ASSIGNMENT - 5

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?
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.
3. What is the need of regularization in machine learning?
4. What is Gini_impurity index?
5. Are unregularized decision-trees prone to overfitting? If yes, why?
6. What is an ensemble technique in machine learning?
7. What is the difference between Bagging and Boosting techniques?
8. What is out-of-bag error in random forests?
9. What is K-fold cross-validation?
10. What is hyper parameter tuning in machine learning and why it is done?
11. What issues can occur if we have a large learning rate in Gradient Descent?
12. Can we use Logistic Regression for classification of Non-Linear Data? If not, why?
13. Differentiate between Adaboost and Gradient Boosting.
14. What is bias-variance trade off in machine learning?
15. Give short description each of Linear, RBF, Polynomial kernels used in SVM.

1)Ans:- 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.

R-squared (also known as the coefficient of determination) measures the proportion of variation in the dependent variable that is explained by the independent variables in the model. In other words, it indicates how well the model fits the data, with values ranging from 0 to 1. Higher R-squared values indicate a better fit, as they mean that a larger proportion of the variation in the dependent variable is explained by the independent variables in the model.

On the other hand, RSS measures the total sum of squared differences between the actual values of the dependent variable and the predicted values by the model. It represents the amount of unexplained variation in the data, and lower RSS values indicate a better fit, as they mean that the model is able to explain more of the variation in the data.

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.

2)Ans:- TSS:-The total sum of squares is a variation of the values of a dependent variable from the sample mean of the dependent variable. Essentially, the total sum of squares quantifies the total variation in a sample. It can be determined using the following formula:

$$TSS=\sum_{i=1}^{n}(y_i-\bar{y})^2$$

ESS(Explained Sum of Squares):- is a statistical quantity used in modeling of a process. ESS gives an estimate of how well a model explains the observed data for the process.

It tells how much of the variation between observed data and predicted data is being explained by the model proposed. Mathematically, it is the sum of the squares of the difference between the predicted data and mean data.

$$\mathrm{ESS} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 .$$

RSS (Residual Sum oF Square):- The residual sum of squares essentially measures the variation of modeling errors. In other words, it depicts how the variation in the dependent variable in a regression model cannot be explained by the model. Generally, a lower residual sum of squares indicates that the regression model can better explain the data, while a higher residual sum of squares indicates that the model poorly explains the data.

$$RSS = \sum^{n}_{i=1} (y^i - f(x_i))^2$$

The relationship between these three sums of squares can be expressed by the following equation:

$$TSS=RSS+\text{Residual SS}$$

This equation states that the total sum of squares (TSS) can be decomposed into the sum of the squared differences between the observed values (Yy_i) and the mean (Y¯y¯) of the dependent variable (TSS), the sum of the squared differences between the predicted values (Y^Yy^$_i$) and the mean (Y¯y¯) of the dependent variable (RSS), and the sum of the squared differences between the observed values (YYy_i) and the predicted values (Y^Yy^$_i$) (Residual SS).

3)ans:- Here are some key reasons for the need of regularization in machine learning:

1. Preventing Overfitting:
    - Regularization helps prevent overfitting, which occurs when a model becomes too complex and fits the training data too closely. Overfitted models may not generalize well to new, unseen data.
2. Handling Collinearity:
    - In situations where features are highly correlated (multicollinearity), regularization techniques like Ridge regression can be beneficial. Ridge regression adds a penalty term to the objective function, which can help stabilize the model when dealing with correlated features.
3. Feature Selection:
    - Regularization methods, such as Lasso regression, have the property of automatically performing feature selection by driving the coefficients of irrelevant or less important features to zero. This can simplify the model and improve interpretability.
4. Improving Numerical Stability:
    - Regularization can improve the numerical stability of the optimization process, especially when dealing with ill-conditioned matrices. It can help prevent situations where small changes in the input data result in large changes in the model parameters.
5. Controlling Model Complexity:
    - Regularization provides a way to control the complexity of a model. By introducing a penalty for large coefficients, it encourages the model to find a balance between fitting the data well and avoiding unnecessary complexity.

4)Ans:- Gini Impurity is a measurement used to build Decision Trees to determine how the features of a dataset should split nodes to form the tree. More precisely, the Gini Impurity of a dataset is a number between 0-0.5, which indicates the likelihood of new, random data being misclassified if it were given a random class label according to the class distribution in the dataset.

- The Gini Index is the additional approach to dividing a decision tree.
- Purity and impurity in a junction are the primary focus of the Entropy and Information Gain framework.
- The Gini Index, also known as Impurity, calculates the likelihood that somehow a randomly picked instance would be erroneously cataloged.

6)ans:- The ensemble methods in machine learning combine the insights obtained from multiple learning models to facilitate accurate and improved decisions. These methods follow the same principle as the example of buying an air-conditioner cited above.

