Assignment Machine Learning

Question1:- 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 1) R-squared and Residual Sum of Squares (RSS) are both measures of the goodness of fit of a regression model, but they are slightly different of each other. R-squared is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variables in a regression model. R-squared measures the extent to which changes in the dependent variable can be predicted by changes in the independent variable(s). Higher R-squared values indicate a better fit of the regression model to the data. So R-squared is often used to compare different models and select the best .

And Residual Sum of Squares (RSS) measures the difference between the observed values of the dependent variable and the predicted values by the model. It represents the sum of the squared differences between the actual and predicted values of the dependent variable. The goal is to minimize the residual sum of squares to obtain a better model fit.

Question 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 2) →The Total Sum Of Square(TSS) is the sum of squared differences between the observed *dependent* variable and the overall mean. Think of it as the dispersion of the observed variables around the mean similar to the variance in descriptive statistics. But TSS measures the total variability of a dataset, commonly used in regression analysis and ANOVA.

→The residual sum of squares(RSS) 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 Sum Of Square (ESS) is a statistical quantity used in modeling of a process. ESS gives an estimate of how well a model explains the observed data for the process .It tells how much of the variation between observed data and predicted data is being explained by the model proposed. Mathematically, it is the sum of the squares of the difference between the predicted data and mean data.

Question3:- What is the need of regularization in machine learning?

Answer 3) 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.

Question4:- What is Gini–impurity index?

Answer 4) 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.

Question5:- Are unregularized decision-trees prone to overfitting? If yes, why?

Answer5) 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.

Question 6:- What is an ensemble technique in machine learning?

Answer 6) ensemble technique helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of experts and to allow them to vote. Advantage of ensemble technique is improvement in predictive accuracy.

- The problems overcome by ensemble Technique is **Statistical Problem**
  The Statistical Problem arises when the hypothesis space

is too large for the amount of available data. Hence, there are many hypotheses with the same accuracy on the data and the learning algorithm chooses only one of them! There is a risk that the accuracy of the chosen hypothesis is low on unseen data!

- **Computational Problem**
  The Computational Problem arises when the learning algorithm cannot guarantees finding the best hypothesis.
- **Representational Problem**
  The Representational Problem arises when the hypothesis space does not contain any good approximation of the target class(es)

Question7:- What is the difference between Bagging and Boosting techniques?

Answer 7) Bagging and boosting are different ensemble techniques that use multiple models to reduce error and optimize the model. The bagging technique combines multiple models trained on different subsets of data, whereas boosting trains the model sequentially, focusing on the error made by the previous model. Bagging and Boosting are advanced ensemble methods in machine learning.

→Bagging

It Combines multiple models trained on different subsets of data.

Its main objective is to reduce variance by averaging out individual model error

For Data sampling use Bootstrap to create subsets of the data.

Each model serves equal weight in the final decision.

→Boosting

It train models sequentially, focusing on the error made by the previous model.

Its main objective is to Reduces both bias and variance by correcting misclassifications of the previous model

For Data sampling use Re-weights the data based on the error from the previous model, making the next models focus on misclassified instances.

Models are weighted based on accuracy, i.e., better-accuracy models will have a higher weight.

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

Answer 8) OOB (out-of-bag) errors are an estimate of the performance of a random forest classifier or regressor on unseen data. In scikit-learn, the OOB error can be obtained using the oob score attribute of the random forest classifier or regressor. The OOB error is computed using the samples that were not included in the training of the individual trees. This is different from the error computed using the usual training and validation sets, which are used to tune the hyperparameters of the random forest.

The OOB error can be useful for evaluating the performance of the random forest on unseen data. It is not always a reliable estimate of the generalization error of the model, but it can provide a useful indication of how well the model is performing.

Question 9:- What is K-fold cross-validation?

Answer 9) In K-fold cross-validation, the data set is divided into a number of K-folds and used to assess the model's ability as new data become available. K represents the number of groups into which the data sample is divided. For example, if you find the k value to be 5, you can call it 5-fold cross-validation. Each fold is used as a test set at some point in the process.

