Sol.-1

R-Squared A statistical measure that determines the proportion of variance in the dependent variable that can be explained by the independent variable.
The residual sum of squares (**RSS) calculates the degree of variance in a regression model**. It estimates the level of error in the model's prediction. The smaller the residual sum of squares, the better your model fits your data; the larger the residual sum of squares, the worse. It is the sum of squares of the observed data minus the predicted data.

The **residual sum of squares (RSS) is the absolute amount of explained variation, whereas R-squared is the absolute amount of variation as a proportion of total variation**

R2 SQUARED means: it represents the proportion of the variance in your data which is explained by your model; the closer to one, the better the fit.

The residual sum of squares (RSS) is the sum of the squared distances between your actual versus your predicted values.

The actual number you get depends largely on the **scale** of your response variable. Taken alone, the RSS isn't so informative.

Sol.2

Residual sum of Squares (RSS) is defined as the sum of squares of the residual for each data point in the plot/data. It is the measure of the difference between the expected and the actual observed output.

Total Sum of Squares (TSS) is defined as the sum of errors of the data points from the mean of the response variable.

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.

TSS=ESS+RSS

Sol.-3

While training a machine learning model, the model can easily be overfitted or under fitted. To avoid this, we use regularization in machine learning **to properly fit a model onto our test set**. Regularization techniques help reduce the chance of overfitting and help us get an optimal model.

Sol.4

The Gini Index or Gini Impurity is calculated by **subtracting the sum of the squared probabilities of each class from one**. It favours mostly the larger partitions and are very simple to implement. In simple terms, it calculates the probability of a certain randomly selected feature that was classified incorrectly.

Sol.5

Out of all machine learning techniques, decision trees are **amongst the most prone to overfitting**. No practical implementation is possible without including approaches that mitigate this challenge.

Reasons for Overfitting are as follows:

1. High variance and low bias
2. The model is too complex
3. The size of the training data

Sol.6

The ensemble method is a machine learning technique that combines multiple models to produce better results than any single model. Ensemble learning algorithms work by training multiple models on the same data and then combining the models to create a more accurate prediction. The key to making this work is to train the models so that they are different from each other, but still produce similar results. This way, the ensemble can provide a more accurate prediction than any single model.

There are many different ways to create an ensemble, but the most common methods are bagging and boosting.

Sol.7

Ensemble Learning. These two decrease the variance of a single estimate as they combine several estimates from different models. So the result may be a model with higher stability. Let’s understand these two terms in a glimpse.

4. **Bagging**: It is a homogeneous weak learners’ model that learns from each other independently in parallel and combines them for determining the model average.
5. **Boosting**: It is also a homogeneous weak learners’ model but works differently from Bagging. In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm.

Sol.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 that has been fitted on a subset of the sample data. For every training observation or sample not in the sample subset of where is the set of features and is the target, we use to predict the outcome for . The error can easily be computed as .
The out-of-bag error is thus the average value of this error across all decision trees.

Sol.9

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.

Sol.10

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.

Sol.11

The learning rate can seen as step size, η. As such, gradient descent is taking successive steps in the direction of the minimum. If the step size η is too large, it can (plausibly) "jump over" the minima we are trying to reach, ie. we overshoot. This can lead to **osculations around the minimum or in some cases to outright divergence**.

Sol.12

Logistic regression is known and used as a linear classifier. It is used to come up with a hyper*plane* in feature space to separate observations that belong to a class from all the other observations that do *not* belong to that class. The decision boundary is thus *linear*. Robust and efficient implementations are readily available (e.g. scikit-learn) to use logistic regression as a linear classifier.

Sol.13

AdaBoost
AdaBoost or Adaptive Boosting is the first Boosting ensemble model. The method automatically adjusts its parameters to the data based on the actual performance in the current iteration. Meaning, both the weights for re-weighting the data and the weights for the final aggregation are re-computed iteratively.

In practice, this boosting technique is used with simple classification trees or stumps as base-learners, which resulted in improved performance compared to the classification by one tree or other single base-learner.

Gradient Boosting
Gradient Boost is a robust machine learning algorithm made up of Gradient descent and Boosting. The word 'gradient' implies that you can have two or more derivatives of the same function. Gradient Boosting has three main components: additive model, loss function and a weak learner.

Sol.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.