In learning models, noise, variance, and bias are the major sources of error. The ensemble methods in machine learning help minimize these error-causing factors, thereby ensuring the accuracy and stability of machine learning (ML) algorithms.

Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods in machine learning usually produce more accurate solutions than a single model would. This has been the case in a number of machine learning competitions, where the winning solutions used ensemble methods

5)Ans:- Overfitting can be one problem that describes if your model no longer generalizes well.

Overfitting happens when any learning processing overly optimizes training set error at the cost test error. While it's possible for training and testing to perform equality well in cross validation, it could be as the result of the data being very close in characteristics, which may not be a huge problem. In the case of decision tree's they can learn a training set to a point of high granularity that makes them easily overfit. Allowing a decision tree to split to a granular degree, is the behavior of this model that makes it prone to learning every point extremely well — to the point of perfect classification — ie: overfitting.

7)ans:-difference between bagging and boosting are as follows-

Bagging is a method of merging the same type of predictions. Boosting is a method of merging different types of predictions.

Bagging decreases variance, not bias, and solves over-fitting issues in a model. Boosting decreases bias, not variance.

In Bagging, each model receives an equal weight. In Boosting, models are weighed based on their performance.

Models are built independently in Bagging. New models are affected by a previously built model's performance in Boosting.

In Bagging, training data subsets are drawn randomly with a replacement for the training dataset. In Boosting, every new subset comprises the elements that were misclassified by previous models.

Bagging is usually applied where the classifier is unstable and has a high variance. Boosting is usually applied where the classifier is stable and simple and has high bias.

8)ans:- The out-of-bag (OOB) error is a concept associated with the training process of ensemble models, particularly the Random Forest algorithm. Random Forest is an ensemble learning method that builds multiple decision trees during training and combines their predictions for more robust and accurate results.

In a Random Forest, each tree is trained on a bootstrap sample, which is a random sample of the original dataset obtained by sampling with replacement. This means that some data points may be included multiple times in the training set, while others may not be included at all. The OOB error is the error rate of the model on the data points that were not included in the bootstrap sample for training a particular tree.

9)ans:- Cross-validation is a statistical method used to estimate the skill of machine learning models.

It is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

- That k-fold cross validation is a procedure used to estimate the skill of the model on new data.
- There are common tactics that you can use to select the value of k for your dataset.
- There are commonly used variations on cross-validation such as stratified and repeated that are available in scikit-learn.

10)ans:- Hyperparameter tuning is the process of selecting the optimal values for a machine learning model's hyperparameters. Hyperparameters are settings that control the learning process of the model, such as the learning rate, the number of neurons in a neural network, or the kernel size in a support vector machine. The goal of hyperparameter tuning is to find the values that lead to the best performance on a given task.

Here are some key reasons why hyperparameter tuning is important:

Model Performance Improvement: Properly tuned hyperparameters can lead to improved model performance. Suboptimal choices of hyperparameters may result in underfitting or overfitting, which can negatively impact a model's ability to generalize to new, unseen data.

Generalization: Models with well-tuned hyperparameters are more likely to generalize well to new and unseen data. Hyperparameter tuning helps in creating models that are robust and perform well across different datasets.

Avoiding Overfitting: Hyperparameter tuning helps in preventing overfitting, where a model becomes too complex and starts fitting the noise in the training data rather than learning the underlying patterns.

Computational Efficiency: Hyperparameter tuning can also help in making models more computationally efficient. By finding the right set of hyperparameters, training times can be reduced, and resources can be utilized more effectively.

11)ans:- Using a large learning rate in gradient descent can lead to several issues, primarily related to the optimization process and the convergence of the model. Here are some common problems associated with a large learning rate:

1. Divergence: One of the most significant issues is that a large learning rate can cause the optimization process to diverge instead of converging to the minimum of the loss function. Instead of finding the minimum, the algorithm might oscillate or take steps that are so large that it overshoots the optimal point.
2. Overshooting the Minimum: With a large learning rate, the algorithm may take steps that are too large, causing it to overshoot the minimum of the loss function. This back-and-forth oscillation around the minimum can prevent the algorithm from converging.
3. Instability: A large learning rate can lead to instability in the training process. Small fluctuations in the training data or noise can be amplified, making the optimization process unpredictable and less reliable.
4. Failure to Converge: The optimization algorithm may fail to converge to a minimum within a reasonable number of iterations. The learning rate determines the size of the steps taken during each iteration, and if it is too large, the algorithm may not make progress towards the optimal solution.
5. Skipping the Minimum: In extreme cases, a large learning rate may cause the algorithm to skip over the minimum altogether. This is because the steps taken are so large that the algorithm moves past the optimal point without properly adjusting the model parameters.

