Q1. 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 –

Residual sum of square is defined as sum of squared difference between the actual and the predicted values and its value depend upon the scale of the target. It is also known as scale variant statistics. The lower the value of RSS, better the model of prediction, since it basically an error so minimum the residual better the model.

$$\textit{residual sum of squares} = \sum_{i=1}^{n} (y_i - \hat{y_i})^2$$

Where,

Y_i = Actual value

Y_i^ =Predicted value.

The values of RSS vary according to the scale of the target and its makes difficult to judge what might be the good value of RSS. So, we come up with scale invariant statistics i.e., R-squared.

R-squared statistic or coefficient of determination is a scale invariant statistic that provides the proportion of variation in target variable explained by the Linear regression model.

R-squared = (TSS-RSS)/TSS

= Explained variation/ Total variation

= 1 – Unexplained variation/ Total variation

Where, TSS stands for Total sum of squares that is basically the squares of the actual values and their means.

$$TSS = \sum (y_i - \bar{y})^2$$

Where,

Y_i = The actual value

$\bar{Y}$ = Mean of actual values

R-squared value always lies between 0 and 1. The closer to one, the better the fit of the model.

R-squared represents the complete proportion of variances and on the other hand RSS is not being so informative.

Q2. 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.

Answer-

TSS (total sum of squares)- It is the values of squares of the differences between the actual data and their means.

$$TSS = \sum (y_i - \bar{y})^2$$

Where,

Y_i = The actual value

Ȳ = Mean of actual values

ESS (Explained sum of squares)- It measures how much variation there is in a modelled values and it is calculated by the sum of squares of the deviations of the predicted values from the mean values of the response variable.

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

And

ESS = total sum of squares – residual sum of squares

Residual sum of square is defined as sum of squared difference between the actual and the predicted values and its value depend upon the scale of the target. It is also known as scale variant statistics. The lower the value of RSS, better the model of prediction, since it basically an error so minimum the residual better the model.

$$residual\ sum\ of\ squares = \sum_{i=1}^{n} (y_i - \widehat{y_i})^2$$

Where,

Yi = Actual value

Yi^ =Predicted value

Q3. What is the need of regularization in machine learning?

Answer –

Regularization is used to prevent the model from overfitting and underfitting by adding some extra information to it.

When train data is very much similar to test data, model accuracy will increase with learning the training data and it is called overfitted model and vice-versa is known as underfitted model. As a result, model accuracy will increase and when the data changes model accuracy will decrease because test data will be different from the trained data.

There are mainly two regularization techniques used in Machine learning:-

1. Lasso regression
2. Ridge regression

Q4. What is Gini–impurity index?

Answer-

Gini impurity is a function that determines how well a decision tree was split. Basically, it helps us to determine which splitter is best so that we can build a pure decision tree. Gini impurity ranges values from 0 to 0.5. It is one of the methods of selecting the best splitter; another famous method is Entropy which ranges from 0 to 1.

$$Gini(t) = 1 - \sum_{i=1}^{j} P(i|t)^2$$

Where,

The j represents the number of classes in the label, and

The P represents the ratio of class at the ith node.

Q5. Are unregularized decision-trees prone to overfitting? If yes, why?

Answer-

Hands-on implementation of pre-pruning, post pruning, and ensemble of Decision Trees, Decision Trees are a non-parametric supervised machine learning approach for classification and regression tasks. Overfitting is a common problem; a data scientist needs to handle while training decision tree models. Comparing to other machine learning algorithms, decision trees can easily overfit. Overfitting refers to the condition when the model completely fits the training data but fails to generalize the testing unseen data. Overfit condition arises when the model memorizes the noise of the training data and fails to capture important patterns. A perfectly fit decision tree performs well for training data but performs poorly for unseen test data. If the decision tree is allowed to train to its full strength, the model will overfit the training data. There are various techniques to prevent the decision tree model from overfitting.

Q6. What is an ensemble technique in machine learning?

Answer-

Ensemble methods are techniques that aim at improving the accuracy of results in models by combining multiple models instead of using a single model. The combined models increase the accuracy of the results significantly. This has boosted the popularity of ensemble methods in machine learning.

Main types of ensemble techniques: -

1. Bagging
   Bagging, the short form for bootstrap aggregating, it is mainly applied in classification and regression. It increases the accuracy of models through decision trees, which reduces variance to a large extent. The reduction of variance increases accuracy, eliminating overfitting, which is a challenge to many predictive models
2. Boosting
   Boosting is an ensemble technique that learns from previous predictor mistakes to make better predictions in the future. The technique combines several weak base learners to form one strong learner, thus significantly improving the predictability of models. Boosting works by arranging weak learners in a sequence, such that weak learners learn from the next learner in the sequence to create better predictive models
3. Stacking
   Stacking, another ensemble method, is often referred to as stacked generalization. This technique works by allowing a training algorithm to ensemble several other similar learning algorithm predictions. Stacking has been successfully implemented in regression, density estimations, distance learning, and classifications. It can also be used to measure the error rate involved during bagging.

