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?

ans: R-squared (R^2) is generally considered a better measure of goodness of fit in regression compared to Residual Sum of Squares (RSS). Here's why:

Interpretability: R-squared represents the 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 higher values indicating a better fit. This makes it easier to interpret compared to RSS, which is simply the sum of the squared differences between the observed and predicted values.

Normalization: R-squared is normalized, meaning it is independent of the scale of the dependent variable. This allows for comparisons between different models and datasets. RSS, on the other hand, is not normalized and its value depends on the scale of the dependent variable, making comparisons difficult.

Model Comparison: R-squared can be used to compare the goodness of fit between different models. Higher R-squared values indicate a better fit, whereas comparing RSS alone may not provide meaningful insights, especially when dealing with datasets of different sizes or scales.

Influence of Sample Size: RSS tends to increase with sample size, even if the model fit does not improve. R-squared, however, is adjusted for the number of predictors in the model, providing a more accurate measure of goodness of fit.

Overall, R-squared provides a more comprehensive and interpretable measure of how well the regression model fits the data compared to RSS. However, it's important to note that R-squared has limitations,

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: In regression analysis, Total Sum of Squares (TSS), Explained Sum of Squares (ESS), and Residual Sum of Squares (RSS) are important metrics used to assess the goodness of fit of the regression model. Here's a brief explanation of each:

1. Total Sum of Squares (TSS) :

TSS represents the total variability in the dependent variable (Y) before the regression model is applied. It measures the total deviation of the observed dependent variable values from their mean. TSS is calculated as the sum of the squared differences between each observed dependent variable value and the overall mean of the dependent variable.

Mathematically, TSS is given by:

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

where:

- Y_i is the observed value of the dependent variable for observation i,
- $\bar{Y}$ is the mean of the observed dependent variable values,
- n is the total number of observations.

2. Explained Sum of Squares (ESS) :

ESS represents the variability in the dependent variable (Y) that is explained by the regression model. It measures the deviation of the predicted values from the mean of the dependent variable. ESS is calculated as the sum of the squared differences between each predicted value and the overall mean of the dependent variable.

Mathematically, ESS is given by:

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

where:

- $\hat{Y}_i$ is the predicted value of the dependent variable for observation i based on the regression model.

3. Residual Sum of Squares (RSS) :

RSS represents the unexplained variability in the dependent variable (Y) after the regression model is applied. It measures the deviation of the observed values from the predicted values (residuals). RSS is calculated as the sum of the squared differences between each observed value and its corresponding predicted value.

Mathematically, RSS is given by:

\[RSS = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \]

where:

- \(Y_i \) is the observed value of the dependent variable for observation \(i \),

- \(\hat{Y}_i \) is the predicted value of the dependent variable for observation \(i \) based on the regression model.

The relationship between these metrics can be expressed by the equation:

\[TSS = ESS + RSS \]

This equation illustrates that the total variability in the dependent variable (TSS) can be decomposed into the variability explained by the regression model (ESS) and the unexplained residual variability (RSS).

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

Ans- Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or under fitting

4. What is Gini–impurity index?

Ans- Gini Impurity is a measurement used to Decision build Trees to determine how the features of a dataset should split nodes to form the tree.

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

Ans:Yes, unregularized decision trees are indeed prone to overfitting. Here's why:

1. High Variance : Decision trees are capable of capturing intricate patterns in the training data, even noise or outliers. Without any constraints, they can grow excessively deep, leading to complex trees that perfectly fit the training data but generalize poorly to unseen data. This high variance in the model makes it susceptible to overfitting.

2. Memorization of Training Data : Unregularized decision trees have no mechanisms to prevent them from memorizing the training data. As a result, they may learn to memorize noise or specific characteristics of the training set that don't generalize well to new data.

3. Lack of Pruning : Without regularization techniques such as pruning, decision trees will continue to split nodes until each leaf node is pure (contains only instances of a single class). This can result in overly specific rules that are tailored to the training data but fail to generalize to new data.

4. Sensitive to Small Changes : Decision trees can be highly sensitive to small changes in the training data. Adding or removing a single data point can lead to significant changes in the tree structure, potentially causing overfitting by adapting too much to the noise in the data.

