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?

Answer: R-squared is a commonly preferred measure of goodness of fit in regression, as it measures 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 higher values indicating a better fit. On the other hand, Residual Sum of Squares (RSS) measures the sum of the squared differences between the actual and predicted values of the dependent variable, with lower values indicating a better fit. In general, R-squared is preferred as it provides a more interpretable measure of the quality of the regression fit.

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.

Answer: TSS (Total Sum of Squares), ESS (Explained Sum of Squares), and RSS (Residual Sum of Squares) are measures used in regression analysis to assess the fit of a model.
TSS measures the total variance in the dependent variable. It is calculated as the sum of the squared differences between each actual dependent variable value and the mean of all dependent variable values.
ESS measures the variance in the dependent variable that is explained by the independent variables in the model. It is calculated as the sum of the squared differences between each predicted dependent variable value and the mean of all dependent variable values.
RSS measures the error between the actual and predicted dependent variable values. It is calculated as the sum of the squared differences between each actual and predicted dependent variable value.
The relationship between these three metrics is given by: TSS = ESS + RSS
In summary, TSS measures the total variance in the dependent variable, ESS measures the explained variance, and RSS measures the residual or unexplained variance.

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

Answer: Regularization is a technique used in machine learning to prevent overfitting, which occurs when a model is too complex and captures noise or random fluctuations in the training data rather than the underlying relationship. Overfitting can lead to poor performance on unseen data, as the model may fit the training data too closely, causing it to become highly sensitive to small variations in the input data.
Regularization works by adding a penalty term to the cost function that the model minimizes during training. This penalty term discourages the model from having too many parameters or high parameter values, effectively limiting its complexity. The most common forms of regularization in machine learning are L1 (Lasso) and L2 (Ridge) regularization.
In brief, the need for regularization in machine learning arises to prevent overfitting and improve the generalization performance of a model by controlling its complexity.

4. What is Gini-impurity index?

Answer: The Gini impurity index is a measure of the impurity or randomness of a set of items. It is commonly used in decision tree algorithms and other machine learning algorithms to evaluate the quality of a split in a dataset.
The Gini impurity index ranges from 0 to 1, with 0 representing a pure set (i.e., all items belong to the same class) and 1 representing a completely mixed set (i.e., items belong to all classes with equal probability). The Gini impurity index is calculated as the sum of the squared probabilities of each

class, minus the squared probability of the most frequent class.
In decision tree algorithms, the goal is to find the split that minimizes the Gini impurity index, thereby maximizing the purity of the sets resulting from the split. This helps to create more accurate classifications and predictions, as the leaves of the decision tree become more homogeneous.
In summary, the Gini impurity index is a measure of the randomness or impurity of a set of items, used to evaluate the quality of a split in decision tree algorithms and other machine learning algorithms.

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

Answer: Yes, unregularized decision trees are prone to overfitting. Decision trees are a popular machine learning algorithm that recursively divide a dataset into smaller subsets based on the feature that best splits the data according to some criterion (such as the Gini impurity index). However, without regularization, decision trees can become very complex and create highly specific splits that capture the noise or random fluctuations in the training data, rather than the underlying relationship. This leads to overfitting, where the tree becomes too closely adapted to the training data, resulting in poor generalization performance on unseen data.
In summary, unregularized decision trees are prone to overfitting because they can become too complex, capturing the noise in the training data instead of the underlying relationship, leading to poor generalization performance on unseen data.

6. What is an ensemble technique in machine learning?

Answer: Ensemble technique in machine learning is a method that combines multiple models to create a more robust and accurate prediction by leveraging the strengths of multiple models. This is done by training multiple models on the same dataset, or on different subsets of the data, and then combining their predictions through techniques such as voting, averaging, or stacking. Ensemble techniques have been shown to be effective in improving the performance of machine learning algorithms and handling complex and high-dimensional datasets.

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

Answer: Bagging and Boosting are two ensemble techniques in machine learning. Bagging trains multiple models on random subsets of the data and combines their predictions through averaging or voting. Boosting trains a sequence of models, each correcting the mistakes made by the previous model, and combines the predictions through a weighted average. Bagging reduces variance while Boosting reduces bias.

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

Answer: Out-of-bag (OOB) error in random forests is an estimate of the generalization error of the model, which is calculated without using a validation set. In random forests, each tree is trained on a different random subset of the training data, and some samples are not used in the training of any tree. These unused samples, also known as OOB samples, can be used to estimate the model's performance. The OOB error is calculated as the average prediction error on the OOB samples. The OOB error is a good approximation of the model's generalization error and can be used for model selection and tuning.

9. What is K-fold cross-validation?

Answer: K-fold cross-validation is a method of evaluating a machine learning model by dividing the data into K partitions and training the model on K-1 folds and testing it on the remaining fold. This process is repeated K times with each fold being used as the test set once. The performance of the model is then averaged over the K iterations to give an estimate of its generalization error, or its performance on new, unseen data. K-fold cross-validation is a more robust method compared to a simple train/test split and helps avoid overfitting.

