MACHINE LEARNING – Worksheet 5

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?

   R-Squared: 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. R-squared shows how well the data fit the regression model.

   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. It estimates the variance in the residuals, or error term. The smaller the residual sum of squares, the better our model fits the data, the greater the residual sum of squares, the poorer our model fits the data.

   Each measures is better on its own based on the dataset which we work, To Conclude R-Squared is a better measure for goodness of fit in regression model as it shows the overall fit of the model comparing with RSS where it shows the variance of the residuals.

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

   Total Sum of Squares (TSS or SST) tells how much variation is there in the dependent variable.

   Explained Sum of Squares tells how much of the variation is there in the dependent variable that is explained by model.

   Residual sum of squares tells how much of the variation is there in the dependent variable that is not explained by model.

   Equation relating these three metrics with each other is based on the below formula

   Total Sum of Squares = Explained Sum of Squares + Residual Sum of Squares

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

   When we train our dataset in machine learning model there is a chance of overfit or underfit of the model. In order to prevent this, we use regularization technique to obtain the best model. Regularization is a technique used to reduce the errors by fitting the function appropriately on the given training set and avoid overfitting on test data.

4. What is Gini–impurity index?

   The Gini impurity measure is one of the methods used in decision tree algorithms to decide the optimal split from a root node. The Gini Index varies between 0 and 1, where 0 represents purity of the classification and 1 denotes random distribution of elements among various classes.

A Gini Index of 0.5 shows that there is equal distribution of elements across some classes. The Gini Index works on categorical variables and gives the results in terms of success or failure and hence performs only binary split.

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

Yes, unregularized decision-trees prone to overfitting, as it creates over complex trees which tends to overfit and it will not generalize well to new data. This is one of the disadvantage of decision tree.

6. What is an ensemble technique in machine learning?

Ensemble technique combines several individual predictive models to seek better final predictive model. The three main classes of ensemble learning methods are bagging, stacking, and boosting,

- Bagging involves fitting many decision trees on different samples of the same dataset and averaging the predictions.
- Stacking involves fitting many different models types on the same data and using another model to learn how to best combine the predictions.
- Boosting involves adding ensemble members sequentially that correct the predictions made by prior models and outputs a weighted average of the predictions.

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

Bagging	**Boosting**
Bagging is a homogeneous weak learners' model that learns from each other independently in parallel and combines them for determining the model average.	Boosting is also a homogeneous weak learners' model that learn sequentially and adaptively to improve model predictions of a learning algorithm.
Bagging attempts to tackle the over-fitting issue.	Boosting tries to reduce bias.
If the classifier is unstable (high variance), then we need to apply bagging.	If the classifier is steady and straightforward (high bias), then we need to apply boosting.
Every model receives an equal weight.	Models are weighted by their performance.
Objective to decrease variance, not bias.	Objective to decrease bias, not variance.
It is the easiest way of connecting predictions that belong to the same type.	It is a way of connecting predictions that belong to the different types.

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

Random Forest is one of the machine learning algorithms that use bootstrap aggregation, where each new tree is fit from a bootstrap sample of the training observations. The out-of-bag error is the average error for each predicted outcome calculated using predictions from the trees that do not contain that data point in their respective bootstrap sample. This way, the Random Forest model is constantly being validated while being trained.

9. What is K-fold cross-validation?

k-fold cross validation signifies the data set splits into a K number. It divides the dataset at the point where the testing set utilizes each fold. Let's understand the concept with the help of 5-fold cross-validation or K+5. In this scenario, the method will split the dataset into five folds. The model uses the first fold in the first iteration to test the model. It uses the remaining data sets to train the model. The second fold helps in testing the dataset and other support with the training process. The same process repeats itself till the testing set uses every fold from the five folds.

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

A hyperparameter is a parameter whose value is set before the learning process begins. Hyperparameters tuning is crucial as they control the overall behavior of a machine learning model. There are various strategies to tune hyperparameters for Machine learning models. Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. The two best strategies for Hyperparameter tuning are:

GridSearchCV: This method tries every possible combination of each set of hyper-parameters from a grid of hyperparameters values. Using this method, we can find the best set of values in the parameter search space. This usually uses more computational power and takes a long time to run since this method needs to try every combination in the grid size.

