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?**

   ANSWER :- R-squared and residual sum of squared (RSS) Both are commonly used in measure of goodness of fit for a regression model, but they serve different purpose and have different properties but R-squared is uses more.

   R-squared is a measure of proportion of variance in a dependent variable that is predictable from a independent variable (s). it is defined as the ratio of the explained variance of the total variance. R-squared values range from 0 to 1 , with higher value indicating better fit. R-squared is a popular measure of fit because it is a easy to interpret and has a clear upper bound 1.

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**

   ANSWER :- In regression analysis TSS,ESS and RSS are important metrics that help us understand the models performance.

   Total Sum Of Squares (TSS) :- This measures the total variation in dependent variable(y) with respect to its mean. TSS can be calculated as the sum of squared differences between each data point and the mean of y.

   Explained Sum Of Squares (ESS) :- This measures is the portion of the total variation that is explained by the dependent variable (s) (x). ESS can be calculated as the sum of squared differences between the fitted values (y_hat) and the mean of y.

   Residual Sum Of Square (RSS) :-This measures the remaining unexplained variation in the dependent variable after accounting for the independent variable (s). RSS can be calculated as the sum of squared differences between the actual values (y). and the fitted values (y_hat)

TSS = ESS + RSS

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

ANSWER:- Regularization is a technique used in machine learning to avoid the overfitting data, which occurs when a model learn the training data to well and perform the poorly unseen data. Overfitting happened when a model has too many parameters relative to the number of training examples allowing it to fit the noise in the training data instead of the underlying pattern.

Regularization add to the penalty term to the loss function, which discourages the model from learning overly complex pattern in the training data. This penalty term is a function of the models parameters and it encourage the model to have smaller parameter values leading to simpler model that generalize better to unseen data.

There are two types of regularization

L1 Regularization Lasso

L2 Regularization Ridge

4. **What is Gini–impurity index?**

ANSWER :- The Gini-impurity index is a measure used in decision tree algorithms to determine the quality of a split. It measures the degree of probability of a particular variable being wrongly classified when it is randomly chosen. The Gini impurity ranges from 0 to 1, where 0 indicates perfect purity and 1 indicates maximum purity, and the Gini impurity index for a set of instances can be calculated as

$$\text{Gini impurity} = 1 - \sum (p\_i)^2$$

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

ANSWER :- Yes unregularized decision tree prone to overfitting because High flexibility, lack of regularization, Data snooping

1.High flexibility :- Decision trees can model complex relationship between the input features and the target variable. This flexibility allow them to fit the training data very closely, even when the relationship are noisy or spurious

2.Lack of regularization:- unregularized decision tree do not have any mechanism to prevent overfitting in particular, there is no penalty for adding more nodes are branches to the tree, which can lead to deep in complex trees that fit the noise and training data.

3.Data snooping :- Decision trees into recursively partition the feature space into smaller and smaller regions, based on the value of the input features. This process can lead the data snooping, where the tree captures the random fluctuation and the training data instead of the underlying patterns.

6. What is an ensemble technique in machine learning?

ANSWER:- An ensemble technique in machine learning is a method that combines the prediction of multiple model to produce a final prediction. The idea behind ensemble method is that by combining the strength of multiple models, we can often achieve better performance than any signal model would achieve on its own. Ensemble method can be used for both regression and classification model and there are several types of ensemble techniques.

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

ANSWER:-

Bagging : Bagging (Bootstrap aggregating) is a technique that involves creating multiple subsets of the training data with replacement training a model on each sunset, and then combining the predictions of each model to produce a final prediction. Bagging can reduce the variance of the model and

improve its generalization performance. An example of a bagging algorithm is the random forest algorithm.

Boosting : Boosting is a technique that involves training multiple model in sequence, where each is model trained to correct the errors made by the previous model. The final prediction is a weighted combination of the predictions of each models. Boosting can reduce the bias and variance of the model and improve its generalizations performance. Examples of boosting algorithms include AdaBoost, Gradient Boosting, XGBoost.

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

ANSWER:- In random forest, out-of-bag (OOB) error is an estimate of an organization error of the model. It is calculated by using the samples that where not included in the bootstrap samples used to train is decision tree.

When creating a decision tree in a random forest, a random subset of the training data is selected with replacement bootstrapping. Approximately one-third of the samples are not included in the bootstrap simple and are called out-of-bag samples. These samples are used to estimate the generalization error of the model.

To calculate the OOB error, the model makes predictions on the OOB samples using the decision trees that did not include those samples in their bootstrap samples. The predictions are then compared to the rule labels to calculate the error rate. This process is repeated for each decision tree in the forest, and the OOB error is averaged across all trees.

