MACHINE LEARNING ASSIGNMENT

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

Q1. 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 better measure of goodness of fit in regression compared to the Residual Sum of Squares (RSS). R-squared represents the proportion of variance explained by the model, it is obtained by dividing the residual error and total error.On the other hand, RSS simply measures the total squared difference between the observed and predicted values, without comparing its total variation. Thus, R-squared offers a more comprehensive assessment of how well the model fits the data. Secondly when R-squared is near 1 model is good whereas when near 0 or –ve model is worse. Hence R-squared is better for us.

Q2. 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: Total Sum of Squares (TSS) in regression analysis measures the total variability in the dependent variable (Y) around its mean. It represents the total deviation of each data point from the overall mean of Y.

Residual sum of squares is used to calculate the data variance in error or residuals. It is used to calculate the error left between regression data and regression function after running the model. The smaller the value of the residual sum of squares, the better the model.

Explained Sum of Squares (ESS) quantifies the portion of total variability in the dependent variable (Y) that is explained by the regression model. It measures how much of the variability in Y can be attributed to the independent variables included in the regression equation, indicating the model's explanatory power.

The equation relating these metrics is: **TSS=ESS+RSS**

Q3. What is the need of regularization in machine learning?

Answer: Regularization is a technique used to prevent overfitting by adding a penalty term to the model's objective function during training. The objective is to discourage the model from fitting the training data too closely and promote simpler models that generalize better to unseen data. Regularization methods control the complexity of models by penalizing large coefficients or by selecting a subset of features, thus helping to strike the right balance between bias and variance.

Q4. What is Gini-impurity index?

Answer: The Gini-impurity index also known as the Gini Index, is a measure used in decision tree algorithms to assess the impurity or disorder of a dataset. The Gini Index calculates the probability of misclassification for a randomly selected instance within a specific feature. The Gini Index ranges from 0 to 1, where 0 indicates perfect purity (all elements belong to a single class). 1 indicates maximum impurity (elements are evenly distributed across classes).

Q5. Are unregularized decision-trees prone to overfitting? If yes, why?

Answer: Yes, unregularized decision trees are prone to overfitting. Decision trees tend to overfit when the available training data is limited, as they attempt to extract patterns even from noise12. Decision trees can grow to a considerable depth, resulting in intricate decision boundaries. As the tree becomes deeper, it becomes more susceptible to overfitting

Q6. What is an ensemble technique in machine learning?

Answer: An ensemble technique in machine learning involves combining multiple individual models to create a more robust and accurate predictive model. This approach has diversity among the constituent models, such as decision trees, neural networks, or regression models, to remove the weaknesses and enhance overall performance. Ensemble methods like bagging, boosting, and stacking employ different strategies to aggregate predictions from multiple models, often resulting in improved generalization and predictive power compared to any single model alone.

Q7. What is the difference between Bagging and Boosting techniques?

Answer: Bagging stands for Bootstrap aggregating, which combines several models for better predictive results. In statistical classification and regression, bagging improves the stability and accuracy of machine learning algorithms by decreasing the variance and reducing the chances of overfitting.

Boosting involves building a strong classifier from several weak classifiers using the weak models in series. The first step is to build a model from the training set. Then we create the second model, which tries to correct the error incurred while training the first. This process is continued while adding new models until the maximum number of models is reached, or the training data is finished.

Q8. What is out-of-bag error in random forests?

Answer: The out-of-bag (OOB) error in random forests is a measure of prediction error calculated during the training process. In random forests, each decision tree is trained on a bootstrap sample of the original data, leaving out around some of the samples, which are not included in the training of that particular tree. These out-of-bag samples are then used to evaluate the performance of the tree. The OOB error is the average error calculated across all trees in the random forest, based solely on their respective out-of-bag samples. It serves as an estimate of the model's performance on unseen data without the need for a separate validation set.

Q9. What is K-fold cross-validation?

Answer: K-fold cross-validation method divides the input dataset into K groups of samples of equal sizes. These samples are called folds. For each learning set, the prediction function uses k-1 folds, and the rest of the folds are used for the test set. This approach is a very popular CV approach because it is easy to understand, and the output is less biased than other methods.

