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 method:**

- R-squared is a statistical method that determines the goodness of fit.
- It measures the strength of the relationship between the dependent and independent variables on a scale of 0-100%.
- The high value of R-square determines the less difference between the predicted values and actual values and hence represents a good model.
- It is also called a **coefficient of determination,** or **coefficient of multiple determination** for multiple regression.
- It can be calculated from the below formula:

$$\text{R-squared}=\frac{\text{Explained variation}}{\text{Total Variation}}$$

Residual Sum Of Squares(RSS):

The residual sum of squares (RSS) **measures the level of variance in the error term, or residuals, of a regression model**. The smaller the residual sum of squares, the better your model fits your data; the greater the residual sum of squares, the poorer your model fits your data

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 Squares):

Total variation in target variable is the sum of squares of the difference between the actual values and their mean.

$$TSS = \sum (y_i - \vec{y})^2$$

TSS or Total sum of squares gives the total variation in Y. We can see that it is very similar to the variance of Y. While the variance is the average of the squared sums of difference between actual values and data points, TSS is the total of the squared sums.

ESS (Explained Sum of Squares):

The explained sum of squares (ESS) is the sum of the squares of the deviations of the predicted values from the mean value of a response variable, in a standard regression model — for example, $y_i = a + b_1x_{1i} + b_2x_{2i} + ...$

RSS (Residual Sum of Squares):

Residual for a point in the data is the difference between the actual value and the value predicted by our linear regression model.

$$Residual = actual - predicted = y\_\hat{y}$$

Using the residual values, we can determine the sum of squares of the residuals also known as **Residual sum of squares** or RSS.

$$RSS = \sum (y_i - \hat{y}_i)^2$$

R-squared = (TSS-RSS)/TSS

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

Ans: Regularization is one of the most important concepts of machine learning. It is a technique to prevent the model from over fitting by adding extra information to it.

Sometimes the machine learning model performs well with the training data but does not perform well with the test data. It means the model is not able to predict the output when deals with unseen data by introducing noise in the output, and hence the model is called over fitted. This problem can be deal with the help of a regularization technique.

This technique can be used in such a way that it will allow to maintain all variables or features in the model by reducing the magnitude of the variables. Hence, it maintains accuracy as well as a generalization of the model.

It mainly regularizes or reduces the coefficient of features toward zero. In simple words, "*In regularization technique, we reduce the magnitude of the features by keeping the same number of features.*"

4. What is Gini–impurity index?

Ans:

- Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.

- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
- Gini index can be calculated using the below formula:
- Gini Index= 1- $\sum_j P_j^2$

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

Ans:

Yes, unregularized decision trees prone to over fitting because a too-large tree increases the risk of overfitting, and a small tree may not capture all the important features of the dataset. Therefore, a technique that decreases the size of the learning tree without reducing accuracy is known as Pruning.

6. What is an ensemble technique in machine learning?

Ans:

Ensemble learning is one of the most powerful machine learning techniques that use the combined output of two or more models/weak learners and solve a particular computational intelligence problem. E.g., a Random Forest algorithm is an ensemble of various decision trees combined.

Ensemble learning is primarily used to improve the model performance, such as classification, prediction, function approximation, etc. In simple words, we can summarise the ensemble learning as follows:

"An ensembled model is a machine learning model that combines the predictions from two or more models."

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

Ans:

1. Bagging:

Bagging is a method of ensemble modeling, which is primarily used to solve supervised machine learning problems. It is generally completed in two steps as follows:

- **Bootstrapping:** It is a random sampling method that is used to derive samples from the data using the replacement procedure. In this method, first, random data samples are fed to the primary model, and then a base learning algorithm is run on the samples to complete the learning process.
- **Aggregation:** This is a step that involves the process of combining the output of all base models and, based on their output, predicting an aggregate result with greater accuracy and reduced variance.

Example: In the Random Forest method, predictions from multiple decision trees are ensembled parallelly. Further, in regression problems, we use an average of these predictions to get the final output, whereas, in classification problems, the model is selected as the predicted class.

2. Boosting:

Boosting is an ensemble method that enables each member to learn from the preceding member's mistakes and make better predictions for the future. Unlike the bagging method, in boosting, all base learners (weak) are arranged in a sequential format so that they can learn from the mistakes of their preceding learner. Hence, in this way, all weak learners get turned into strong learners and make a better predictive model with significantly improved performance.

