# 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 ofgoodness of fit model in regression and why?**

   Ans: 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.

   Whereas R-Squared is a statistical measure of fit that indicates how much variation of a dependent variable is explained by the independent variable(s) in a regression model. i.e. R-squared is the absolute amount of variation as a proportion of total variation.

   R-squared is better measure of goodness of fit model in regression out of the two because ($R^2$ 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). Where as the residual sum of squares can be zero. The smaller the residual sum of squares, the better your model fits your data; the greater the residual sum of squares, the poorer your model fits your data. A value of zero means your model is a perfect fit.

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.**

   Ans: **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:

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

   Where:

   - $y_i$ – the value in a sample
   - $\bar{y}$ – the mean value of a sample

   **ESS:**- The Explained Sum of Squares tells you how much of the variation in the dependent variable your model explained.

   ESS = Σ(Y-HAT– mean of Y)$^2$.

   **RSS**:- 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:

   RSS = Σ e2

   The three metrics can be related with each other using following equation:

   **Total sum of squares (TSS) = Explained sum of squares (ESS) + Residual sum of squares (RSS)**.

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 overfitting 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 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 or Gini Impurity is calculated by subtracting the sum of the squared probabilities of each class from one. It favors mostly the larger partitions and are very simple to implement.

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.

6. **What is an ensemble technique in machine learning?**

Ans: Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would. This has been the case in a number of machine learning competitions, where the winning solutions used ensemble methods.

An ensemble method is a technique which uses multiple independent similar or different models/weak learners to derive an output or make some predictions. For e.g. A random forest is an ensemble of multiple decision trees.

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

Ans: Bagging is a technique for reducing prediction variance by producing additional data for training from a dataset by combining repetitions with combinations to create multi-sets of the original data.

Boosting is an iterative strategy for adjusting an observation's weight based on the previous classification.

Bagging is the simplest way of combining predictions that belong to the same type while Boosting is a way of combining predictions that belong to the different types.

Bagging aims to decrease variance, not bias while Boosting aims to decrease bias, not variance.

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

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.

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

Ans: The out-of-bag (OOB) error is the average error for each calculated using predictions from the trees that do not contain in their respective bootstrap sample. This allows the RandomForestClassifier to be fit and validated whilst being trained .

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

Ans: 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.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

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.

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
    1. Take the group as a hold out or test data set
    2. Take the remaining groups as a training data set
    3. Fit a model on the training set and evaluate it on the test set
    4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores

Importantly, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set 1 time and used to train the model k-1 times.
The results of a k-fold cross-validation run are often summarized with the mean of the model skill scores. It is also good practice to include a measure of the variance of the skill scores, such as the standard deviation or standard error.

**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.
Hyperparameters are parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning. 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: 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.
Gradient Descent is too sensitive to the learning rate. If it is too big, the algorithm may bypass the local minimum and overshoot. If it too small, it might increase the total computation time to a very large extent.

12. **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.
    Non-linear problems can't be solved with logistic regression because it has a linear decision surface.

13. **Differentiate between Adaboost and Gradient Boosting**.
    Ans: 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**.**

    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.

    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.

    In the case of Gradient Boosting, the shortcomings of the existing weak learners can be identified by gradients and with AdaBoost, it can be identified by high-weight data points.

14. **What is bias-variance trade off in machine learning?**
    Ans: In statistics and machine learning, 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.

    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.

15. **Give short description each of Linear, RBF, Polynomial kernels used in SVM.**
    Ans: 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 radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification.

    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.