1) R-Squared (R^2)

When we perform regression, then how good the model fit was for the regression depends on how well we pre-processed the data and what algorithm we used for fitting the regression model. Now there needs to be some kind of metrics to determine how good the fit is.
R-Squared (also denoted by R^2) is one of those metrics. **R-squared** is a statistical measure that represents the goodness of fit of a regression model. The ideal value for r-square is 1. The closer the value of r-square to 1, the better is the model fitted.

RSS
The residual sum of squares (RSS) is a statistical technique used to measure the amount of variance in a data set that is not explained by a regression model itself. Instead, it estimates the variance in the residuals, or error term.

Linear regression is a measurement that helps determine the strength of the relationship between a dependent variable and one or more other factors, known as independent or explanatory variables.

2) In statistics, the explained sum of squares (ESS), alternatively known as the model sum of squares or sum of squares due to regression (SSR – not to be confused with the residual sum of squares (RSS) or sum of squares of errors), is a quantity used in describing how well a model, often a regression model, represents the data being modelled. In particular, the explained sum of squares measures how much variation there is in the modelled values and this is compared to the total sum of squares (TSS), which measures how much variation there is in the observed data, and to the residual sum of squares, which measures the variation in the error between the observed data and modelled values.

Equation result that $TSS = ESS + RSS$

3) Regularization is often used as a solution to the overfitting problem in Machine Learning. Common causes for overfitting are

1. When the model is complex enough that it starts modeling the noise in the training data.

2. When the training data is relatively small and is an insufficient representation of the underlying distribution that it is sampled from, the model fails to learn a generalizable mapping.

Regularization consists of different techniques and methods used to address the issue of over-fitting by reducing the generalization error without affecting the training error much. Choosing overly complex models for the training data points can often lead to overfitting. On the other hand, a simpler model leads to underfitting the data. Hence choosing just the right amount of complexity in the model is critical. Since the complexity of the model can not be directly inferred from the available training data, it is often impossible to stumble upon the right model complexity for training.

4)

Decision Tree is one of the most popular and powerful classification algorithms that we use in machine learning. As the name itself signifies, decision trees are used for making decisions from a given dataset. The concept behind the decision tree is that it helps to select appropriate features for splitting the tree into subparts similar to how a human mind thinks.

To build the decision tree in an efficient way we use the concept of Entropy/Information Gain and Gini Impurity..

The Gini impurity measure is one of the methods used in decision tree algorithms to decide the optimal split from a root node, and subsequent splits. It is the most popular and the easiest way to split a decision tree and it works only with categorical targets as it only does binary splits.

Gini Impurity is calculated using the formula,

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

5) Overfitting can be one problem that describes if your model no longer generalizes well. 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, overfitting.

following steps to avoid overfitting:

- Use a test set that is not exactly like the training set, or different enough that error rates are going to be easy to see.
- Ensure you have enough data.

6) Ensemble learning is a technique in machine learning which takes the help of several base models and combines their output to produce an optimized model. This type of machine learning algorithm helps in improving the overall performance of the model. Here the base model which is most commonly used is the Decision tree classifier. A decision tree basically works on several rules and provides a predictive output, where the rules are the nodes and their decisions will be their children and the leaf nodes will constitute the ultimate decision.

Different types of ensembles, but our major focus will be on the below two types:

- Bagging - Bagging is an ensemble technique that helps in reducing variance in our model and hence avoids overfitting. Bagging is an example of the parallel learning algorithm

- Boosting - Boosting is a sequential learning algorithm that helps in reducing bias in our model and variance in some cases of supervised learning.

7) Let see the comparison between Bagging and Boosting.

Bagging	**Boosting**
Objective to decrease variance, not bias.	Objective to decrease bias, not variance.
Each model is built independently.	New models are affected by the implementation of the formerly developed model.
It is the simplest way of connecting predictions that belong to a similar type.	It is a method of connecting predictions that belong to multiple types.
Bagging tries to tackle the over-fitting problem.	Boosting tries to reduce bias.
Several training data subsets are randomly drawn with replacement from the whole training dataset.	Each new subset includes the components that were misclassified by previous models.
Bagging can solve the over-fitting problem.	Boosting can boost the over-fitting problem.

8)

The out-of-bag error is the average error for each predicted outcome calculated using predictions from the trees that do not contain that data point in their respective bootstrap sample. This way, the Random Forest model is constantly being validated while being trained. Let us consider the *j*th decision tree DTj that has been fitted on a subset of the sample data. For every training observation or sample $zi=(xi,yi)$ not in the sample subset of DTj where xi is the set of features and yi is the target, we use DTj to predict the outcome oi for xi. The error can easily be computed as $|oi-yi|$.
The out-of-bag error is thus the average value of this error across all decision trees.

9) K-Fold Cross Validation is a common type of cross validation that is widely used in machine learning.
K-fold cross validation is performed as per the following steps:

1. Partition the original training data set into k equal subsets. Each subset is called a fold. Let the folds be named as f_1, f_2, …, f_k .
2. For i = 1 to i = k
    1. Keep the fold f_i as Validation set and keep all the remaining *k-1* folds in the Cross validation training set.
    2. Train your machine learning model using the cross validation training set and calculate the accuracy of your model by validating the predicted results against the validation set.
3. Estimate the accuracy of your machine learning model by averaging the accuracies derived in all the *k* cases of cross validation.