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

Answer: Hyperparameter tuning in machine learning is the process of finding the optimal values for the model's hyperparameters, which are parameters that are not learned from the data, but set

prior to training. The goal of hyperparameter tuning is to find the best set of hyperparameters that will result in the best performance of the model on unseen data.
Hyperparameter tuning is done because the choice of hyperparameters can have a significant impact on the model's performance. Different hyperparameter settings can result in models with different capacities, training times, and generalization errors.
Hyperparameter tuning is often done using techniques such as grid search, random search, or Bayesian optimization. The choice of technique depends on the number and nature of the hyperparameters and the computational resources available. In general, hyperparameter tuning is an important step in the machine learning pipeline and can greatly improve the performance of the final model.

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

Answer: A large learning rate in gradient descent can result in the following issues:

1. Overshooting the minimum: A large learning rate can cause the model to overshoot the minimum of the loss function and oscillate back and forth, never converging to the optimal solution.
2. Instability: A large learning rate can result in unstable and erratic updates of the model parameters, making it difficult to find the optimal solution.
3. Slow convergence: A large learning rate can slow down the convergence of the optimization process, making it take longer to reach the optimal solution.
4. Divergence: In extreme cases, a large learning rate can cause the optimization process to diverge, meaning that the loss function value increases instead of decreasing over time.

To avoid these issues, it is important to choose an appropriate learning rate that is small enough to ensure stability and convergence, but not so small that the optimization process takes an excessively long time to converge. Techniques such as learning rate schedules and adaptive learning rate methods can also be used to adjust the learning rate during training to ensure convergence and good performance.

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

Answer: Logistic Regression is a linear model and can only model linear relationships between the features and the target variable. Therefore, if the relationship between the features and the target variable is non-linear, Logistic Regression may not be the most suitable method for classification.
In such cases, non-linear models such as Decision Trees, Random Forests, Support Vector Machines (SVMs), or Neural Networks may be more appropriate. These models can capture more complex relationships between the features and the target variable and can handle non-linear data more effectively.
However, it's possible to use Logistic Regression for non-linear classification problems by transforming the features or adding polynomial features to the data to make the relationship between the features and the target variable linear. But it's important to keep in mind that this approach can lead to overfitting and requires careful feature selection and model tuning to ensure good performance.

13. Differentiate between Adaboost and Gradient Boosting.

Answer: AdaBoost and Gradient Boosting are both boosting algorithms used for improving the performance of weak learners.
AdaBoost (Adaptive Boosting) is a boosting algorithm that focuses on giving more weight to misclassified samples during each iteration, thereby giving the weak learner more importance to improve on those samples in the next iteration. The final prediction is made by combining the predictions of all weak learners in a weighted manner.
Gradient Boosting, on the other hand, is a boosting algorithm that uses gradient descent optimization to minimize the loss function. In each iteration, a new weak learner is trained to correct the residual errors of the previous iteration. The final prediction is made by combining the predictions of all weak learners through weighted averaging.
In summary, AdaBoost focuses on adjusting the sample weights in each iteration, while Gradient Boosting focuses on correcting the residual errors in each iteration through gradient descent optimization.

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

Answer: The bias-variance trade-off is a fundamental concept in machine learning that refers to the trade-off between two sources of error in a model:

1. Bias: Bias refers to the error that is introduced by approximating a real-world problem with a simpler model. A model with high bias tends to make strong assumptions about the underlying data distribution and therefore has limited capacity to capture the true relationship between the features and the target variable. This can result in underfitting, where the model is not complex enough to capture the patterns in the data.

2. Variance: Variance refers to the error that is introduced by having a model that is too complex and highly sensitive to small fluctuations in the training data. A model with high variance tends to have many parameters that can be adjusted to fit the training data very closely, resulting in overfitting, where the model fits the noise in the training data rather than the underlying patterns.

The goal in machine learning is to find a model that strikes a balance between these two sources of error, by having a low bias and a low variance. This can be challenging, as decreasing one source of error typically results in an increase in the other source of error.

The bias-variance trade-off is an important concept in model selection and regularization, where techniques such as cross-validation and regularization are used to control the complexity of the model and avoid overfitting.

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

Answer:

1. Linear Kernel: The linear kernel is the simplest and most straightforward kernel function in support vector machines (SVMs). It is used when the data is linearly separable. It computes the dot product between the input vectors and maps them into a high-dimensional feature space in which a linear boundary can be used to separate the classes.

2. Radial Basis Function (RBF) Kernel: The RBF kernel is a non-linear kernel that is often used in SVMs when the data is not linearly separable. The RBF kernel maps the input vectors into a higher-dimensional feature space where the classes can be separated by a non-linear boundary. The RBF kernel is a radial function that measures the distance between the input vectors and the center of the feature space.

3. Polynomial Kernel: The polynomial kernel is another non-linear kernel used in SVMs for non-linearly separable data. It maps the input vectors into a high-dimensional feature space by computing polynomial combinations of the input features. The polynomial kernel allows for the creation of non-linear decision boundaries in the feature space. The degree of the polynomial and other parameters can be adjusted to achieve the desired level of non-linearity.