MACHINE LEARNING

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?

   R-squared is the absolute amount of variation as a proportion of total variation and RSS is the absolute amount of explained variation, usually a higher r-squared indicates more variability is explained by the model. However it is not always the case that a higher r-squared is good for the regression model. The RSS measures the amount of error remaining between the regression function and the data set after the model has been run. A smaller RSS represents a regression function that is well fit to the data. Hence it is better to go with RSS rather than r-squared.

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.

   The total sum of squares (TSS) measures how much variation there is in the observed data. Explained sum of square (ESS) or Regression sum of squares or Model 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.
   The residual sum of squares (RSS) measures the level of variance in the error term, or residuals, of a regression model.
   TSS = ESS + RSS

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

   In Machine Learning we often divide the dataset into training and test data, the algorithm while training the data can either learn the data too well, even the noises which is called over fitting or do not learn from the data, cannot find the pattern from the data which is called under fitting.
   Now, both over fitting and underfitting are problems one need to address while building models. Regularization in Machine Learning is used to minimize the problem of overfitting, the result is that the model generalizes well on the unseen data once overfitting is minimized. To avoid overfitting, regularization discourages learning a more sophisticated or flexible model. Regularization will try to minimize a loss function by inducing penalty.

4. What is Gini–impurity index?

   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.

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

   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.

6. What is an ensemble technique in machine learning?
   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.

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

   Bagging : Bagging (or Bootstrap Aggregation), is a simple and very powerful ensemble method. Bagging is the application of the Bootstrap procedure to a high-variance machine learning algorithm, typically decision trees.

   The idea behind bagging is combining the results of multiple models (for instance, all decision trees) to get a generalized result. Now, bootstrapping comes into picture.

   Bagging (or Bootstrap Aggregating) technique uses these subsets (bags) to get a fair idea of the distribution (complete set). The size of subsets created for bagging may be less than the original set.

   Boosting : Boosting is a sequential process, where each subsequent model attempts to correct the errors of the previous model. The succeeding models are dependent on the previous model.

   In this technique, learners are learned sequentially with early learners fitting simple models to the data and then analyzing data for errors. In other words, we fit consecutive trees (random sample) and at every step, the goal is to solve for net error from the prior tree.

   When an input is misclassified by a hypothesis, its weight is increased so that next hypothesis is more likely to classify it correctly. By combining the whole set at the end converts weak learners into better performing model.

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

   Generally, in machine learning and data science, it is crucial to create a trustful system that will work well with the new, unseen data. Overall, there are a lot of different approaches and methods to achieve this generalization. Out-of-bag error is one of these methods for validating the machine learning model. This approach utilizes the usage of bootstrapping in the random forest. Since the bootstrapping samples the data with the possibility of selecting one sample multiple times, it is very likely that we won't select all the samples from the

original data set. Therefore, one smart decision would be to exploit somehow these unselected samples, called out-of-bag samples. Correspondingly, the error achieved on these samples is called out-of-bag error. What we can do is to use out-of-bag samples for each decision tree to measure its performance. This strategy provides reliable results in comparison to other validation techniques such as train-test split or cross-validation.

9. What is K-fold cross-validation?

   Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

   The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

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

    In machine learning, we need to differentiate between parameters and hyperparameters. A learning algorithm learns or estimates model parameters for the given data set, then continues updating these values as it continues to learn. After learning is complete, these parameters become part of the model. For example, each weight and bias in a neural network is a parameter. Hyperparameters, on the other hand, are specific to the algorithm itself, so we can't calculate their values from the data. We use hyperparameters to calculate the model parameters. Different hyperparameter values produce different model parameter values for a given data set. Hyperparameter tuning consists of finding a set of optimal hyperparameter values for a learning algorithm while applying this optimized algorithm to any data set. That combination of hyperparameters maximizes the model's performance, minimizing a predefined loss function to produce better results with fewer errors.

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

    In order for Gradient Descent to work, we must set the learning rate to an appropriate value. This parameter determines how fast or slow we will move towards the optimal weights. If the learning rate is very large we will skip the optimal solution. If it is too small we will need too many iterations to converge to the best values. So using a good learning rate is crucial. In simple language, we can define learning rate as how quickly our network abandons the concepts it has learned up until now for new ones.

