MACHINE LEARNING

Q1 to Q15 are subjective answer type questions, Answer them briefly.

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?

   ANSWERE:- R-squared is a better measure of goodness of fit in regression as compared to Residual Sum of Squares (RSS). R-squared is a statistical measure that represents the proportion of the variance in the dependent variable that is predictable from the independent variables. It takes values between 0 and 1, where 1 indicates a perfect fit and 0 indicates that the model explains none of the variability in the data. On the other hand, RSS is the sum of squared differences between the actual and predicted values of the dependent variable. It doesn't provide any information about how well the model explains the variance in the dependent variable. In conclusion, R-squared provides a more intuitive and interpretable measure of the goodness of fit as it gives an idea of how much of the dependent variable's variance can be explained by the independent variables, while RSS only provides the magnitude of the error.

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.

   ANSWERE:- In regression analysis, TSS (Total Sum of Squares), ESS (Explained Sum of Squares), and RSS (Residual Sum of Squares) are measures of the variance in the dependent variable.

   TSS (Total Sum of Squares): It represents the total variance in the dependent variable. It is calculated as the sum of the squares of the differences between the actual values of the dependent variable and its mean.

   ESS (Explained Sum of Squares): It represents the variance in the dependent variable that is explained by the independent variables. It is calculated as the sum of the squares of the differences between the predicted values of the dependent variable and its mean.

   RSS (Residual Sum of Squares): It represents the variance in the dependent variable that is not explained by the independent variables. It is calculated as the sum of the squares of the differences between the actual and predicted values of the dependent variable.

   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?

   ANSWERE:- regularization is an important technique in machine learning as it helps to improve the generalization performance of a model and prevent overfitting, leading to more accurate and reliable predictions on unseen data.Regularization is a technique used in machine learning to prevent overfitting, which occurs when a model is too complex and fits the training data too closely, resulting in poor generalization to unseen data. Regularization helps to reduce the complexity of the model by adding a penalty term to the loss function during training. This penalty term discourages the model from assigning too much importance to any single feature or weight, and as a result, the model becomes more robust and less likely to overfit the data.

4. What is Gini–impurity index?

   ANSWERE:- Gini impurity index is a useful metric for evaluating the quality of splits in decision trees and random forests, helping to identify the features that are most effective in separating the data into different classes.Gini impurity is a measure of the randomness or uncertainty associated with a set of labels in a classification problem. It is used as a criterion for making splits in decision trees and random forests.The Gini impurity index of a node in a decision tree is calculated as the sum of the probability of

misclassifying a randomly chosen example, given the distribution of labels in the node. The Gini impurity is 0 if all the examples in a node belong to the same class and is maximized when the distribution of labels is evenly split between all classes.

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

   ANSWERE:- unregularized decision trees are prone to overfitting because they have the freedom to grow very deep and capture all the variations in the training data, leading to models that are too complex to generalize well to unseen data. Regularization techniques such as pruning, early stopping, or feature selection can be used to reduce overfitting in decision trees.Decision trees work by recursively splitting the data into subsets based on the values of the features, and assigning a class label to each terminal node. As the tree grows deeper, it can capture even the smallest variations in the training data, leading to a model that fits the training data too closely.

6. What is an ensemble technique in machine learning?

   ANSWERE:- An ensemble technique in machine learning is a method of combining multiple models to produce a more accurate and stable prediction. The idea behind ensemble techniques is that by combining the predictions of several models, the errors of individual models can be averaged out, resulting in a more robust and accurate prediction.
   There are several types of ensemble techniques, including:
   Bagging (Bootstrapped Aggregation): This technique trains multiple instances of the same model on different random subsets of the training data and combines the predictions by taking a majority vote or averaging.
   Boosting: This technique trains a sequence of models, each of which tries to correct the errors of the previous model. The final prediction is obtained by weighting the predictions of individual models.
   Random Forest: This is an extension of bagging to decision trees. In a random forest, each tree is trained on a different random subset of the training data and a random subset of the features. The final prediction is obtained by taking the average or majority vote of the predictions of individual trees.
   Stacking: This technique trains multiple base models and uses their predictions as inputs to train a meta-model that makes the final prediction.

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

   ANSWERE:- Bagging-
   Bagging trains multiple instances of the same model on different random subsets of the training data. The final prediction is obtained by combining the predictions of individual models, typically by taking the average or a majority vote.
   The idea behind bagging is to reduce the variance of the model by averaging out the errors of individual models.
   Bagging works well for models that have high variance, such as decision trees, and can lead to improved performance and stability.
   Boosting-
   Boosting trains a sequence of models, each of which tries to correct the errors of the previous model. The final prediction is obtained by weighting the predictions of individual models. The weights are adjusted so that models that perform well on the training data are given more weight.
   The idea behind boosting is to reduce the bias of the model by combining the strengths of multiple models.
   Boosting works well for models that have high bias, such as linear models, and can lead to improved performance and stability.

