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?

Ans: R-squared is a better measure of goodness of fit in a regression model compared to Residual Sum of Squares (RSS). R-squared, also known as the coefficient of determination provides a 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 a higher value indicating a better fit.

On the other hand, RSS simply measures the sum of the squared differences between the observed and predicted values (residuals). While RSS gives insight into how well the model predicts individual data points, it doesn't provide a normalized measure of overall model performance.

In summary, R-squared is more informative for assessing the goodness of fit as it considers the proportion of variance explained, making it a widely used metric in regression analysis.

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: **Total Sum of Squares (TSS):**

Definition: TSS measures the total variability in the dependent variable (the one you're trying to predict) without considering the regression model.

Equation: TSS = $\Sigma(y_i - \bar{y})^2$, where y_i is each individual observed value, and $\bar{y}$ is the mean of all observed values.

Explained Sum of Squares (ESS):

Definition: ESS represents the variability in the dependent variable that is explained by your regression model.

Equation: ESS = $\Sigma (\hat{y}_i - \bar{y})^2$, where $\hat{y}_i$ is the predicted value for each observation, and $\bar{y}$ is the mean of all observed values.

Residual Sum of Squares (RSS):

Definition: RSS measures the unexplained variability in the dependent variable, i.e., the difference between the observed and predicted values.

Equation: RSS = $\Sigma (y_i - \hat{y}_i)^2$, where y_i is each individual observed value, and $\hat{y}_i$ is the predicted value for each observation.

The equation relating these three metrics is:

TSS = ESS + RSS

In simple terms, TSS is the total variation in the data, ESS is the part explained by your model, and RSS is the part left unexplained. The sum of ESS and RSS equals the total variation (TSS) in the data.

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

Ans: Regularization in machine learning is needed to prevent overfitting, which occurs when a model learns the training data too well, including its noise and outliers. Regularization techniques, such as L1 and L2 regularization, add a penalty term to the model's cost function, discouraging overly complex models and promoting generalization to new, unseen data. This helps the model perform better on unseen data by preventing it from fitting the training data too closely and capturing irrelevant details. In simpler terms, regularization helps maintain a balance between fitting the training data well and avoiding overemphasis on noise, leading to a more robust and accurate model.

4. What is Gini–impurity index?

Ans: The Gini impurity index is a measure used in decision tree algorithms to evaluate how often a randomly chosen element would be incorrectly classified. In simpler terms, it helps decision trees determine the best way to split data based on the classes or categories within it. It ranges from 0 to 1, where 0 represents perfect purity (all elements belong to the same class), and 1 represents maximum impurity (elements are evenly distributed across all classes). The decision tree algorithm aims to minimize Gini impurity when making decisions about how to split the data.

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

Ans: Yes, unregularized decision trees are prone to overfitting. Overfitting occurs when a decision tree is too complex and captures noise or random fluctuations in the training data rather than the underlying patterns. Unregularized decision trees can create branches for specific data points, leading to a model that performs well on the training data but poorly on new, unseen data. Regularization techniques, such as pruning or limiting the tree depth, help mitigate overfitting by simplifying the tree structure and improving its generalization to new data.

6. What is an ensemble technique in machine learning?
Ans: An ensemble technique in machine learning is a method that combines the predictions of multiple individual models to improve overall performance. It's like forming a "team" of models to make predictions collectively, often resulting in better accuracy and robustness than individual models alone. Popular ensemble methods include Random Forests and Gradient Boosting.

7. What is the difference between Bagging and Boosting techniques?
Ans: Bagging (Bootstrap Aggregating) and Boosting are both ensemble machine learning techniques, but they differ in their approach to combining multiple models.
Bagging:
- Idea: Build multiple independent models in parallel.
- Data Sampling: Each model is trained on a random subset of the training data (with replacement).
- Model Diversity: Models are diverse due to different subsets of data.
- Voting: Final prediction is often an average or voting among individual model predictions.
Boosting:
- Idea: Build multiple models sequentially, where each model corrects errors made by the previous ones.
- Data Weighting: Emphasizes on misclassified instances from previous models in the training process.
- Model Dependency: Models are dependent as each one focuses on improving the mistakes of the previous.
- Weighted Voting: Final prediction is a weighted sum of individual model predictions.
In short, Bagging creates diverse models independently, while Boosting builds models sequentially, emphasizing correction of errors made by the previous ones.

8. What is out-of-bag error in random forests?
Ans: The out-of-bag (OOB) error in random forests is a measure of the model's performance. It is calculated by evaluating the model on data points that were not used during the training process. In a random forest, each decision tree is trained on a subset of the data, and some data points are left out (out-of-bag). The OOB error is the average prediction error on these omitted data points. It provides an estimate of how well the model may generalize to new, unseen data.

