MACHINE LEARNING

1) R-Squared or RSS (Residual sum of squares) are measures used to assess the goodness of fit for a regression analysis, they differ in aspects on what they determine from the model's performance.

R-squared or the co-efficient of determination measures the proportion of variance in the dependent variable i.e. response that is explained by the independent variables i.e. predictors in the model. Higher R-squared values (closer to 1) indicates a better fit, it ranges between 0 to 1, where 0 indicates that the model doesn't explain any variance and 1 indicates a perfect fit.

Whereas, RSS measures the total sum of squared differences between the actual values of the dependent variable and the predicted values by the model. It reflects the overall magnitude of the residuals that is the difference between observed and predicted values. Lower RSS indicates a better fit because they represent less unexplained variation in the data.

So determining which of the two is a better measure of goodness of fit model in regression, Neither measure is inherently better, both serve different purposes, while R-squared gives a useful measure to assess the overall fit of the model and to compare different models, RSS gives an idea of the overall magnitude of the errors. A perfect model should have a high R-squared and a low RSS indicating that it explains a large proportion of the variation in the dependent variable and has a low degree of error in its predictions.

Which measure is the best depends on the researching aim and the nature of data analyzed.

2) TSS or Total sum of squares is the squared difference between each variable and the mean. It measures how much the actual values of the dependent variable vary from the mean of the dependent variable.

ESS or Explained sum of square also known as regression sum of squares is used to describe the difference between the predicted value and the mean of

the dependent variable. In other words, it is the sum of the squares of the difference between the predicted data and mean data.

RSS or Residual sum of squares also known as sum of squares error is used to measure the amount of variance in a data set that is not explained by a regression model itself. It finds the difference between the observed or actual value of the variable and the estimated value according to the regression line. The lower the value, the better fit of the data with the regression line.

The relationship between these three statistical techniques can be

TSS = ESS + RSS

Where it expresses that total variability equals explained variability and unexplained variability.

3) Regularization is a technique in machine learning to calibrate machine learning models to minimise adjusted lost function and prevent overfitting, also used for handling noisy or incomplete data which ultimately improves the generalization performance of models.

Regularization is necessary in machine learning to counter many common problems, primarily we use it for-

a) Handling noisy data – Data often contains many irrelevant information like outliers, noise or missing data. Regularization helps the model focus on most important patterns.

b) Preventing Overfitting – Overfitting occurs when model learns data while capturing noise and details that do not generalize to new, unseen data. Regularization helps control the complexity of the model.

c) Balancing Bias and variance – Regularization helps in finding the right balance between bias and variance. Too much regularization results in overly simple model with high bias, too little may result in a complex model with high variance.

4) Gini Impurity index also known as gini index or gini coefficient is a measure used in Decision Tree algorithms to assess the impurity or disorder within a

node classifying tasks as how the features of a dataset should split nodes to form the tree.

5) Unregularized decision trees are indeed prone to overfitting, as Overfitting occurs when a model learns the data with all its noise like outliers in addition to the underlying patterns. Many reasons why overfitting can happen is a decision tree, some reasons include

If a decision tree is allowed to grow unregularized, it creates complex trees having nodes for each data point in the data set, which creates poor generalization to unseen data.

Decision trees have high variance, they are sensitive to specific details of the training data, this sensitivity to data fluctuations contributes to overfitting.

6) Ensemble technique in machine learning involves combining predictions from multiple models to improve the overall performance. As noise, variance and bias are major sources of error, Ensemble methods help minimize these error causing factors, ensuring accuracy.

7) Bagging and Boosting are ensemble techniques used to improve and stabilize machine learning models, Difference between bagging and boosting techniques are-

Bagging also known as bootstrap aggregating involves training multiple instances of the same learning algorithm on different subsets of training data and combing the predictions. The subsets are product of sampling with replacement meaning repetition of some samples is possible in given subsets. It aims to reduce variance.

Boosting focuses on training models sequentially, where each new model attempts to correct the errors made by the previous models. Boosting aims to reduce bias and are used with weak models.

8) *Out-of-bag error* is a unique feature of random forest models that allows for estimating the model's generalization performance without needing a separate

validation set. This makes it a valuable tool for assessing the accuracy and potential for overfitting in random forests.