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

   ANSWERE:- The OOB error is a useful measure of the generalization error of random forest models and can be used to evaluate and improve the performance of the model on unseen data.Out-of-bag (OOB) error is a measure of the generalization error of a random forest model. In a random forest, each tree is trained on a different random subset of the training data (with replacement). The OOB error is the average prediction error on the samples that are not included in the training set of each tree.the OOB error provides an estimate of the performance of the random forest model on unseen data. Since each tree in the random forest only uses a random subset of the data, the OOB samples provide a natural validation set that can be used to estimate the performance of the model without the need for cross-validation.

9. What is K-fold cross-validation?

   ANSWERE:- K-fold Cross-Validation is when the dataset is split into a K number of folds and is used to evaluate the model's ability when given new data. K refers to the number of groups the data sample is split into. For example, if you see that the k-value is 5, we can call this a 5-fold cross-validation.

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

    ANSWERE:- Hyperparameter tuning is the process of selecting the best set of hyperparameters for a machine learning model to improve its performance on a specific task. Hyperparameters are parameters that are not learned from the data, but set before training a model, such as learning rate, number of hidden layers, etc.
    Hyperparameter tuning is done because the performance of a machine learning model depends heavily on the choice of hyperparameters. A poorly chosen set of hyperparameters can result in underfitting or overfitting, poor generalization, and a suboptimal model. The goal of hyperparameter tuning is to find the hyperparameters that result in the best performance for a given task.

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

    ANSWERE:- If the learning rate is set too large in gradient descent, the model parameters may overshoot their optimal values, resulting in convergence to a suboptimal solution or even failure to converge. Additionally, the cost function may oscillate or show chaotic behavior, making it difficult to reach convergence. A high learning rate can also cause the cost function to fluctuate greatly, leading to unstable convergence or even divergence of the optimization process.

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

    ANSWERE:- Yes, logistic regression can be used for classification of non-linear data, but it may not perform well if the relationship between the features and the target variable is highly non-linear. Logistic regression models assume a linear relationship between the input features and the log-odds of the target variable, so if the relationship is more complex, a more sophisticated model such as a decision tree, random forest, or support vector machine may be a better choice.

13. Differentiate between Adaboost and Gradient Boosting.

    ANSWERE:- Boosting Approach: Adaboost uses a technique called "boosting" to iteratively improve the performance of the model by giving more weight to misclassified examples in each iteration. In contrast, Gradient Boosting uses a technique called "boosting" to fit the new model to the negative gradient of the loss function.

    Model Type: Adaboost uses simple base models such as decision stumps, while Gradient Boosting can use a variety of base models such as decision trees, linear regression, or any other differentiable model.

    Optimization: Adaboost is based on a greedy algorithm that adds a base model to the ensemble in the direction of the maximum improvement in accuracy. In contrast, Gradient Boosting uses gradient descent to minimize the loss function and improve the performance of the model.

    Speed: Adaboost is relatively fast and simple to implement, while Gradient Boosting can be slower to train due to its more sophisticated optimization procedure.

    Performance: In practice, Gradient Boosting has been shown to perform better than Adaboost in many cases, but Adaboost remains a popular method due to its simplicity and fast training time.

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

    ANSWERE:- The bias-variance tradeoff is a fundamental concept in machine learning that refers to the tradeoff between a model's ability to fit the training data well (low bias) and its ability to generalize well to new, unseen data (low variance).
    Bias refers to the error that is introduced by approximating a real-world relationship with a simplified model. High-bias models have a strong tendency to underfit the training data, meaning that they do not capture the complexity of the underlying relationship and perform poorly on the training set.
    Variance, on the other hand, refers to the error that is introduced by the model's sensitivity to small fluctuations in the training data. High-variance models have a strong tendency to overfit the training data, meaning that they fit the noise in the data rather than the underlying relationship.

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

ANSWERE:- Linear Kernel: The linear kernel is the simplest kernel and is defined as the dot product of the input vectors. It is used when the data is linearly separable.

Radial Basis Function (RBF) Kernel: The RBF kernel is a non-linear kernel that transforms the input data into a higher dimensional space in which it becomes linearly separable. The RBF kernel is defined as the Gaussian function of the Euclidean distance between the input vectors. It is often used when the data is not linearly separable.

Polynomial Kernel: The polynomial kernel is a non-linear kernel that can be used to fit a polynomial function to the input data. The degree of the polynomial function can be specified as a hyperparameter. The polynomial kernel is defined as the dot product of the input vectors raised to the power of the degree of the polynomial function. It can be used when the data is not linearly separable and a more complex non-linear relationship between the features and the target variable is suspected.