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

The Random Forest is also known as Decision Tree Forest. It is one of the popular decision tree-based ensemble models. The accuracy of these models is higher than other decision trees. This algorithm is used for both classification and regression applications.

In a random forest, we create a large number of decision trees, and in each decision tree, every observation is fed. The final output is the most common outcome for each observation. We take a majority vote for each classification model by feeding a new observation into all the trees.

An error estimate is made for cases that were not used when constructing the tree. This is called an out-of-bag(OOB) error estimate mentioned as a percentage.

The decision trees are prone to overfitting, and this is the main drawback of it. The reason is that trees, if deepened, are able to fit all types of variations in the data, including noise. It is possible to address this by partial pruning, and the results are often less than satisfactory.

9. What is K-fold cross-validation?
Ans:

K-fold cross-validation approach divides the input dataset into K groups of samples of equal sizes. These samples are called **folds**. For each learning set, the prediction function uses k-1 folds, and the rest of the folds are used for the test set. This approach is a very popular CV approach because it is easy to understand, and the output is less biased than other methods.

The steps for k-fold cross-validation are:

- Split the input dataset into K groups
- For each group:
- Take one group as the reserve or test data set.
- Use remaining groups as the training dataset
- Fit the model on the training set and evaluate the performance of the model using the test set.

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

Ans:

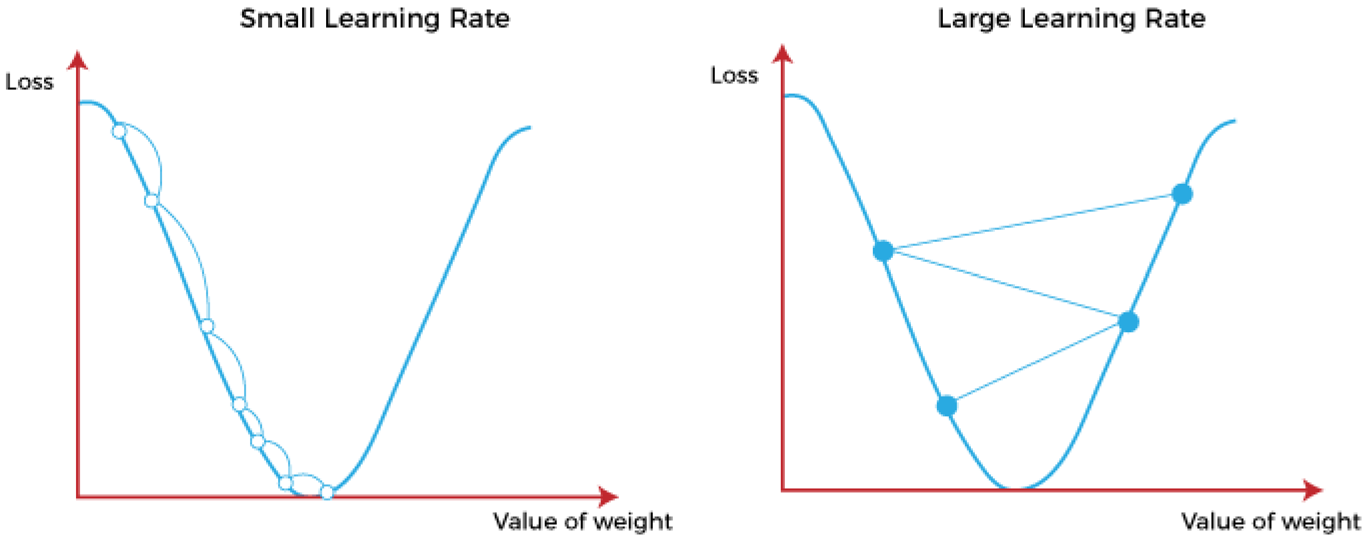
Hyperparameter tuning is significant for the appropriate working of the models of Machine Learning (ML). A method like **Grid Search** appears to be a basic utility for hyperparameter optimization.

The **Grid Search** Method considers some hyperparameter combinations and selects the one returning a lower error score. This method is specifically useful when there are only some hyperparameters in order to optimize. However, it is outperformed by other weighted-random search methods when the Machine Learning model grows in complexity.

