the weights of the training instances. In each iteration, AdaBoost gives higher weights to misclassified instances from the previous iteration, forcing subsequent weak learners to focus on the difficult-to-classifyexamples.

Gradient Boosting: Gradient Boosting is a general framework for boosting algorithms that focuses on reducing both bias and variance. It sequentially builds an ensemble of weak learners by fitting each new learner to the residual errors (or gradients) of the previous model. This is done by minimizing a loss function using gradient descent or a similar optimization algorithm.

2. Loss Function:

AdaBoost: AdaBoost uses an exponential loss function, which assigns larger penalties to misclassified instances. It focuses on reducing classification errors by giving higher weights to misclassified instances in subsequent iterations.

Gradient Boosting: Gradient Boosting can handle a variety of loss functions, including least squares loss for regression problems and logistic loss for classification problems. It focuses on minimizing the overall error by fitting subsequent weaklearners to the residual errors of the previous model.

3.Weight Update:

AdaBoost: AdaBoost updates the weights of training instances at each iteration to emphasize the misclassified examples.It gives more weight to incorrectly classified instances and less weight to correctly classified instances, allowing subsequent weak learners to focus on the difficult-to-classify examples.

Gradient Boosting: Gradient Boosting updates the model's parameters (e.g., regression coefficients, tree splits) by minimizing the loss function with respect to the residuals of the previous model. It fits each weak learner to the gradient of the loss function, hence the name "Gradient Boosting."

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

ANS The bias-variance tradeoff is a fundamental concept in machine learning that refers to the balance between bias and variance in the performance of a predictive model. It describes the tradeoff between the model's ability to capture the true underlying patterns in the data (bias) and its sensitivity to fluctuations or noise in the training data (variance). the bias–variance tradeoff describes the relationship between a model's complexity, the accuracy of its predictions, and how well it can make predictions on previously unseen data that were not used to train the model.

If the algorithm is too simple (hypothesis with linear equation) then it may be on high bias and low variance condition and thus is error-prone. If algorithms fit too complex (hypothesis with high degree equation) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as a Trade-off or Bias Variance Trade-off. This tradeoff in complexity is why there is a tradeoff between bias and variance.

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

ANS Short description of linear , RBF and Polynomial kernels used in SVM:

Linear Kernel:The linear kernel is the simplest kernel function used in SVMs.

It computes the dot product between the feature vectors in the original feature space.

It is suitable for linearly separable data or when the number of features is large compared to the number of samples.

The decision boundary in the transformed feature space is a hyperplane.

RBF (Radial Basis Function) Kernel:The RBF kernel, also known as the Gaussian kernel, is a popular choice for SVMs due to its flexibility.

It maps the data into an infinite-dimensional space using a Gaussian similarity function.

It is suitable for nonlinearly separable data or when the decision boundary is complex and nonlinear.

The RBF kernel has two hyperparameters: \(C\), which controls the regularization strength, and \(\gamma\), which controls the width of the Gaussian kernel.

Polynomial Kernel:The polynomial kernel maps the data into a higher-dimensional space using polynomial functions.

It computes the similarity between two feature vectors as the inner product raised to a specified power \(d\).

It is suitable for data that exhibits polynomial relationships between features.

The polynomial kernel has two hyperparameters: \(C\), which controls the regularization strength, and \(d\), which controls the degree of the polynomial.