9) K-fold cross-validation is a technique for assessing the performance of predictive models. The dataset is partitioned into k subsets called folds. The model is trained and evaluated k times, using a different fold as the validation set each time. Performance metrics from each fold are averaged to estimate the model's generalization performance. This method aids in model assessment, selection, and hyperparameter tuning, providing a more reliable measure of a model's effectiveness.

10) Hyperparameter tuning is the process of selecting the optimal values for a machine learning models hyperparameters. These are the settings that control the learning process of the model. When we're training machine learning models, each dataset and model needs a different set of hyperparameters, which are a kind of variable. The only way to determine these is through multiple experiments, where we pick a set of hyperparameters and run them through our model. This is called hyperparameter tuning.

11) The learning rate is an important hyperparameter that greatly affects the performance of gradient descent. It determines how quickly or slowly our model learns, and it plays an important role in controlling both convergence and divergence of the algorithm. When the learning rate is too large, gradient descent can suffer from divergence. This means that weights increase exponentially, resulting in exploding gradients which can cause problems such as instabilities and overly high loss values. On the other hand, if the learning rate is too small, then gradient descent can suffer from slow convergence or even stagnation—which means it may not reach a local minimum at all unless many iterations are performed on large datasets. In order to avoid these issues with different learning rates for each parameter/variable, we use adaptive techniques such as Adagrad and Adam which adjust their own learning rates throughout training based on real-time observations of parameters during optimization.

12) While logistic regression is a powerful classification algorithm, it's not well-suited for directly handling non-linear relationships in the data. Here's why:
Linear Decision Boundary:
Logistic regression inherently creates a linear decision boundary, meaning it separates classes using a straight line (in 2D) or a hyperplane (in higher dimensions). This works well when the data points belonging to different classes are naturally separable by a linear boundary.

Inherent Linearity:
The model's prediction function, the logistic function, is itself a linear function of the input features. This limits its ability to capture complex, non-linear patterns.

13) Difference between Adaboost and Gradient Boosting.

a) Adaboost - This method focuses on training upon misclassified observations. Alters the distribution of the training dataset to increase weights on sample observations that are difficult to classify

a) Gradient Boosting - This approach trains learners based upon minimising the loss function of a learner (i.e., training on the residuals of the model)

b) Adaboost - The weak learners in case of adaptive boosting are a very basic form of decision tree known as stumps.

b) Gradient Boosting - Weak learners are decision trees constructed in a greedy manner with split points based on purity scores (i.e., Gini, minimise loss). Thus, larger trees can be used with around 4 to 8 levels. Learners should still remain weak and so they should be constrained (i.e., the maximum number of layers, nodes, splits, leaf nodes)

c) Adaboost - The final prediction is based on a majority vote of the weak learners' predictions weighted by their individual accuracy.

c) Gradient Boosting - All the learners have equal weights in the case of gradient boosting. The weight is usually set as the learning rate which is small in magnitude.

14) The bias-variance tradeoff is a concept in machine learning that describes the relationship between a model's complexity, the accuracy of its predictions, and how well it can make predictions on previously unseen data. It is the balance between bias (underfitting) and variance (overfitting). Ideally, an algorithm should have low bias and low variance, but decreasing one can increase the other. This tradeoff in complexity is why there is a tradeoff between bias and variance. Striking the right balance is crucial for achieving optimal model performance.

15)

1. Linear Kernel:
a) Projects data points onto a higher-dimensional space using a linear transformation.
b) Suitable for: Linearly separable data or when computational efficiency is a priority.
c) Equation: K(x, y) = x^T * y (dot product of two vectors)

2. RBF (Radial Basis Function) Kernel:
a) Transforms data into an infinite-dimensional space, enabling complex non-linear decision boundaries.
b) Suitable for: Non-linear relationships, highly dimensional data, and when generalization ability is crucial.
c) Equation: K(x, y) = exp(-gamma * ||x - y||^2), where gamma is a hyperparameter controlling kernel width.

3. Polynomial Kernel:
a) Combines the features of linear and non-linear kernels, allowing for flexible decision boundaries.
b) Suitable for: Datasets with features that interact with each other in a non-linear way.
c) Equation: K(x, y) = (x^T * y + 1)^d, where d is the degree of the polynomial.

Statistics

1) d
2) c
3) c
4) b
5) c
6) b
7) a
8) a
9) b
10) a