ASSIGNMENT - 5

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 preferable because it quantifies how well the regression model fits the data, expressed as a percentage between 0 and 100. This makes it easy to interpret and compare across different models. It also indicates the proportion of variability in the dependent variable that is explained by the independent variables. In contrast, RSS lacks normalization, making comparisons between models challenging, and it doesn't provide a clear indication of model fit relative to a baseline.

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 :- TSS (Total Sum of Squares) - It is the total variability in the dependent variable. It measures how much the data points differ from the mean of the dependent variable.

ESS (Explained Sum of Squares) - It is the variability in the dependent variable explained by the regression model. It measures how much of the total variability is accounted for by the independent variables in the model.

RSS (Residual Sum of Squares) - It is the unexplained variability in the dependent variable, also known as the error term. It measures the discrepancy between the observed values and the values predicted by the regression model.

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

In simple terms, the total variability in the dependent variable (TSS) equals the sum of the variability explained by the regression model (ESS) and the unexplained variability (RSS).

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 but fails to generalize to new, unseen data. Regularization techniques help to control the complexity of a model by penalizing large coefficients or model complexity, thus discouraging overfitting. In simple terms, regularization ensures that the model focuses on important patterns in the data rather than noise, leading to better performance on unseen data.

4. **What is Gini impurity index?**

Ans :- Gini impurity measures how mixed up the classes are in a group of data. If all the data in a group belongs to the same class, the Gini impurity is 0 (no mixing). If the data is evenly spread across different classes, the Gini impurity is 0.5 (maximum mixing). In simple terms, it helps decide how to split data into groups when building decision trees.

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

Ans:- Yes, unregularized decision trees are prone to overfitting. This is because they can keep growing deeper and more complex until they perfectly fit the training data, capturing noise and outliers. This excessive complexity can lead to poor performance on new, unseen data.

6. **What is an ensemble technique in machine learning?**

Ans:**-** An ensemble technique in machine learning is like asking a bunch of different people for advice instead of just one. Instead of relying on just a single model to make predictions, you use multiple models together. Each model might have its own strengths and weaknesses, but when you combine their predictions, you often get a more accurate result overall. It's like putting together a team of experts to tackle a problem together.

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

Ans:- Bagging and boosting are both ensemble techniques, but they work differently:

- **Bagging (Bootstrap Aggregating):** In bagging, you create multiple models by training them on different random subsets of the data. Each model gets a say in the final prediction, and then you average out their predictions. Bagging is like taking a vote from a bunch of different people.
- **Boosting**: Boosting also creates multiple models, but it does it in a sequential way. Each new model focuses on the mistakes that the previous ones made, trying to correct them. It's like students studying for a test: if one student struggles with a topic, they spend more time on it until they get it right. The final prediction is a weighted sum of the predictions from all the models.

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

Ans:- In simple terms, the out-of-bag error in random forests is like using leftover ingredients from a recipe to see how well you cooked. When making a random forest, each tree is trained on a random sample of the data, leaving some data untouched. The out-of-bag

error measures how accurately each tree predicts the untouched data it didn't train on. It's like a chef tasting the dish made with the leftovers to judge how well they cooked.

9. **What is K-fold cross-validation?**

Ans: - K-fold cross-validation is a method in machine learning to evaluate how well a model will generalize to new data. Here's a simple breakdown:

- Splitting Data: You divide your dataset into K subsets, or "folds".
- Training and Testing: You train your model K times. Each time, you pick one of the folds as your testing set and the rest as your training set.
- Evaluation: After training each time, you test the model's performance on the testing set.
- Average Performance: Finally, you average the performance metrics across all K training-testing iterations to get a more reliable estimate of how well your model might perform on unseen data.

By repeating this process with different data subsets, you get a more robust understanding of your model's performance, helping you to make better decisions about its effectiveness.

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

Ans: - Hyperparameter tuning in machine learning is like finding the best settings for a model. Here is the simple explanation:

- **What it is**: Every machine learning model has settings called hyperparameters, like the learning rate or the number of trees in a random forest.
- **Why it's done**: We tune these hyperparameters to find the best combination that gives the most accurate predictions. It's like adjusting the knobs on a radio to get the clearest signal.

By tuning these settings, we aim to improve the model's performance and make it more effective in solving the problem we're working on.

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

Ans:- A large learning rate in Gradient Descent can lead to:

- **Overshooting:** Rapidly bouncing back and forth around the minimum point, failing to converge.
- **Instability:** Unpredictable and erratic behavior, hindering convergence towards the optimal solution.
- **Missing the Global Minimum:** Settling for a suboptimal solution instead of reaching the best possible outcome.

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

Ans:- No, Logistic Regression is not suitable for classifying non-linear data because it can only create linear decision boundaries, meaning it can only separate data points with a straight line or plane. If the data is non-linear, like when points can't be separated with a straight line, Logistic Regression won't work well because it can't capture these complex patterns.

13. Differentiate between Adaboost and Gradient Boosting.

Ans: - These are some key differences that highlight how Adaboost and Gradient Boosting differ in their methodologies and behavior.

1. **Approach to Correcting Errors**:
    - Adaboost: Adjusts weights of misclassified data points to focus more on difficult instances.
    - Gradient Boosting: Minimizes a loss function by sequentially fitting models to the residuals of the previous model.
2. **Weight Adjustment**:
    - Adaboost: Adjusts weights of data points based on their classification performance in each iteration.
    - Gradient Boosting: Uses gradients (derivatives) of the loss function with respect to the model's predictions to optimize the model.
3. **Sensitivity to Noisy Data**:
    - Adaboost: More sensitive to noisy data and outliers due to its iterative weight adjustment approach.
    - Gradient Boosting: Less sensitive to noisy data compared to Adaboost.

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

Ans: - The bias-variance tradeoff in machine learning is like a balancing act.

- **Bias** is like having a model that's too simple and misses important details.
- **Variance** is like having a model that's too sensitive to small changes in the data and picks up noise.

The tradeoff is about finding a model that's just right, neither too simple nor too complex. If a model is too simple, it will have high bias but low variance. If it's too complex, it will have low bias but high variance. We want a balance between the two to make accurate predictions on new, unseen data.

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

1. **Ans:-** Linear Kernel:
    - The linear kernel is straightforward and works well when the data can be separated by a straight line or plane.
    - It calculates the similarity between data points by their dot product, essentially measuring how much they align in the feature space.
2. RBF (Radial Basis Function) Kernel:
    - The RBF kernel is versatile and can capture complex relationships between data points.
    - It measures the similarity between data points based on their distance in the feature space, with closer points having higher similarity.
    - It uses a Gaussian function to map the data into a higher-dimensional space.
3. Polynomial Kernel:
    - The polynomial kernel is useful for capturing non-linear relationships in the data.
    - It calculates similarity between data points by elevating the dot product of their feature vectors to a power defined by the degree of the polynomial.
    - Higher degrees allow the kernel to capture more complex patterns in the data.

These kernels are used in Support Vector Machines (SVMs) to transform data into a higher-dimensional space where it's easier to find a linear separation or decision boundary.