9. What is K-fold cross-validation?
Ans: K-fold cross-validation is a technique used in machine learning to assess the performance and generalizability of a model. It involves splitting the dataset into K subsets, or folds. The model is trained on K-1 folds and tested on the remaining fold, and this process is repeated K times. The performance metrics are averaged over the K iterations, providing a more robust evaluation of the model's effectiveness. This helps ensure that the model's performance is consistent across different subsets of the data and reduces the risk of overfitting or underfitting.

10. What is hyper parameter tuning in machine learning and why it is done?
Ans: Hyperparameter tuning in machine learning involves adjusting the configuration settings of a model, known as hyperparameters, to optimize its performance. Hyperparameters are not learned from the data but are set before the training process. Examples include learning rates, the number of hidden layers in a neural network, or the depth of a decision tree.
The goal of hyperparameter tuning is to find the best combination of these settings to enhance a model's accuracy and generalization on new, unseen data. It's akin to finding the right "settings" for a machine learning algorithm to achieve the best results in a specific task. Techniques like grid search or random search are often employed to systematically explore different hyperparameter values and identify the optimal configuration for a given model and dataset.

11. What issues can occur if we have a large learning rate in Gradient Descent?
Ans: If the learning rate in Gradient Descent is too large, it can lead to the following issues:
Overshooting the Minimum: The algorithm may consistently overshoot the minimum point in the cost function, making it difficult to converge to the optimal solution.
Divergence: Instead of converging to a solution, the algorithm may diverge, causing the parameters to oscillate or even increase without finding the minimum.
Instability: Large learning rates can result in unstable and unpredictable updates to the model parameters, making it challenging to find an optimal solution.
Missing the Minimum: The algorithm might skip over the minimum or bounce back and forth across it, preventing convergence to the best set of parameters.
Using an appropriate learning rate is crucial for the success of the Gradient Descent algorithm, as it affects the convergence and stability of the optimization process.

12. Can we use Logistic Regression for classification of Non-Linear Data? If not, why?
Ans: No, Logistic Regression is generally not suitable for the classification of non-linear data. It assumes a linear relationship between the input features and the output, making it less effective when dealing with complex, non-linear patterns in the data. For non-linear data, other techniques like decision trees, support vector machines, or neural networks may be more appropriate.

13. Differentiate between Adaboost and Gradient Boosting.
Ans: Adaboost and Gradient Boosting are both machine learning techniques used for boosting, a method to enhance the performance of weak learners (simple models) by combining them into a strong learner (complex model).
Adaboost: Focuses on adjusting the weights of misclassified data points in each iteration.
Learns sequentially, giving more emphasis to previously misclassified samples.
Weights of weak learners are adjusted to prioritize difficult-to-classify instances.
Can be sensitive to noisy data and outliers.
Gradient Boosting: Builds a series of weak learners to correct the errors of the previous ones.
Optimizes a loss function by gradient descent, minimizing the difference between predicted and actual values.
Generally, uses decision trees as weak learners.
Robust to outliers but may be prone to overfitting if not properly tuned.

14. What is bias-variance trade off in machine learning?
Ans: The bias-variance tradeoff in machine learning is a balance between two types of errors that a model can make.
Bias: It represents the error introduced by approximating a real-world problem, which is often complex, by a simplified model. High bias can lead to underfitting, where the model is too simple and cannot capture the underlying patterns in the data.
Variance: It measures how much the model's predictions fluctuate for different training datasets. High variance can lead to overfitting, where the model is too complex and learns the training data too well, including its noise and fluctuations.
The goal is to find a model that minimizes both bias and variance, achieving a good balance. Adjusting the complexity of the model and the amount of data used for training are key factors in managing the bias-variance tradeoff.

15. Give short description each of Linear, RBF, Polynomial kernels used in SVM.
Ans: **Linear Kernel:** Description: The linear kernel is the simplest SVM kernel. It works well for linearly separable data, where a straight line can effectively separate the classes.
Beginner Version: Think of it like drawing a straight line between two groups of points on a graph.
RBF (Radial Basis Function) Kernel: Description: The RBF kernel is versatile and can handle both linear and non-linear data. It computes the similarity between data points in a high-dimensional space, mapping them into a feature space.
Beginner Version: Imagine transforming your data into a higher-dimensional space to better separate points.
Polynomial Kernel: Description: The polynomial kernel calculates the similarity between points based on polynomial functions. It's effective for data that doesn't have a clear linear separation in lower dimensions.
Beginner Version: Picture curves and bends on your graph, allowing for more flexible ways to separate different groups.