MACHINE LEARNING

Q1 to Q15 are subjective answer type questions, Answer them briefly.

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?

Ans :- R-squared is a measure of what proportion of the variance in the value of the dependent or response variable is explained by the regression model built using one or more independent or predictor variables. Mathematically, the value of R-squared can be calculated as the following:

R-squared = sum of squares regression (SSR) / sum of squares total (SST)

Residual sum of squares also known as adjusted r-squared can be defined as the proportion of variance explained by the model while taking into account both the number of predictor variables and the number of samples used in the regression analysis. The adjusted r-squared increases only when adding an additional variable to the model improves its predictive capability more than expected by chance alone.

Mathematically, adjusted r-squared can be calculated as the function of R-squared in the following manner:

Adjusted r_squared SSR SST degrees of freedom

RSS represents the residual sum of squares or sum of squares residual error (SSE)

The adjusted R-squared takes into account the number of predictor variables and the number of records used while calculating the value of R-squared. Hence, it is a better measure than R-squared in terms of how much variance in the response variable is explained by the regression model. The adjusted R-squared is a better measure of how well the model actually fits the data than just the R-squared value, which can be misleading if there are many predictor variables included in the regression. It is important to use both measures when assessing a linear regression model.

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.

Ans :- The total sum of squares (TSS) measures how much variation there is in the observed data, while the residual sum of squares measures the variation in the error between the observed data and modeled values. In statistics, the values for the residual sum of squares and the total sum of squares (TSS) are oftentimes
compared to each other.
The residual sum of squares (RSS) is also known as the sum of squared estimate of errors (SSE).
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.It tells how much of the variation between observed data and predicted data is being explained by the model proposed. Mathematically, it is the sum of the squares of the difference between the predicted data and mean data.

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

Ans:- Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfittings or underfitting.

Using Regularization, we can fit our machine learning model appropriately on a given test set and hence reduce the errors in it.

4. What is Gini–impurity index?

Ans: - Gini Impurity index 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?

Ans:- Decision trees are prone to overfitting, especially when a tree is particularly deep. This is due to the amount of specificity we look at leading to smaller sample of events that meet the previous assumptions. This small sample could lead to unsound conclusions.
Decision tree splits your data set by applying some functions on it, in such a way that data set finally ends up in different homogeneous buckets, i.e a bucket belongs to only one single class. When we create decision tree, we need to regularize it. Regularization in terms of decision tree means to control the growth of the tree. When decision tree becomes too large they tend to over-fit. To avoid over-fitting, we regularize the tree. By doing so, the tree does not grow to its full potential, i.e, it gets restricted.

6. What is an ensemble technique in machine learning?

Ans:- Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model.
Ensemble methods aim at improving predictability in models by combining several models to make one very reliable model.
The most popular ensemble methods are boosting, bagging, and stacking.
Ensemble methods are ideal for regression and classification, where they reduce bias and variance to boost the accuracy of models.

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

Ans:- Difference between boosting and bagging techniques are:-

1) Boosting use several datasets to train the models with some replacements in datasets.
Bagging every time increase the weight of the dataset to train the next learner.
2) Boosting reduces bias in machine learning.
Bagging reduces variance and overfitting in machine learning.
3) Boosting is a sequential homogeneous model.
Bagging is a parallel homogeneous model.
4) Boosting is used when the classifier has high bias and is straightforward.
When the classifier has high variance and is not stable bagging is used.
5) Boosting increase observation weight on detecting error in prediction.
Bagging has same weight on observation.
6) In boosting every model is affected by the previously connected model.
In bagging all models work independently.
7) Adda Boost is an example of boosting.
Random Forest is an example of bagging.

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

Ans:- 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?

Ans :- K-fold Cross-Validation is when the dataset is split into a K number of folds and is used to evaluate the model's ability when given new data. K refers to the number of groups the data sample is split into. For example, if you see that the k-value is 5, we can call this a 5-fold cross-validation. Each fold is used as a testing set at one point in the process.

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

Ans :- 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. Note that the learning algorithm optimizes the loss based on the input data and tries to find an optimal solution within the given setting. However, hyperparameters describe this setting exactly.

Hyperparameter tuning takes advantage of the processing infrastructure of Google Cloud to test different hyperparameter configurations when training your model. It can give you optimized values for hyperparameters, which maximizes your model's predictive accuracy.

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

Ans :- Gradient descent is an optimization algorithm that estimates the error gradient for the current state of the model using examples from the training dataset, then updates the weights of the model using the back-propagation of errors algorithm, referred to as simply backpropagation.

The amount that the weights are updated during training is referred to as the step size or the "learning rate."

Specifically, the learning rate is a configurable hyperparameter used in the training of neural networks that has a small positive value, often in the range between 0.0 and 1.0.

The learning rate controls how quickly the model is adapted to the problem. Smaller learning rates require more training epochs given the smaller changes made to the weights each update, whereas larger learning rates result in rapid changes and require fewer training epochs.

A learning rate that is too large can cause the model to converge too quickly to a suboptimal solution, whereas a learning rate that is too small can cause the process to get stuck.

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

Ans :- Logistic Regression has traditionally been used as a linear classifier, i.e. when the classes can be separated in the feature space by linear boundaries. That can be remedied however if we happen to have a better idea as to the shape of the decision boundary…

Logistic regression is known and used as a linear classifier. It is used to come up with a hyperplane 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.

13. Differentiate between Adaboost and Gradient Boosting.

Ans :- Both are boosting algorithms which means that they convert a set of weak learners into a single strong learner. They both initialize a strong learner (usually a decision tree) and iteratively create a weak learner that is added to the strong learner. They differ on how they create the weak learners during the iterative process.

At each iteration, adaptive boosting changes the sample distribution by modifying the weights attached to each of the instances. It increases the weights of the wrongly predicted instances and decreases the ones of the correctly predicted instances. The weak learner thus focuses more on the difficult instances. After being trained, the weak learner is added to the strong one according to his performance (so-called alpha weight). The higher it performs, the more it contributes to the strong learner.

On the other hand, gradient boosting doesn't modify the sample distribution. Instead of training on a newly sample distribution, the weak learner trains on the remaining errors (so-called pseudo-residuals) of the strong learner. It is another way to give more importance to the difficult instances. At each iteration, the pseudo-residuals are computed and a weak learner is fitted to these pseudo-residuals. Then, the contribution of the weak learner (so-called multiplier) to the strong one isn't computed according to his performance on the newly distribution sample but using a gradient descent optimization process. The computed contribution is the one minimizing the overall error of the strong learner.

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

Ans :- 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.

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

Ans :- SVM is an algorithm that has shown great success in the field of classification. It separates the data into different categories by finding the best hyperplane and maximizing the distance between points. To this end, a kernel function will be introduced to demonstrate how it works with support vector machines. Kernel functions are a very powerful tool for exploring high-dimensional spaces.

Linear kernel- Linear Kernel is used when the data is Linearly separable, that is, it can be separated using a single Line. It is one of the most common kernels to be used. It is mostly used when there are a Large number of Features in a particular Data Set. One of the examples where there are a lot of features, is Text Classification, as each alphabet is a new feature. So we mostly use Linear Kernel in Text Classification.

Polynomial kernel- A polynomial kernel is a kind of SVM kernel that uses a polynomial function to map the data into a higher-dimensional space. It does this by taking the dot product of the data points in the original space and the polynomial function in the new space.

RBF kernel- The RBF kernel works by mapping the data into a high-dimensional space by finding the dot products and squares of all the features in the dataset and then performing the classification using the basic idea of Linear SVM.