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?**

Answer:

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. R-square is a comparison of the residual sum of squares (SSres) with the total sum of squares (SStot).
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.

2. **What are TSS (Total Sum of Squares), ESS (Explained Sum of Squares) and RSS (Residual Sumof Squares) in regression. Also mention the equation relating these three metrics with each other.**

Answer:

Total Sum of Squares (TSS) explains the variation between observations or dependent variable's values and its mean. A higher value indicates that the model does not fit the data well and vice versa.

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 model.

Residual Sum of Squares (RSS) is a statistical method used to measure the deviation in a dataset unexplained by the regression model. Residual or error is the difference between the observation's actual and predicted value. If the RSS value is low, it means the data fits the estimation model well, indicating the least variance.

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

Answer:

Regularization is one of the most important concepts of machine learning. It is a technique to prevent the model from overfitting by adding extra information to it. Sometimes the machine learning model performs well with the training data but does not perform well with the test data. It means the model is not able to predict the output when deals with unseen data by introducing noise in the output, and hence the model is called over fitted. This problem can be deal with the help of a regularization technique. Also, it maintains accuracy as well as a generalization of the model.

4. **What is Gini–impurity index?**

Answer:

Gini Index, also known as **Gini impurity**, calculates the amount of probability of a specific feature that is classified incorrectly when selected randomly. If all the elements are linked with a single class, then it can be called pure.

The Gini index varies between values 0 and 1, where 0 expresses the purity of classification, i.e. All the elements belong to a specified class or only one class exists there. And 1 indicates the random distribution of elements across various classes. The value of 0.5 of the Gini Index shows an equal distribution of elements over some classes.

The Gini Index is determined by deducting the sum of squared of probabilities of each class from one, mathematically, Gini Index can be expressed as:

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

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

Answer:

Yes, unregularized decision-trees prone to overfitting, especially when a tree is particularly deep. Overfitting can be one problem that describes if your model no longer generalizes well. Overfitting happens when the learning algorithm continues to develop hypotheses that reduce training set error at the cost of an increased test set error.

Two approaches to avoiding overfitting in building decision trees are:

i. Pre-pruning that stops growing the tree earlier, before it perfectly classifies the training set.
ii. Post-pruning that allows the tree to perfectly classify the training set, and then post prune the tree.

6. What is an ensemble technique in machine learning?

Answer:

Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model. Ensemble method also helps to reduce the variance in the predicted data, minimize the biasness in the predictive model and to classify and predict the statistics from the complex problems with better accuracy.

Types of Ensemble Methods:

1. **Bagging**: This ensemble method combines two machine learning models i.e., Bootstrapping and Aggregation into a single ensemble model.

2. **Boosting**: The boosting ensemble also combines different same type of classifier. Boosting is one of the sequential ensemble methods in which each model or classifier run based on features that will utilize by the next model.

3. **Stacking**: This method also combines multiple classifications or regression techniques using a meta-classifier or meta-model. The lower levels models are trained with the complete training dataset and then the combined model is trained with the outcomes of lower level models.

4. **Random Forest**: The random forest is slightly different from bagging as it uses deep trees that are fitted on bootstrap samples. The output of each tress is combined to reduce variance. While growing each tree, rather than generating a bootstrap sample based on observation in the dataset, we also sample the dataset based on features and use only a random subset of such a sample to build the tree.

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

Answer:

Difference between Bagging and Boosting are:

a. Bagging technique can be an effective approach to reduce the variance of a model, to prevent over-fitting and to increase the accuracy of unstable models. On the other hand, Boosting enables us to implement a strong model by combining a number of weak models together.

b. In contrast to bagging, samples drawn from the training dataset are not replaced back into the training set during the boosting exercise.

c. If you analyze the decision boundaries, known as stumps, that are computed by the Adaptive Boosting algorithm when compared with the Decision trees, you will note that the decision boundaries computed by AdaBoost can be very sophisticated. Although this can help us implement a strong predictive model, the ensemble learning increases the computational complexity compared to individual classifiers.

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

Answer:

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. It is an error estimation technique often used to evaluate the accuracy of a random forest and to select appropriate values for tuning parameters, such as the number of candidate predictors that are randomly drawn for a split, referred to as mtry. However, for binary classification problems with metric predictors it has been shown that the out-of-bag error can overestimate the true prediction error depending on the choices of random forests parameters

9. What is K-fold cross-validation?

Answer:

K-Fold Cross Validation is a common type of cross validation that is widely used in machine learning.

It is performed as per the following steps:

i. Partition the original training data set into k equal subsets. Each subset is called a fold. Let the folds be named as f1, f2, …, fk. (For i = 1 to i = k)
ii. Keep the fold fi as Validation set and keep all the remaining k-1 folds in the Cross-validation training set.
iii. Train machine learning model using the cross-validation training set and calculate the accuracy of model by validating the predicted results against the validation set.
iv. Estimate the accuracy of your machine learning model by averaging the accuracies derived in all the k cases of cross validation.
v. 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.
vi. Generally, the value of k is taken to be 10, but it is not a strict rule, and k can take any value.