Q7. What is the difference between Bagging and Boosting techniques?

Answer-

Bagging :-

Bagging, the short form for bootstrap aggregating, is mainly applied in classification and regression. It increases the accuracy of models through decision trees, which reduces variance to a large extent. The reduction of variance increases accuracy, eliminating overfitting, which is a challenge to many predictive models.

Bagging is classified into two types, i.e., bootstrapping and aggregation. Bootstrapping is a sampling technique where samples are derived from the whole population (set) using the replacement procedure. The sampling with replacement method helps make the selection procedure randomized. The base learning algorithm is run on the samples to complete the procedure.

Aggregation in bagging is done to incorporate all possible outcomes of the prediction and randomize the outcome. Without aggregation, predictions will not be accurate because all outcomes are not put into consideration. Therefore, the aggregation is based on the probability bootstrapping procedures or on the basis of all outcomes of the predictive models.

Bagging is advantageous since weak base learners are combined to form a single strong learner that is more stable than single learners. It also eliminates any variance, thereby reducing the overfitting of models. One limitation of bagging is that it is computationally expensive. Thus, it can lead to more bias in models when the proper procedure of bagging is ignored.

Boosting: -

Boosting is an ensemble technique that learns from previous predictor mistakes to make better predictions in the future. The technique combines several weak base learners to form one strong learner, thus significantly improving the predictability of models. Boosting works by arranging weak learners in a sequence, such that weak learners learn from the next learner in the sequence to create better predictive models.

Boosting takes many forms, including gradient boosting, Adaptive Boosting (AdaBoost), and XGBoost (Extreme Gradient Boosting). AdaBoost uses weak learners in the form of decision trees, which mostly include one split that is popularly known as decision stumps. AdaBoost's main decision stump comprises observations carrying similar weights.

Gradient boosting adds predictors sequentially to the ensemble, where preceding predictors correct their successors, thereby increasing the model's accuracy. New predictors are fit to counter the effects of errors in the previous predictors. The gradient of descent helps the gradient booster identify problems in learners' predictions and counter them accordingly.

XGBoost makes use of decision trees with boosted gradient, providing improved speed and performance. It relies heavily on the computational speed and the performance of the target model. Model training should follow a sequence, thus making the implementation of gradient boosted machines slow.

Q8. What is out-of-bag error in random forests?

Answer-

The RandomForestClassifier is trained using *bootstrap aggregation*, where each new tree is fit from a bootstrap sample of the training observations. The out-of-bag (OOB) error is the average error for each observation calculated using predictions from the trees that do not contain the observation in their respective bootstrap sample. This allows the RandomForestClassifier to be fit and validated whilst being trained.

Q9. What is K-fold cross-validation?

Answer-

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. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

Q10. What is hyper parameter tuning in machine learning and why it is done?

Answer-

Hyperparameter is a parameter whose value is set before the learning process begins and it define the model architecture. Hyperparameter tuning is an essential part of controlling the behavior of a machine learning model. Every machine learning will have different hyperparameters that can be set. Hyperparameters are the process of finding the best model architecture.

Q11. What issues can occur if we have a large learning rate in Gradient Descent?

Answer-

While building a deep learning project the most common problem we all face is choosing the correct hyper-parameters (often known as optimizers). This is critical as the hyper-parameters determine the expertise of the machine learning model. 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. If it is too small, we will need too many iterations to converge to the best values. So, using a good learning rate is crucial.

Q12. 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 features space while linear boundaries. That can be remedied however if we happen to have better idea as to shape of decision boundary. Logistic regression is known and used as a linear classifier.

Q13. Differentiate between Adaboost and Gradient Boosting.

Answer-

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 reweighting 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 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. The technique yields a direct interpretation of boosting methods from the perspective of numerical optimization in a function space and generalizes them by allowing optimization of an arbitrary loss function.

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

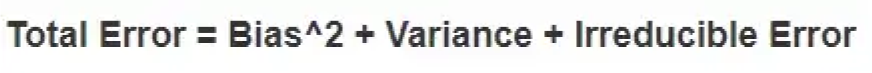
Q14. What is bias-variance trade off in machine learning?

Answer-

If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand if our model has large number of parameters then it's going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data.

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.

To build a good model, we need to find a good balance between bias and variance such that it minimizes the total error.



Total Error = Bias^2 + Variance + Irreducible Error

An optimal balance of bias and variance would never overfit or underfit the model.

Q15. Give short description each of Linear, RBF, Polynomial kernels used in SVM.

Answer-

Linear Kernels- 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.

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.

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

The linear, polynomial and RBF or Gaussian kernel are simply different in case of making the hyperplane decision boundary between the classes. The kernel functions are used to map the original dataset (linear/nonlinear) into a higher dimensional space with view to making it linear dataset. Usually linear and polynomial kernels are less time consuming and provides less accuracy than the RBF or Gaussian kernels. 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.