9. What is K-fold cross-validation?

ANSWER:- K-fold cross-validation is a technique used to evaluate the performance of a machine learning model by splitting the data into k folds, or subsets, and training and testing the model k times. Each time, a different fold is used as the test set, and the remaining K-1 folds are used as the training set. The performance of the model is then averaged across the K runs to obtain a more robust estimate of its generalization performance.

K-folds cross-validation can help reduce the variance of the performance estimate, as it uses more of the data for testing than simple holdout validation. It can also help reduce the risk of overfitting, as it provides more testing data for the model to learn.

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

ANSWER:- hyperparameter tuning is the process of adjusting the values of the hyperparameter is the machine learning model to optimize its performance. Hyperparameters are parameters that are not learned from the data, but are set before training the model. Examples hyperparameters include the learning rate, the numbers of hidden layers in a neural networks, or the regularizations strength.

Hyperparameter tuning is done to improve the generalization performance of the model, by finding the hyperparameter values that result in the best performance on the validation set. By adjusting the hyperparameter, we can control the bias-variance trade-off the model, and find the optimal balance the between underfitting and overfitting.

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

ANSWER:- In gradient descent , learning rate determines how much the weight in the model are adjusted with respect to the gradient during each iteration. If the learning rate is too large. Several issues can occur.

Oscillation:- The model might oscillate around the minimum value instead of converging, leading to unstable training.

Divergence:- The model might diverge, meaning that the weights become extremely large or small, causing numerical instability and poor performance.

Missing The Minimum:- A large learning rate might cause the model to skip over the optimal minimum value, resulting in worse generalization performance

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

ANSWER:- Logistic Regression is a linear model, which means that it can only model linear decision boundaries. Therefore, it may not be suitable for classifying non-linear data. However, there are techniques to extend Logistic Regression to handle non-linear data, such as using polynomial features or kernel methods.

13. Differentiate between Adaboost and Gradient Boosting.

ANSWER:- Adaboost and Gradient Boosting are both ensemble method use for classification and regression problem. While they share some similarities.

Sr.No	Adaboost	Gradient Boosting
1	Adaboost uses a different objective function than Gradient boosting, Adaboost aims to minimize the exponential loss function.	Gradient Boosting the minimizes a differentiable loss function using gradient descents.
2	Adaboost uses a decision tree with a single split as the weak learner.	Gradient Boosting can use any differentiable loss function as the weak learner.
3	Adaboost the weight of the training instance are updated based on the prediction error of the previous model.	Gradient Boosting the weight of the training instance are updated based on the gradient of the loss function
4	Adaboost does not have an explicit regularization term.	Gradient Boosting has a regularization term to prevent overfitting.

5	Adaboost is the simpler algorithm than gradient Boosting as it only requires minimizing the exponential loss function.	Gradient Boosting requires more computational resources, as it involves computing the gradient of the loss function.

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

ANSWER:- In machine learning the bias-variance trade off is a fundamental concepts that describes the trade off between bias and variance when building and tuning models.

Bias refers to the simplifying assumption made by the model to make the target function easier to approximate. High bias can lead to underfitting, where the model is too simple to capture the complexity of the data, resulting in poor predictive performance.

Variance, on the other hand, refers to the amount by which our model would change if we estimate it using a different training dataset. High variance can lead to overfitting, where the model is excessively complex and captures the noise in the data rather than the underlying pattern, also resulting in poor predictive performance.

The goal of any machine learning algorithm is to find the right balance between bias and variance, such that the model is complex enough to capture the underlying pattern in the data, but not so complex that it is overly sensitive to the noise in the data.

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

ANSWER:- These three are brief description common kernels used in Support Vector Machine(SVMs): Linear, Radial Basis Function (RBF), and intercept term. Mathematically, it can be represented as:

Linear Kernel: The linear kernel is the simplest kernel function used in SVMs. It calculates the inner product between two input vectors, and adds an intercept term. Mathematically, it can be represented as:

K(x,y)=x^Ty+r

Where x and y are inputs vectors, and r is the intercept term. The linear kernel is used when the data is linearly separable, meaning that it can be separated into different classes using a single hyperplane

Radial Basis Function(RBF) Kernel: The RBF kernel is a popular kernel function used in SVMs when the data is not linearly separable. It measures the similarity between into input vectors based on the Euclidean distance between them. Mathematically, it can be represented as:

K(x,y)=exp(-gamma * ||x-y||^2)

Where x and y are input vectors, gamma is a free parameter that controls the width if the Gaussian function, and ||x-y||^2Is the squared Euclidean distance distance between the two vectors

Polynomial Kernel: The polynomial kernel is another kernel function used in SVMs when the data is not linearly separabl. It calculates the inner product between two input vectors raised to a power, and adds an intercept term. Mathematically, it can be represented as:

K(x,y)=(gamma * x^Ty+r)^d

Where x and y are inputs vectors, gamma is a free parameter that controls the influence of each other, r is the intercept term, d is the degree of the polynomial.