MACHINE LEARNING

ASSIGNMENT - 5

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?**
   - Both R-squared and Residual Sum of Squares (RSS) are measures of goodness of fit in regression analysis, but they capture different aspects of the model's performance. Therefore, both measures are useful in evaluating the goodness of fit of a model, but they serve different purposes. R-squared is a useful measure to assess the overall fit of the model and to compare different models, while RSS is useful to identify the degree of the error in the model's predictions. In general, a good model should have both a high R-squared value and a low RSS value, indicating that it explains a large proportion of the variation in the dependent variable and has a low degree of error in its predictions. However, in some cases, one measure may be more important than the other, depending on the research question and the nature of the data being analyzed.

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**.
   - ESS : It measures the deviation of the predicted values from the mean of the dependent variable.

RSS : It It measures the deviation of the observed values from the predicted valuesmeasures the deviation of the observed values.

TSS : It measures the total deviation of each data point from the mean of the dependent variable.

$TSS=ESS+RSS$

3. **What is the need of regularization in machine learning?**
   - A scenario where the machine learning model tries to learn from the details along with the noise in the data and tries to fit each data point on the curve is called Overfitting. While training a machine learning model, the model can easily be overfitted or under fitted. Regularization techniques help reduce the chance of overfitting and help us get an optimal model.

4. **What is Gini–impurity index?**
   - Gini impurity measures how often a randomly chosen element would be incorrectly labeled if it was randomly assigned a label according to the distribution of classes in that node. Gini impurity is a measure used in decision tree algorithms to quantify a dataset's impurity level or disorder.

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

- Yes, unregularized decision trees are prone to overfitting. decision tree trying to learn from a dataset, like students' scores and whether they passed an exam or not. Unregularized decision trees can be very flexible and intricate. They aim to learn the training data as precisely as possible, creating a complex structure that fits the noise in the data, not just the actual patterns. A teacher who memorizes the exact answers to a set of practice questions without understanding the underlying concepts. When faced with new questions, they might struggle because they've memorized specific details rather than learning the broader principles. Similarly, an unregularized decision tree might memorize the specifics of the training data instead of capturing the general patterns. This makes it perform exceptionally well on the training data but poorly on new, unseen data because it has essentially "memorized" the noise. Regularization techniques, like pruning or limiting the depth of the tree, are introduced to prevent this overfitting. They help the decision tree generalize better to new, unseen data by promoting a simpler and more robust structure that captures the underlying patterns rather than the noise.

6. **What is an ensemble technique in machine learning?**
   - Ensemble methods are techniques that aim at improving the accuracy of results in models by combining multiple models instead of using a single model. The combined models increase the accuracy of the results significantly.

Main Types of Ensemble Methods – Bagging, Boosting and Stacking.

7. **What is the difference between Bagging and Boosting techniques?**
   - Bagging is the simplest way of combining predictions that belong to the same type while Boosting is a way of combining predictions that belong to the different types. Bagging aims to decrease variance, not bias while Boosting aims to decrease bias, not variance.

8. **What is out-of-bag error in random forests?**
   - Bootstrap sampling involves randomly selecting data points with replacement from the original dataset to form a new subset. As a result, some data points are included multiple times, while others may not be included at all. Since the bootstrap samples are created with replacement, some data points are left out of each sample. These left-out data points constitute the out-of-bag samples for each tree. For each tree in the random forest, the model's performance is evaluated using the out-of-bag samples that were not used during its training. This provides an unbiased estimate of the model's generalization error.

9. **What is K-fold cross-validation?**
   - K-fold cross-validation is a popular technique used in machine learning to assess the performance and generalization ability of a model. It involves splitting the dataset into K subsets (folds), training the model K times, each time using K-1 folds for training and the remaining fold for validation.

