ASSIGNMENT – 5 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- Both R-squared and Residual Sum of Squares (RSS) are measures of goodness of fit in regression analysis, but they capture different aspects of the model's performance.

R-squared (also known as the coefficient of determination) measures the proportion of variation in the dependent variable that is explained by the independent variables in the model. In other words, it indicates how well the model fits the data, with values ranging from 0 to 1. Higher R-squared values indicate a better fit, as they mean that a larger proportion of the variation in the dependent variable is explained by the independent variables in the model.

On the other hand, RSS measures the total sum of squared differences between the actual values of the dependent variable and the predicted values by the model. It represents the amount of unexplained variation in the data, and lower RSS values indicate a better fit, as they mean that the model is able to explain more of the variation in the data.

Therefore, both measures are useful in evaluating the goodness of fit of a model, but they serve different purposes. R-squared is a useful measure to assess the overall fit of the model and to compare different models, while RSS is useful to identify the degree of the error in the model's predictions.

In general, a good model should have both a high R-squared value and a low RSS value, indicating that it explains a large proportion of the variation in the dependent variable and has a low degree of error in its predictions. However, in some cases, one measure may be more important than the other, depending on the research question and the nature of the data being analyzed.

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.

   Answer-In statistics, the **explained sum of squares** (**ESS**), alternatively known as the **model sum of squares** or **sum of squares due to regression** (**SSR** – not to be confused with the residual sum of squares (RSS) or sum of squares of errors), is a quantity used in describing how well a model, often a regression model, represents the data being modelled. In particular, the explained sum of squares measures how much variation there is in the modelled values and this is compared to the total sum of squares (TSS), which

measures how much variation there is in the observed data, and to the residual sum of squares, which measures the variation in the error between the observed data and modelled values.

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 — for example, $y_i = a + b_1x_{1i} + b_2x_{2i} + ... + \varepsilon_i$, where y_i is the i^{th} observation of the response variable, x_{ji} is the i^{th} observation of the j^{th} explanatory variable, a and b_j are coefficients, i indexes the observations from 1 to n, and ε_i is the i^{th} value of the error term. In general, the greater the ESS, the better the estimated model performs.

If and are the estimated coefficients, then

is the i^{th} predicted value of the response variable. The ESS is then:

where the value estimated by the regression line .[1]

In some cases (see below): 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?

Answer-Regularization in Machine Learning is used to minimize the problem of overfitting, the result is that the model generalizes well on the unseen data once overfitting is minimized.

To avoid overfitting, regularization discourages learning a more sophisticated or flexible model. Regularization will try to minimize a loss function by inducing penalty.

4. What is Gini–impurity index?

Answer-Gini Impurity 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?

Answer-Overfitting can be one problem that describes if your model no longer generalizes well.

Overfitting happens when any learning processing overly optimizes training set error at the cost test error. While it's possible for training and testing to perform equality well in cross validation, it could be as the result of the data being very close in characteristics, which may not be a huge problem. In the case of decision tree's they can learn a training set to a point of high granularity that makes them easily overfit. Allowing a decision tree to split to a granular degree, is the behavior of this model that makes it prone to learning every point extremely well — to the point of perfect classification — ie: overfitting.

6. What is an ensemble technique in machine learning?

   Answer-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.

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

Answer-Owing to the proliferation of Machine learning applications and an increase in computing power, data scientists have inherently implemented algorithms to the data sets. The key to which an algorithm is implemented is the way bias and variance are produced. Models with low bias are generally preferred.
Organizations use supervised machine learning techniques such as decision trees to make better decisions and generate more profits. Different decision trees, when combined, make ensemble methods and deliver predictive results.
The main purpose of using an ensemble model is to group a set of weak learners and form a strong learner. The way it is done is defined in the two techniques: Bagging and Boosting that work differently and are used interchangeably for obtaining better outcomes with high precision and accuracy and fewer errors. With ensemble methods, multiple models are brought together to produce a powerful model.

8. What is out-of-bag error in random forests?
   Answer- The *out-of-bag* (OOB) error is the average error for each z_i calculated using predictions from the trees that do not contain z_i in their respective bootstrap sample. This allows the `RandomForestClassifier` to be fit and validated whilst being trained .

9. What is K-fold cross-validation?
   Answer-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.
10. What is hyper parameter tuning in machine learning and why it is done?
    Answer-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.
11. What issues can occur if we have a large learning rate in Gradient Descent?

Answer-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?

    Answer-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.

    Answer-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.

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

    Answer-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. The **bias–variance dilemma** or **bias–variance problem** is the conflict in trying to simultaneously minimize these two sources of error that prevent supervised learning algorithms from generalizing beyond their training set

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

Answer-The **Support Vector Machine** is a **supervised learning algorithm** mostly used for **classification** but it can be used also for **regression**. The main idea is that based on the labeled data (training data) the algorithm tries to find the **optimal hyperplane** which can be used to classify new data points. In two dimensions the hyperplane is a simple line.

Usually a learning algorithm tries to learn the **most common characteristics (what differentiates one class from another)** of a class and the classification is based on those representative characteristics learnt (so classification is based on differences between classes). The **SVM** works in the other way around. It **finds** the **most similar examples** between classes. Those will be the **support vectors**.