12)ans:- Here are the reasons why logistic regression might not be suitable for non-linear data:

1. Limited Capacity for Non-Linearity: Logistic Regression has a limited capacity to capture complex, non-linear relationships between features and the target variable. If the decision boundary between classes is non-linear, logistic regression may struggle to model it accurately.
2. Underfitting Non-Linear Data: Since logistic regression is inherently a linear model, it may underfit non-linear data, leading to poor predictive performance. It cannot represent the intricate decision boundaries that may exist in non-linear datasets.

3. Feature Engineering Requirements: To handle non-linear relationships with logistic regression, you might need to perform feature engineering, transforming input features or introducing higher-degree polynomial features. This, however, can increase the risk of overfitting and add complexity to the model.

For non-linear classification tasks, other machine learning models that can capture more complex relationships are often preferred. Some examples include:

1. Polynomial Regression: This involves adding polynomial features to the input data to enable linear models, like logistic regression, to capture non-linear patterns.
2. Support Vector Machines (SVM): SVM can use kernel functions to map the input data into a higher-dimensional space, allowing it to learn non-linear decision boundaries.
3. Decision Trees and Random Forests: These models can naturally capture non-linear relationships in the data by partitioning the feature space into regions.
4. Neural Networks: Deep learning models, such as neural networks, can learn complex non-linear mappings and are capable of handling a wide range of patterns.

In summary, while logistic regression is effective for linearly separable data, it may not perform well on non-linear datasets. When dealing with non-linear relationships, it's often more appropriate to explore other models that are specifically designed to handle such complexities.

13)ans:- Adaboost (Adaptive Boosting) and Gradient Boosting are both ensemble learning techniques used in machine learning, but they have some differences in terms of their algorithms and training processes. Here are the key distinctions between Adaboost and Gradient Boosting:

1. Weighting of Weak Learners:
    - Adaboost: It assigns weights to the training instances and adjusts them during the training process. Misclassified instances are given higher weights, forcing the algorithm to focus more on them in subsequent iterations.
    - Gradient Boosting: It builds trees sequentially, with each tree trying to correct the errors made by the previous ones. Each tree is fit to the residuals (the differences between the predicted and actual values) of the preceding trees.
2. Sequential vs Parallel Learning:
    - Adaboost: It builds weak learners sequentially. Each subsequent learner corrects the errors of the previous ones.
    - Gradient Boosting: It builds weak learners sequentially, similar to Adaboost, but allows for parallelization in the sense that each tree can be built independently if the computational resources allow.
3. Loss Function:
    - Adaboost: It typically uses the exponential loss function, which gives more emphasis to misclassified instances. The algorithm focuses on instances that are harder to classify correctly.
    - Gradient Boosting: It is more flexible in terms of loss functions. It can be customized by choosing different loss functions based on the specific problem, such as mean squared error (for regression) or log loss (for classification).
4. Learning Rate:

  - Adaboost: It introduces a learning rate parameter to control the contribution of each weak learner. A smaller learning rate makes the boosting process more conservative.
  - Gradient Boosting: It also has a learning rate parameter, which scales the contribution of each weak learner. A smaller learning rate makes the algorithm more robust but requires more boosting iterations.
5. Robustness to Outliers:
  - Adaboost: It is sensitive to outliers since it focuses on correcting misclassifications, and outliers may receive higher weights in the process.
  - Gradient Boosting: It is more robust to outliers as it minimizes the impact of individual data points by fitting subsequent trees to the residuals.

14)ans:- Bias-Variance Tradeoff is a fundamental concept in machine learning that deals with the balance between model bias and variance. In simpler terms, it refers to the tradeoff between a model's ability to accurately represent the underlying data patterns (low bias) and its susceptibility to fluctuations with changes in the training data (high variance).

15)ans:- Linear Kernel:

- Description: The linear kernel is the simplest type of kernel used in SVM. It represents a linear relationship between the input features in the original space and is often used when the data is already linearly separable.
- Mathematics: The linear kernel computes the dot product between the feature vectors in the original space.
- Use Case: Suitable for linearly separable data or when the data is expected to be approximately linearly separable.

Radial Basis Function (RBF) Kernel:

- Description: The RBF kernel, also known as the Gaussian kernel, is a popular choice due to its flexibility. It transforms the input data into an infinite-dimensional space and is capable of capturing complex, non-linear relationships in the data.
- Mathematics: The RBF kernel computes the similarity (or distance) between pairs of data points based on the Gaussian function.
- Use Case: Effective for non-linear and complex data distributions. It is widely used when the decision boundary is expected to be highly non-linear.

Polynomial Kernel:

- Description: The polynomial kernel transforms the input data into a higher-dimensional space using polynomial functions. It is useful for capturing non-linear relationships, but it may be less computationally efficient compared to the RBF kernel.
- Mathematics: The polynomial kernel computes the dot product raised to a power, introducing polynomial terms in the feature space.
- Use Case: Suitable for problems where the decision boundary is expected to be polynomial, but it may be less commonly used than the linear or RBF kernels in practice.