MACHINE LEARNING

1.

- R-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:

- RSS=∑i=1n(yi−y^i)2RSS=∑i=1n(yi−y^i)2

- The actual number you get depends largely on the scale of your response variable. Taken alone, the RSS isn't so informative.
- A higher R-squared value will indicate a more useful beta figurehigher the R-squared, the better the model fits your data

2. RSS

RSS is a statistical method used to detect the level of discrepancy in a dataset not revealed by regression. If the residual sum of squares results in a lower figure, it signifies that the regression model explains the data better than when the result is higher. In fact, if its value is zero, it's regarded as the best fit with no error at all.

$$\text{RSS} = \sum_{i=1}^{n}(y_i - \hat{y})^2$$

$$= \sum_{i=1}^{n}(y_i - (\alpha + \beta x_i))^2$$

TSS

The total sum of squares is a variation of the values of a dependent variable from the sample mean of the dependent variable. Essentially, the total sum of squares quantifies the total variation in a sample. It can be determined using the following formula:

$$\text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

ESS

- The explained sum of squares (ESS) is the sum of the squares of the deviations of the predicted values from

$$\mathrm{ESS} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 .$$

the mean value of a response variable, in a standard regression model

total sum of squares (TSS) = explained sum of squares (ESS) + residual sum of squares (RSS).

3.
In Machine Learning we often divide the dataset into training and test data, the algorithm while training the data can either
1) learn the data too well, even the noises which is called over fitting
2) 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.
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

5.
Yes,unregularized decision-trees prone to overfitting.
In decision trees, over-fitting occurs when the tree is designed so as to perfectly fit all samples in the training data set. Thus it ends up with branches with strict rules of sparse data. Thus this effects the accuracy when predicting samples that are not part of the training set.

One of the methods used to address over-fitting in decision tree is called pruning which is done after the initial training is complete. In pruning, you trim off the branches of the tree, i.e., remove the decision nodes starting from the leaf node such that the overall accuracy is not disturbed

6.

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

1. 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.
2. 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

- While Bagging and boosting make seem similar due to the use of N learners in both techniques, They are inherently quite different. While the Bagging technique is a simple way of combining predictions of the same kind, boosting combines predictions that belong to different types.

- In Bagging, each model is created independent of the other, But in boosting new models, the results of the

previously built models are affected.

- Bagging gives equal weight to each model, whereas in Boosting technique, the new models are weighted based on their results.

- In boosting, new subsets of data used for training contain observations that the previous model misclassified. Bagging uses randomly generated training data subsets.

- Bagging tends to decrease variance, not bias. In contrast, Boosting reduces bias, not variance.

- The bagging technique tries to resolve the issue of overfitting training data, whereas Boosting tries to reduce the problem of Bias.

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.
- The OOB error is computed using the samples that were not included in the training of the individual trees. This is different from the error computed using the usual training and validation sets, which are used to tune the hyperparameters of the random forest.

9.
k-fold cross-validation is one of the most popular strategies widely used by data scientists. It is a data partitioning strategy so that you can effectively use your dataset to build a more generalized model. The main intention of doing any kind of machine learning is to develop a more generalized model which can perform well on unseen data involve splitting the dataset, but with different approaches.

- Using k-fold cross-validation for evaluating a model's performance

- Using k-fold cross-validation for hyperparameter tuning

10.

- 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.
- Hyperparameter tuning is an essential part of controlling the behavior of a machine learning model. If we don't correctly tune our hyperparameters, our estimated model parameters produce suboptimal results, as they don't minimize the loss function. This means our model makes more errors.

11.

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.

12.

We Can we use Logistic Regression for classification of Non-Linear Data because logistic regression only forms linear decision surface .if we use non linear data The reason is that the target label has no linear correlation with the features. In such cases, logistic regression can't predict targets with good accuracy .

13

1. An additive model where shortcomings of previous models were 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.

If the algorithm is too simple then it may be on high bias and low variance condition and thus is error-prone. If algorithms fit too complex 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.

15.

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.

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.

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

$F(x, xj) = \exp(-\text{gamma} * ||x - xj||^2)$

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.