RandomizedSearchCV: The main difference in the RandomizedSearch CV, when compared with GridCV, is that instead of trying every possible combination, this chooses the hyperparameter sample combinations randomly from grid space. Because of this reason, there is no guarantee that we will find the best result like Grid Search. But, this search can be extremely effective in practice as computational time is very less.
The computational time and model performs mainly depends on the n_iter value. Because this value specifies how many times the model should search for parameters. If this value is high, there is a better chance of getting better accuracy, but also this comes with more computational power.

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

Gradient descent is an optimization algorithm that's used when training a machine learning model. It's based on a convex function and tweaks its parameters iteratively to minimize a given function to its local minimum.
For the gradient descent algorithm to reach the local minimum we must set the learning rate to

an appropriate value, which is neither too low nor too high. This is important because if the learning rate is large the steps it takes will be too big and it may not reach the local minimum because it bounces back and forth between the convex function of gradient descent. If we set given learning rate to a very small value, gradient descent will eventually reach the local minimum but that may take a while.

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

    Logistic Regression has traditionally been 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. Non-linear problems can't be solved with logistic regression because it has a linear decision surface.

13. Differentiate between Adaboost and Gradient Boosting

Adaboost	**Gradient boosting**
The shift is made by up-weighting the observations that are miscalculated prior	It identifies complex observations by huge residuals calculated in prior iterations
The trees are called decision stumps.	The trees with week learners are constructed using a greedy algorithm based on split points and purity scores. The trees are grown deeper with eight to thirty-two terminal nodes. The week learners should stay a week in terms of nodes, layers, leaf nodes, and splits
Every classifier has different weight assumptions to its final prediction that depend on the performance.	The classifiers are weighted precisely and their prediction capacity is constrained to learning rate and increasing accuracy
It gives values to classifiers by observing determined variance with data. Here all the week learners possess equal weight and it is usually fixed as the rate for learning which is too minimum in magnitude.	It develops a tree with help of previous classifier residuals by capturing variances in data. The final prediction depends on the maximum vote of the week learners and is weighted by its accuracy.
Maximum weighted data points are used to identify the shortcomings.	Here, the gradients themselves identify the shortcomings.
The exponential loss provides maximum weights for the samples which are fitted in worse conditions.	Gradient boosting cut down the error components to provide clear explanations and its concepts are easier to adapt and understand
Its focus on training the prior miscalculated observations and it alters the distribution of the dataset to enhance the weight on sample values which are hard for classification.	This method trains the learners and depends on reducing the loss functions of that week learner by training the residues of the model

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

The bias-variance tradeoff is a fundamental concept in machine learning and statistics that relates to the ability of a model to accurately capture the underlying patterns in a dataset. In essence, the bias-variance tradeoff refers to the balance between the complexity of a model and its ability to generalize to new, unseen data. The optimal level of complexity for a model depends on the specific problem and the available data and can be managed through techniques such as regularization and ensemble methods.

Bias refers to the degree to which a model is able to accurately capture the underlying patterns in a dataset.
The diversion from original answer is known as Bias. More the diversion, the higher the bias. A model with high bias is said to be underfitting.

Variance refers to the degree to which a model is able to adapt to new data.
The variance measures the "spread" of a distribution. A model with high variance is said to be overfitting.

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

Support vector machines (SVM) are a set of supervised learning methods used for classification, regression and outliers detection.
SVM can even work with a non-linear dataset and for this, we use "Kernel Trick". The function of the kernel trick is to map the low-dimensional input space and transforms into a higher dimensional space.

Radial Basis Function Kernel (RBF): The RBF kernel is the most widely used kernel concept to solve the problem of classifying datasets that cannot be separated linearly. This kernel is known to have good performance with certain parameters, and the results of the training have a small error value compared to other kernels.

Polynomial Kernel: A Polynomial Kernel is more generalized form of the linear kernel. In machine learning, the polynomial kernel is a kernel function suitable for use in support vector machines (SVM) and other kernelizations, where the kernel represents the similarity of the training sample vectors in a feature space. Polynomial kernels are also suitable for solving classification problems on normalized training datasets. This kernel is used when data cannot be separated linearly

Linear Kernel: 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.