11. What issues can occur if we have a large learning rate in Gradient Descent?
Ans:
Gradient Descent is known as one of the most commonly used optimization algorithms to train machine learning models by means of minimizing errors between actual and expected results. Further, gradient descent is also used to train Neural Networks.

Learning Rate:

It is defined as the step size taken to reach the minimum or lowest point. This is typically a small value that is evaluated and updated based on the behavior of the cost function. If the learning rate is high, it results in larger steps but also leads to risks of overshooting the minimum. At the same time, a low learning rate shows the small step sizes, which compromises overall efficiency but gives the advantage of more precision.



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

Ans: No, we cannot use Logistic Regression for classification of Nonlinear data.

When the dependent variable is binary in nature, i.e., 0 and 1, true or false, success or failure, the logistic regression technique comes into existence. Here, the target value (Y) ranges from 0 to 1, and it is primarily used for classification-based problems. Unlike linear regression, it does not need any independent and dependent variables to have a linear relationship.

13. Differentiate between Adaboost and Gradient Boosting.

Ans: **Adaptive boosting or AdaBoost:** This method operates iteratively, identifying misclassified data points and adjusting their weights to minimize the training error. The model continues to optimize sequentially until it yields the strongest predictor.

AdaBoost is implemented by combining several weak learners into a single strong learner. The weak learners in AdaBoost take into account a single input feature and draw out a single split decision tree called the decision stump. Each observation is weighted equally while drawing out the first decision stump.

The results from the first decision stump are analyzed, and if any observations are wrongfully classified, they are assigned higher weights. A new decision stump is drawn by considering the higher-weight observations as more significant. Again if any observations are misclassified, they're given a higher weight, and this process continues until all the observations fall into the right class.

AdaBoost can be used for both classification and regression-based problems. However, it is more commonly used for classification purposes.

Gradient Boosting: Gradient Boosting is also based on sequential ensemble learning. Here the base learners are generated sequentially so that the present base learner is always more effective than the previous one, i.e., and the overall model improves sequentially with each iteration.

The difference in this boosting type is that the weights for misclassified outcomes are not incremented. Instead, the Gradient Boosting method tries to optimize the loss function of the previous learner by adding a new model that adds weak learners to reduce the loss function.

The main idea here is to overcome the errors in the previous learner's predictions. This boosting has three main components:

- **Loss function:**The use of the loss function depends on the type of problem. The advantage of gradient boosting is that there is no need for a new boosting algorithm for each loss function.
- **Weak learner:**In gradient boosting, decision trees are used as a weak learners. A regression tree is used to give true values, which can combine to create correct predictions. Like in the AdaBoost algorithm, small trees with a single split are used, i.e., decision stump. Larger trees are used for large levels,e, 4-8.
- **Additive Model:** Trees are added one at a time in this model. Existing trees remain the same. During the addition of trees, gradient descent is used to minimize the loss function.

Like AdaBoost, Gradient Boosting can also be used for classification and regression problems.

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

Ans. While building the machine learning model, it is really important to take care of bias and variance in order to avoid overfitting and underfitting in the model. If the model is very simple with fewer parameters, it may have low variance and high bias. Whereas, if the model has a large number of parameters, it will have high variance and low bias. So, it is required to make a balance between bias and variance errors, and this balance between the bias error and variance error is known as **the Bias-Variance trade-off**

For an accurate prediction of the model, algorithms need a low variance and low bias. But this is not possible because bias and variance are related to each other:

- If we decrease the variance, it will increase the bias.
- If we decrease the bias, it will increase the variance.

Bias-Variance trade-off is a central issue in supervised learning. Ideally, we need a model that accurately captures the regularities in training data and simultaneously generalizes well with the unseen dataset. Unfortunately, doing this is not possible simultaneously. Because a high variance algorithm may perform well with training data, but it may lead to overfitting to noisy data. Whereas, high bias algorithm generates a much simple model that may not even capture important regularities in the data. So, we need to find a sweet spot between bias and variance to make an optimal model.

Hence, the ***Bias-Variance trade-off is about finding the sweet spot to make a balance between bias and variance errors.***

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

Ans. Polynomial Kernels:

In machine learning, the polynomial kernel is a kernel function commonly used with support vector machines (SVMs) and other kernelized models, that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.

RBF:

In machine learning, the radial basis function kernel, or RBF kernel, is **a popular kernel function used in various kernelized learning algorithms**. In particular, it is commonly used in support vector machine classification.