MACHINE LEARNING

Q.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. Residual Sum of Squares--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.

R-squared--R-Squared (R² or the coefficient of determination) is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable. In other words, r-squared shows how well the data fit the regression model (the goodness of fit).

Residual Sum of Squares is a better measure of goodness of fit model in regression because it 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.

Q.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.

AnsThe Explained SS tells you how much of the variation in the dependent variable your model explained.

Explained SS = Σ(Y-Hat – mean of Y)2.

The residual sum of squares tells you how much of the dependent variable’s variation your model did not explain. It is the sum of the squared differences between the actual Y and the predicted Y:

Residual Sum of Squares = Σ e2

The Total SS (TSS or SST) tells you how much variation there is in the dependent variable.

Total SS = Σ(Yi – mean of Y)2.

Note: Sigma (Σ) is a mathematical term for summation or “adding up.” It’s telling you to add up all the possible results from the rest of the equation.

TSS=RSS+ESS is the required equation

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

Ans. 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 overfitted. This problem can be deal with the help of a regularization technique.

This technique can be used in such a way that it will allow to maintain all variables or features in the model by reducing the magnitude of the variables. Hence, it maintains accuracy as well as a generalization of the model.

It mainly regularizes or reduces the coefficient of features toward zero. In simple words, "In regularization technique, we reduce the magnitude of the features by keeping the same number of features."

Q.4. What is Gini–impurity index?

Ans. Gini Index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen.

But what is actually meant by ‘impurity’?

If all the elements belong to a single class, then it can be called pure. The degree of Gini Index varies between 0 and 1,

where,

'0' denotes that all elements belong to a certain class or there exists only one class (pure), and

'1' denotes that the elements are randomly distributed across various classes (impure).

A Gini Index of '0.5 'denotes equally distributed elements into some classes.

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

Ans. Unregularised decision-trees are prone to overfitting because 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.

Q.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. To better understand this definition lets take a step back into ultimate goal of machine learning and model building. This is going to make more sense as I dive into specific examples and why Ensemble methods are used.

Types

BAGGing, or Bootstrap AGGregating. BAGGing gets its name because it combines Bootstrapping and Aggregation to form one ensemble model. Given a sample of data, multiple bootstrapped subsamples are pulled. A Decision Tree is formed on each of the bootstrapped subsamples. After each subsample Decision Tree has been formed, an algorithm is used to aggregate over the Decision Trees to form the most efficient predictor.

2. Random Forest Models. Random Forest Models can be thought of as BAGGing, with a slight tweak. When deciding where to split and how to make decisions, BAGGed Decision Trees have the full disposal of features to choose from. Therefore, although the bootstrapped samples may be slightly different, the data is largely going to break off at the same features throughout each model. In contrary, Random Forest models decide where to split based on a random selection of features. Rather than splitting at similar features at each node throughout, Random Forest models implement a level of differentiation because each tree will split based on different features. This level of differentiation provides a greater ensemble to aggregate over, ergo producing a more accurate predictor

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

Ans. diff between bagging and boosting

Bagging is a method of merging the same type of predictions. Boosting is a method of merging different types of predictions.

Bagging decreases variance, not bias, and solves over-fitting issues in a model. Boosting decreases bias, not variance.

In Bagging, each model receives an equal weight. In Boosting, models are weighed based on their performance.

Models are built independently in Bagging. New models are affected by a previously built model’s performance in Boosting.

In Bagging, training data subsets are drawn randomly with a replacement for the training dataset. In Boosting, every new subset comprises the elements that were misclassified by previous models.

Bagging is usually applied where the classifier is unstable and has a high variance. Boosting is usually applied where the classifier is stable and simple and has high bias.

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

Ans. Out-of-bag (OOB) error, also called out-of-bag estimate, is a method of measuring the prediction error of random forests, boosted decision trees, and other machine learning models utilizing bootstrap aggregating (bagging). Bagging uses subsampling with replacement to create training samples for the model to learn from. OOB error is the mean prediction error on each training sample xi, using only the trees that did not have xi in their bootstrap sample.[1]

Bootstrap aggregating allows one to define an out-of-bag estimate of the prediction performance improvement by evaluating predictions on those observations that were not used in the building of the next base learner.

Q.9. What is K-fold cross-validation?

Ans. 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. One can build a perfect model on the training data with 100% accuracy or 0 error, but it may fail to generalize for unseen data. So, it is not a good model. It overfits the training data. Machine Learning is all about generalization meaning that model’s performance can only be measured with data points that have never been used during the training process. That is why we often split our data into a training set and a test set.

Data splitting process can be done more effectively with k-fold cross-validation. Here, we discuss two scenarios which involve k-fold cross-validation. Both involve splitting the dataset, but with different approaches.

1)Using k-fold cross-validation for evaluating a model’s performance

2)Using k-fold cross-validation for hyperparameter tuning

Q10. 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.

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.

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

Ans. 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.

If the learning rate is very large we will **skip the optimal solution**.

Q12. Can we use Logistic Regression for classification of Non-Linear Data? If not, why?

Ans. **Logistic regression is neither linear nor is it a classifier**. The idea of a "decision boundary" has little to do with logistic regression, which is instead a direct probability estimation method that separates predictions from decision

Q13. Differentiate between Adaboost and Gradient Boosting.

### Ans. AdaBoost -AdaBoost or Adaptive Boosting is the first [Boosting ensemble model](https://analyticsindiamag.com/ensemble-modeling-explained-through-music/). 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](https://arxiv.org/pdf/1403.1452.pdf) 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](https://analyticsindiamag.com/pseudo-labelling-a-guide-to-semi-supervised-learning/) 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 optimisation in a function space and generalises them by allowing optimisation of an arbitrary loss function.

**The Comparison**

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

**Flexibility**-AdaBoost is the first designed boosting algorithm with a particular loss function. On the other hand, 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.

**Benefits**-AdaBoost minimises 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 utilised to boost the performance of decision trees. Gradient Boosting is used to solve the differentiable loss function problem. The technique can be used for both classification and regression problems.

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

Ans  the bias–variance tradeoff is **the property of a model that the variance of the parameter estimated across samples can be reduced by increasing the bias in the estimated parameters**.

Bias is the simplifying assumptions made by the model to make the target function easier to approximate. Variance is the amount that the estimate of the target function will change given different training data. Trade-off is tension between the error introduced by the bias and the variance.

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

Ans The **Radial Basis Function (RBF)** kernel is one of the most powerful, useful, and popular kernels in the Support Vector Machine (SVM) family of classifiers. In this article, we’ll discuss what exactly makes this kernel so powerful, look at its working, and study examples of it in action. We’ll also provide code samples for implementing the RBF kernel from scratch in Python that illustrates how to use the RBF kernel on your own data sets.

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.

In machine learning, the polynomial kernel is a kernel function commonly used with support vector machines (SVMs) and other kernelized models, that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.