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

    Logistic Regression can't be used for classification of non linear data since Logistic regression is considered a generalized linear model because the outcome always depends on the sum of the inputs and parameters. Or in other words, the output cannot depend on the product (or quotient, etc.) of its parameters.

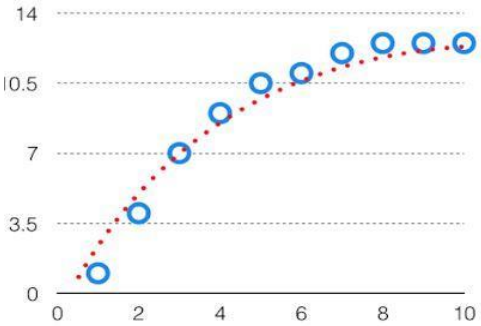
13. Differentiate between Adaboost and Gradient Boosting

S.No	Adaboost	Gradient Boost
1	An additive model where shortcomings of previous models are identified by high-weight data points.	An additive model where shortcomings of previous models are identified by the gradient.
2	The trees are usually grown as decision stumps.	The trees are grown to a greater depth usually ranging from 8 to 32 terminal nodes.
3	Each classifier has different weights assigned to the final prediction based on its performance.	All classifiers are weighed equally and their predictive capacity is restricted with learning rate to increase accuracy.
4	It gives weights to both classifiers and observations thus capturing maximum variance within data.	It builds trees on previous classifier's residuals thus capturing variance in data.

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

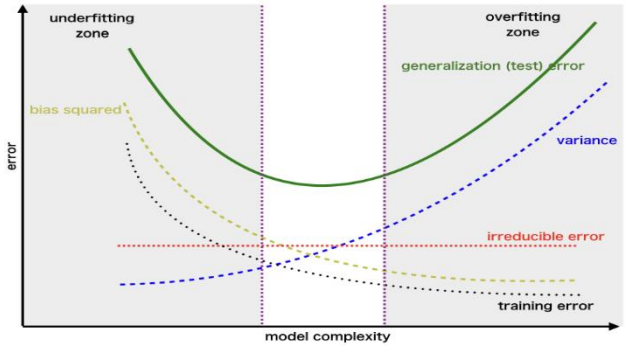
If the algorithm is too simple (hypothesis with linear eq.) then it may be on high bias and low variance condition and thus is error-prone. If algorithms fit too complex (hypothesis with high degree eq.) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as Trade-off or Bias Variance Trade-off.

This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can't be more complex and less complex at the same time. For the graph, the perfect tradeoff will be like.



The best fit will be given by hypothesis on the tradeoff point.

The error to complexity graph to show trade-off is given as –



This is referred to as the best point chosen for the training of the algorithm which gives low error in training as well as testing data.

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

Linear Kernel : It is the most basic type of kernel, usually one dimensional in nature. It proves to be the best function when there are lots of features. The linear kernel is mostly preferred for text-classification problems as most of these kinds of classification problems can be linearly separated.

Linear kernel functions are faster than other functions.

Linear Kernel Formula
F(x, xj) = sum(x.xj)

Here, x, xj represents the data you're trying to classify.

Gaussian Radial Basis Function (RBF) : It is one of the most preferred and used kernel functions in svm. It is usually chosen for non-linear data. It helps to make proper separation when there is no prior knowledge of data.

Gaussian Radial Basis Formula
F(x, xj) = exp(-gamma * ||x - xj||^2)

The value of gamma varies from 0 to 1. You have to manually provide the value of gamma in the code. The most preferred value for gamma is 0.1.

Polynomial Kernel : It is a more generalized representation of the linear kernel. It is not as preferred as other kernel functions as it is less efficient and accurate.

Polynomial Kernel Formula
F(x, xj) = (x.xj+1)^d

Here '.' shows the dot product of both the values, and d denotes the degree.

F(x, xj) representing the decision boundary to separate the given classes.