In the k-fold cross validation method, all the entries in the original training data set are used for both training as well as validation. Also, each entry is used for validation just once.

10) *Hyperparameters in Machine learning are those parameters that are explicitly defined by the user to control the learning process.* These hyperparameters are used to improve the learning of the model, and their values are set before starting the learning process of the model. "*Hyperparameters are defined as the parameters that are explicitly defined by the user to control the learning process."*

Hyperparameters are the knobs or settings that can be tuned before running a training job to control the behavior of an ML algorithm. They can have a big impact on model training as it relates to training time, infrastructure resource requirements (and as a result cost), model convergence and model accuracy.

11) When training a deep neural network with gradient descent and backpropagation, we calculate the partial derivatives by moving across the network from the final output layer to the initial layer. With the chain rule, layers that are deeper in the network go through continuous matrix multiplications to compute their derivatives.

In a network of n hidden layers, n derivatives will be multiplied together. If the derivatives are large, then the gradient will increase exponentially as we propagate down the model until they eventually explode, and this is what we call the problem of exploding gradient. Alternatively, if the derivatives are small, then the gradient will decrease exponentially as we propagate through the model until it eventually vanishes, and this is the vanishing gradient problem.

>Vanishing Gradient
>Exploding Gradient
>Saddle Point

12) Logistic regression is a supervised learning algorithm which is mostly used to solve binary “**classification”** tasks although it contains the word “**regression**” . “Regression” contradicts with “classification” but the focus of logistic regression is on the word “logistic” referring to **logistic function** which actually does the classification task in the algorithm. Logistic regression is a simple yet very effective classification algorithm so it is commonly used for many binary classification tasks. Customer churn, spam email, website or ad click predictions are some examples of the areas where logistic regression offers a powerful solution. It is even used as an activation function for neural network layers. **Non-linear problems can't be solved with logistic regression because it has a linear decision surface**. Linearly separable data is rarely found in real-world scenarios. Good accuracy for many simple data sets and it performs well when the dataset is linearly separable.

13)

The Comparison

Loss Function:
The technique of Boosting uses various loss functions. In case of Adaptive Boosting or AdaBoost, it minimises the exponential loss function that can make the algorithm sensitive to the outliers. With Gradient Boosting, any differentiable loss function can be utilised. Gradient Boosting algorithm is more robust to outliers than AdaBoost.

Flexibility
AdaBoost is the first designed boosting algorithm with a particular loss function. On the other hand, Gradient Boosting is a generic algorithm that assists in searching the approximate solutions to the additive modelling problem. This makes Gradient Boosting more flexible than AdaBoost.

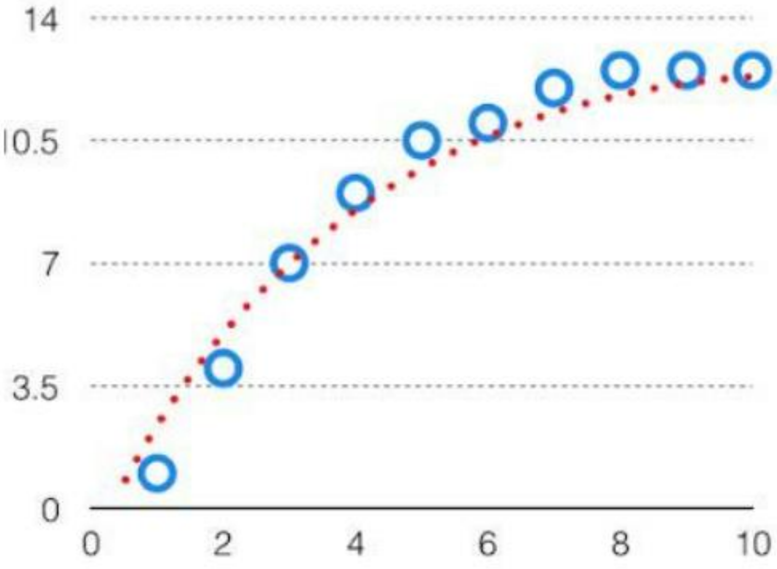
Benefits
AdaBoost minimises loss function related to any classification error and is best used with weak learners. The method was mainly designed for binary classification problems and can be utilised to boost the performance of decision trees. Gradient Boosting is used to solve the differentiable loss

function problem. The technique can be used for both classification and regression problems.

Shortcomings
In the case of Gradient Boosting, the shortcomings of the existing weak learners can be identified by gradients and with AdaBoost, it can be identified by high-weight data points.

14) 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.

15) SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example ***linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.***
Introduce Kernel functions for sequence data, graphs, text, images, as well as

vectors. The most used type of kernel function is **RBF.** Because it has localized and finite response along the entire x-axis.

The kernel functions return the inner product between two points in a suitable feature space. Thus by defining a notion of similarity, with little computational cost even in very high-dimensional spaces.