The original dataset is divided into K subsets (folds) of approximately equal size. For example, if K is 5, the dataset is split into 5 folds. The model is trained and evaluated K times. In each iteration, one of the K folds is used as the validation set, while the remaining K-1 folds are used for training. This process is repeated K times, with each fold used exactly once as a validation set. For each iteration, the model's performance metrics (such as accuracy, precision, recall, or others depending on the problem) are recorded based on its predictions on the validation set. After K iterations, the performance metrics are averaged to obtain a more robust and representative estimate of the model's performance.

K-fold cross-validation ensures that every data point is used for validation exactly once, providing a more comprehensive evaluation.

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

- Hyperparameter tuning in machine learning refers to the process of finding the optimal set of hyperparameters for a given model. Hyperparameters are external configuration settings for a model that are not learned from the data but are set prior to training. Examples of hyperparameters include learning rate, regularization strength, number of hidden layers in a neural network, and so on.

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

- **Divergence** :One of the main issues with a large learning rate is that the optimization process may fail to converge. Instead of reaching the minimum of the loss function, the algorithm might oscillate or diverge, preventing the model from learning a good set of parameters.

Overshooting the minimum: With a large learning rate, the updates to the model parameters are substantial. This can lead to overshooting the minimum of the loss function, causing the algorithm to oscillate back and forth around the optimal values without settling down.

Instability: Large learning rates can make the training process unstable. Small changes in the input data or the initial conditions can result in significantly different parameter values, making the model less robust and harder to generalize to new data.

Missed optimal solutions : A large learning rate may cause the optimization algorithm to skip over the optimal solutions. The rapid updates might prevent the model from exploring the loss landscape effectively, leading to suboptimal or divergent solutions.

Slow convergence: Surprisingly, a learning rate that is too large can also slow down the convergence process. The algorithm may keep overshooting the minimum, making it challenging to approach the optimal parameter values efficiently.

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

- Logistic Regression is a linear classification algorithm, meaning it assumes a linear relationship between the input features and the log-odds of the output. While it works well for problems with linearly separable classes, it may struggle with non-linear relationships in the data.

If the decision boundary that separates the classes is non-linear, Logistic Regression might not capture it effectively. In such cases, using a linear model may result in poor classification performance.

13. Differentiate between Adaboost and Gradient Boosting.

- Objective function: Adaboost aims to correct the misclassifications by adjusting data weights, while Gradient Boosting focuses on minimizing the residuals.

Weights vs Residuals : Adaboost assigns weights to data points, while Gradient Boosting fits each weak learner to the residuals of the previous ones.

Training Process: Adaboost trains weak learners sequentially by adjusting weights, and Gradient Boosting builds trees sequentially by minimizing residuals through gradient descent.

In summary, both Adaboost and Gradient Boosting use an ensemble of weak learners, but their approaches to combining these learners and updating the model differ. Adaboost adjusts data weights to emphasize misclassified points, while Gradient Boosting focuses on minimizing residuals through gradient descent.

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

- Bias is the error introduced by approximating a real-world problem, which may be complex, by a simplified model. Variance is the error introduced by using a model that is too complex, capturing noise in the training data rather than the underlying patterns. The bias-variance tradeoff is about finding a model that is complex enough to capture the underlying patterns in the data but not too complex to be sensitive to noise. Striking the right balance leads to a model that performs well on both training and unseen data. It's a critical consideration when designing and evaluating machine learning models.

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

- **Linear Kernal :** The linear kernel is the simplest and most straightforward kernel. It represents a linear relationship between input features. Suitable for linearly separable data where classes can be separated by a straight line.

RBF Kernal : The RBF kernel, also known as the Gaussian kernel, transforms data into an infinite-dimensional space. It is capable of capturing complex, non-linear relationships. Effective for non-linear data where classes are not easily separable by straight lines.

Polynomial kernel : The polynomial kernel transforms data into a higher-dimensional space using polynomial functions. It can capture non-linear relationships and is more flexible than the linear kernel. Suitable for data with non-linear relationships, and the choice of the degree determines the complexity of the decision boundary.

Choosing the appropriate kernel depends on the nature of the data and the problem at hand. Experimenting with different kernels and their parameters is common practice to achieve the best SVM performance.