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

Answer:

A **hyper parameter** is a parameter whose value is set before the learning process begins and it define the model architecture. Hyper parameters tuning is crucial as they control the overall behavior of a machine learning model. Every machine learning model will have different hyper parameters that can be set. Hyper-parameters are the process of finding the best model architecture.

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

Answer:

Large learning rates put the model at risk of overshooting the minima so it will not be able to converge exploding gradient. The learning rate can be 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, i.e., we overshoot. This can lead to osculation around the minimum or in some cases to outright divergence.

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

Answer:

No, we cannot use Logistic Regression for classification of Non-Linear Data. 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.

13. Differentiate between Adaboost and Gradient Boosting.

Answer:

Difference between **AdaBoost** & **Gradient Boosting** are as follows:

1. AdaBoost is the first designed boosting algorithm with a particular loss function. It minimizes the exponential loss function that can make the algorithm sensitive to the outliers. 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.

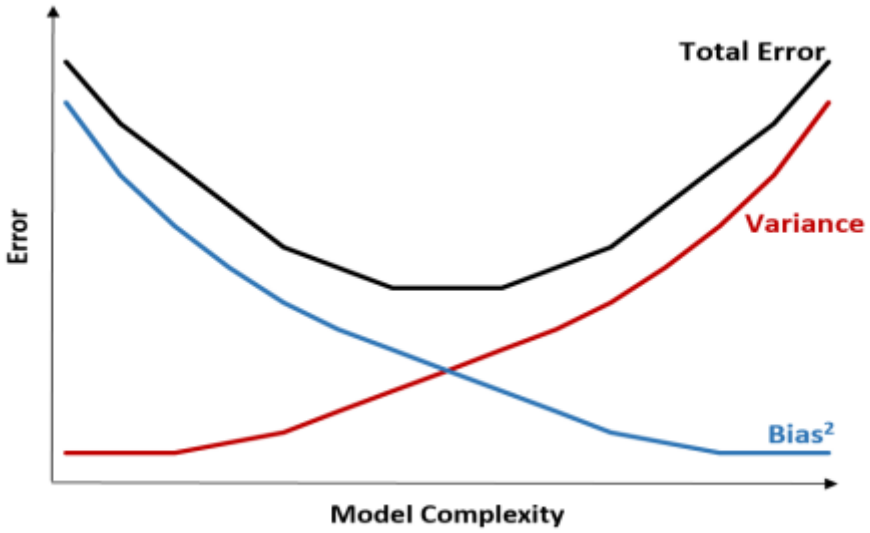
2. AdaBoost minimizes loss function related to any classification error and is best used with weak learners. The method was mainly designed for binary classification problems and can be utilized to boost the performance of decision trees. Any differentiable loss function can be utilized. Gradient Boosting algorithm is more robust to outliers than AdaBoost. Gradient Boosting is used to solve the differentiable loss function problem. The technique can be used for both classification and regression problems.

3. In AdaBoost, the shortcomings of the existing weak learners can be identified by high-weight data points. The shifting is done by up-weighting observations that were misclassified before. While in Gradient Boosting, the shortcomings of the existing weak learners can be identified by gradients. Gradient Boosting identifies the difficult observations by large residuals computed in the previous iterations.

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

Answer:

The bias-variance trade-off refers to the trade-off that takes place when we choose to lower bias which typically increases variance, or lower variance which typically increases bias. The following chart offers a way to visualize this trade-off:



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

Answer:

Linear Kernel: It can be used as a dot product between any two observations. The formula of linear kernel is as below –

K(x,xi)=sum(x∗xi)K(x,xi)=sum(x∗xi)

From the above formula, we can see that the product between two vectors say x & xi is the sum of the multiplication of each pair of input values.

Polynomial Kernel: It is more generalized form of linear kernel and distinguish curved or nonlinear input space. Following is the formula for polynomial kernel –

k(X,Xi)=1+sum(X∗Xi)^dk(X,Xi)=1+sum(X∗Xi)^d

Here d is the degree of polynomial, which we need to specify manually in the learning algorithm.

Radial Basis Function (RBF) Kernel: It is mostly used in SVM classification, maps input space in indefinite dimensional space. Following formula explains it mathematically −

K(x,xi)=exp(−gamma∗sum(x−xi^2))K(x,xi)=exp(−gamma∗sum(x−xi^2))

Here, gamma ranges from 0 to 1. We need to manually specify it in the learning algorithm. A good default value of gamma is 0.1.

The **linear**, **Polynomial** and **RBF or Gaussian kernel** are simply different in case of making the hyperplane decision boundary between the classes.

1. The kernel functions are used to map the original dataset (linear/nonlinear) into a higher dimensional space with view to making it linear dataset.

2. Usually linear and polynomial kernels are less time consuming and provides less accuracy than the RBF or Gaussian kernels.

3. The k cross validation is used to divide the training set into k distinct subsets. Then every subset is used for training and others k-1 are used for validation in the entire training phase. This is done for the better training of the classification task.