- Randomly shuffle the dataset.
- Divide the dataset into k folds
- For each unique group:
    - Use one fold as test data
    - Use remaining groups as training dataset
    - Fit model on training set and evaluate on test set

Question 10:- What is hyper parameter tuning in machine learning and why it is done?

Answer 10) Hyperparameter tuning is an essential part of controlling the behaviour of a machine learning model. If we

don’t correctly tune our hyperparameters, our estimated model parameters produce suboptimal results, as they don’t minimize the loss function. This means our model makes more errors. In practice, key indicators like the accuracy or the confusion matrix will be worse.

We use 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.

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

Answer 11) The learning rate is an important hyperparameter that greatly affects the performance of gradient descent. It determines how quickly or slowly our model learns, and it plays an important role in controlling both convergence and divergence of the algorithm. When the learning rate is too large, gradient descent can suffer from divergence. This means that weights increase exponentially, resulting in exploding gradients which can cause problems such as instabilities and overly high loss values.

In order to avoid these issues with different learning rates for each variable, we use adaptive techniques such as Adagrad and Adam which adjust their own learning rates throughout training based on real-time observations of parameters during optimization . These adaptive measures ensure better results than standard gradient descent while avoiding potential pitfalls in terms of either massive gains or slow losses due to misconfigured static global learning rates like those used with traditional gradient descent algorithms.

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

Answer 12) Logistic regression is known and used as a linear classifier. It is used to come up with a hyperlane 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 Robust and efficient implementations are readily available to use logistic regression as a linear classifier.

Question 13:- Differentiate between Adaboost and Gradient Boosting.

Answer 13) Gradient boosting produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the ensemble in a stage-wise fashion like other boosting methods do, but it generalizes them by

allowing optimization of an arbitrary differentiable loss function.

AdaBoost, short for Adaptive Boosting, is also a boosting technique that combines multiple weak classifiers into a strong one. The key difference is that AdaBoost focuses on training instances that are hard to classify by assigning them higher weights.

Both gradient boosting and AdaBoost are useful for solving regression and classification predictive modeling problems in data science and machine learning. They can build robust models out of simple components, avoid overfitting, and perform regularization.

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

Answer 14) While building the machine learning model, it is really important to take care of bias and variance in order to avoid overfitting and underfitting in the model. If the model is very simple with fewer parameters, it may have low variance and high bias. Whereas, if the model has a large number of parameters, it will have high variance and low bias. So, it is required to make a balance between bias and variance errors, and this balance between the bias error and variance error is known as **the Bias-Variance trade-off.**

For an accurate prediction of the model, algorithms need a low variance and low bias. But this is not possible because bias and variance are related to each other:

- If we decrease the variance, it will increase the bias.
- If we decrease the bias, it will increase the variance.

Bias-Variance trade-off is a central issue in supervised learning. Ideally, we need a model that accurately captures the regularities in training data and simultaneously generalizes well with the unseen dataset. Unfortunately, doing this is not possible simultaneously. Because a high variance algorithm may perform well with training data, but it may lead to overfitting to noisy data. Whereas, high bias algorithm generates a much simple model that may not even capture important regularities in the data. So, we need to find a sweet spot between bias and variance to make an optimal model.

Question 15:- Give short description each of Linear, RBF, Polynomial kernels used in SVM

Answer 15) →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

→Radial Basis Function Support Vector Machine (RBF SVM) is a powerful machine learning algorithm that can be used for classification and regression tasks. It is a non-parametric model that works well with non-linear and high-dimensional data.

RBF SVM works by mapping the input data into a higher-dimensional feature space, where the classes can be separated by a hyperplane. The algorithm uses a kernel function, such as the Radial Basis Function, to measure the similarity between pairs of data points in the feature space.

→A polynomial kernel is a kind of SVM kernel that uses a polynomial function to map the data into a higher-dimensional space. It does this by taking the dot product of the data points in the original space and the polynomial function in the new space. In a polynomial kernel for SVM, the data is mapped into a higher-dimensional space using a polynomial function. The dot product of the data points in the original space and the polynomial function in the new space is then taken. The polynomial kernel is often used in SVM classification problems where the data is not linearly separable. By mapping the data into a higher-dimensional space, the polynomial kernel can sometimes find a hyperplane that separates the classes.