Q10. What is hyper parameter tuning in machine learning and why it is done?

Answer: Hyperparameter tuning is the process of selecting the optimal values for a machine learning model's hyperparameters. These hyperparameters control various aspects of the learning process and model behavior. Unlike regular model parameters (such as weights and biases), hyperparameters are not learned from the data but are set before training begins. Here's why hyperparameter tuning is important due to these decision: Optimal Performance, Avoiding Overfitting, Model Robustness and Task-Specific Adaptation

Q11. What issues can occur if we have a large learning rate in Gradient Descent?

Answer: A large learning rate in Gradient Descent can lead to several issues. Firstly, it may cause the algorithm to overshoot the optimal solution, resulting in divergence instead of convergence. This leads to instability and prevents the algorithm from effectively minimizing the loss function. Additionally, a large learning rate can cause oscillations or erratic behavior in the gradient descent process, making it difficult for the algorithm to converge to the minimum loss. Moreover, it may hinder the fine-tuning of model parameters, leading to suboptimal performance and slower convergence. Thus, selecting an appropriate learning rate is crucial for efficient and effective gradient descent optimization.

Q12. Can we use Logistic Regression for classification of Non-Linear Data? If not, why?

Answer: Logistic Regression, by design, acts as a linear classifier. It assumes a straightforward connection between input features and the log-odds of class probabilities. However, this simplicity comes with limitations. When dealing with nonlinear data distributions, Logistic Regression may struggle. Its linear decision boundaries might fail to capture the intricate patterns needed for accurate classification. In such scenarios, more flexible algorithms like Support Vector Machines (SVMs) with nonlinear kernels, Decision Trees, Random Forests, or neural networks are often preferred. These models excel at capturing the complex relationships inherent in nonlinear data.

Q13. Differentiate between Adaboost and Gradient Boosting.

Answer: AdaBoost and Gradient Boosting are two popular ensemble learning techniques with distinct characteristics.

AdaBoost minimizes the exponential loss function by assigning higher weights to misclassified instances in each iteration, sequentially training weak learners like decision stumps, and combining their predictions to form the final ensemble model. It is typically used for binary classification tasks due to its simplicity and effectiveness, although it is less flexible compared to Gradient Boosting.

In contrast, Gradient Boosting minimizes the gradient of the loss function, focusing on residuals to iteratively add weak learners and build a strong learner. It offers more flexibility and is suitable for both regression and classification tasks, especially when dealing with complex data relationships. XgBoost, an extension of Gradient Boosting, introduces regularization for enhanced performance and stability. Overall, while AdaBoost and Gradient Boosting share similar principles, Gradient Boosting provides greater flexibility and is more widely applicable across various tasks and datasets.

Q14. What is bias-variance trade off in machine learning?

Answer: The bias-variance trade-off in machine learning is like finding the sweet spot between understanding the main patterns in your data (bias) and not getting too swayed by random fluctuations (variance). If a model is too simple and biased, it might miss important details (underfitting). On the other hand, if it's too complex and high in variance, it might just memorize the training data, not being able to generalize well to new data (overfitting). So, a good model needs to strike a balance between these two extremes, ensuring it captures the main patterns while staying flexible enough to adapt to new situations.

Q15. Give short description each of Linear, RBF, Polynomial kernels used in SVM.

Answer: Linear Kernel: The simplest kernel, it computes similarity between data points using the dot product of their feature vectors. Effective for linearly separable data, resulting in linear decision boundaries.

RBF (Radial Basis Function) Kernel: Widely used for handling nonlinear data, it measures similarity based on the distance between data points in a high-dimensional space. It's versatile but requires tuning of parameters like gamma to control the smoothness of decision boundaries.

Polynomial Kernel: Introduces polynomial terms into the model by computing similarity using the dot product of feature vectors raised to a certain power. Offers flexibility for capturing complex relationships, yet requires careful parameter tuning to prevent overfitting.