To mitigate overfitting in decision trees, various regularization techniques can be applied, such as:

- Pruning : Removing parts of the tree that do not provide significant improvements in predictive accuracy can prevent overfitting by simplifying the model.
- Limiting Tree Depth : Constraining the maximum depth of the tree can prevent it from growing excessively deep and capturing noise in the data.
- Minimum Samples per Leaf/Node : Requiring a minimum number of samples to split a node or form a leaf can help prevent the creation of nodes that are specific to outliers or noise.
- Feature Selection : Limiting the number of features considered at each split or using feature importance techniques can help prevent the tree from over-relying on irrelevant or noisy features.
- Ensemble Methods : Using ensemble methods like Random Forests or Gradient Boosted Trees can also reduce overfitting by combining multiple trees and aggregating their predictions.

6. What is an ensemble technique in machine learning?

Ans- The combined models increase the accuracy of the results significantly. This has boosted the popularity of ensemble methods in machine learning

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

Ans- The bagging technique combines multiple models trained on different subsets of data, whereas boosting trains the model sequentially, focusing on the error made by the previous model.

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

Ans- out-of-bag is a method of measuring the prediction error of random forests, boosted decision trees, and other machine learning models utilizing bootstrap aggregating (bagging)

9. What is K-fold cross-validation?

Ans- K-fold cross-validation is a technique for evaluating predictive models

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

Ans- Hyper parameter tuning allows data scientists to tweak model performance for optimal results. This process is an essential part of machine learning, and choosing appropriate hyper parameter values is crucial for success

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

Ans- The choice of learning rate can significantly impact the performance of gradient descent. If the learning rate is too high, the algorithm may overshoot the minimum, and if it is too low, the algorithm may take too long to converge.

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

Ans:Yes, logistic regression can be used for classification of non-linear data, but it might not perform well in capturing complex non-linear relationships between features and the target variable.

Logistic regression models the probability of a binary outcome (e.g., 0 or 1) given one or more predictor variables. It assumes a linear relationship between the predictors and the log-odds of the outcome. However, if the relationship between the predictors and the outcome is non-linear, logistic regression may not accurately capture this non-linearity.

In cases where the decision boundary between classes is highly non-linear, logistic regression may struggle to fit the data well. This can lead to poor classification performance and a high error rate.

To address non-linear relationships in the data, other machine learning algorithms that are capable of capturing non-linear patterns might be more suitable. Some alternatives include:

1. Decision Trees : Decision trees can model complex non-linear decision boundaries by recursively partitioning the feature space.

2. Random Forests : Random forests are ensembles of decision trees that can handle non-linear relationships and reduce overfitting.

3. Support Vector Machines (SVM) : SVM can use kernel functions to map the input features into a higher-dimensional space where the data might be linearly separable.

4. Neural Networks : Deep neural networks, particularly those with multiple hidden layers, are highly flexible models capable of learning complex non-linear relationships in the data.

These algorithms can better handle non-linear data and may outperform logistic regression in scenarios where the relationship between the predictors and the outcome is non-linear. However, it's essential to experiment with different algorithms and evaluate their performance using appropriate metrics to determine the best approach for a specific classification task.

13. Differentiate between Adaboost and Gradient Boosting.

Ans- Adaboost is computed with a specific loss function and becomes more rigid when comes to few iterations. But in gradient boosting, it assists in finding the proper solution to additional iteration modeling problem as it is built with some generic features

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

Ans- The bias-variance trade-off is a fundamental concept in machine learning that relates to the model's ability to capture the true relationship between features and target variable while generalizing well to unseen data.

Bias: Bias refers to the error introduced by approximating a real-world problem with a simplified model. High bias models are overly simplistic and tend to underfit the training data, meaning they fail to capture the true underlying patterns in the data

Variance: Variance refers to the model's sensitivity to small fluctuations or noise in the training data. High variance models are complex and highly flexible, often capturing noise along with the underlying patterns in the training data

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

Ans- **Linear Kernel**: This kernel computes the dot product of the input features, making it suitable for linearly separable data. It works well when the relationship between features and target variable is approximately linear

RBF (Radial Basis Function) Kernel: The RBF kernel computes the similarity between two samples in a high-dimensional space. It is capable of capturing complex relationships between features and target variables.

Polynomial Kernel: The polynomial kernel calculates the similarity between two points in feature space as the polynomial of the original variables. It's particularly useful when the data has non-linear relationships and requires higher-order decision boundaries