ASSIGNMENT-5

1) R-Squared or R2 or r2_score is a better measure of goodness of a fit instead of Residual Sum of Square (RSS). This is because the R-Squared method explains the variance better than the RSS method. R-Squared values if higher tells that the model is fitting the data better.

RSS gives the total residual error in the trained model, and it does not explain the variance between dependent variables and independent variables. R-squared method in short explains how good the regression line if fit to the data hence it is more useful than RSS which only gives factual information like error.

2) 1) Total Sum of Squares (TSS):

The Total Sum of Squares is the sum of squared differences between the observed or actual dependent variables and the overall mean.

The mathematical equation is: $\sum_{i=1}^{n}(yi - \bar{y})^2$

yi- observed value.

$\bar{y}$ - mean

2) Explained Sum of Squares (ESS):

The Explained Sum of Squares is the sum of the squared differences between the predicted value and the mean of the dependent variable.

The mathematical equation is: $\sum_{i=1}^{n}(\hat{y}i - \bar{y})^2$

Where $\hat{y}_i$ – predicted value and $\bar{y}$ - Mean

3) Residual Sum of Squares (RSS):

The Residual Sum of Squares or RSS is the sum of the squared differences between the actual value or observed value and the predicted value.

The mathematical equation $\sum_{i=1}^{n}(yi - \hat{y}i)^2$

Where y_i – actual value and $\hat{y}_i$ – predicted value.

The equation between these is as follows:

TSS = ESS + RSS.

3) Regularization is a method used in Machine Learning so that the model properly fits the data, and the model does not overfits the data. It also helps to remove underfitting of the model.

Overfitting is when the model becomes more complex and learns more about the training data and noise whereas underfitting means the regression line or model is too simple. The next benefit of using regularization is maintaining collinearity and multi-collinearity.

The next one is Feature Selection which selects the best possible features of the dataset. Also, the complexity of the model can be controlled and improving regularization by penalizing more complex models which helps the model to capture underlying patterns in the data and not learning the noise.

4) Gini Index or Impurity is the calculation of probability for a random target variable being misclassified when it is randomly chosen. If the value of Gini Index if higher, then it is likely that the variable is misclassified and if Gini Index is lower then Variable is correctly classified.

The formula for calculating Gini Index or Impurity is:

Gini = $1 - \sum_{i=1}^{j} P(i)^2$

Where j- no. of classes in dataset

Pi- probability ratio.

5) The decision trees which are unregularized can lead to overfitting because of the following reasons:

A) High Variance:

The model learns about very minor things about the training dataset and also learns about noise and outliers which can lead to overfitting and perform bad on test dataset.

B) No Pruning:

If there is no limit set on the depth of the decision tree, then same issue happens here i.e. Learning more about data, noise, outliers, etc. The decision tree fails to find balance between bias and variance.

C) Complexity:

If there are thousands of features in the dataset, then model will become more complex means more decision nodes are created and that automatically increases the depth of the tree. So, this is another thing to look at.

D) One more reason is data distribution sensitivity like the decision tree can create the nodes or decisions or root nodes based on noise or outliers or patterns which are only present in that dataset.

6) Ensemble techniques are the machine learning techniques that create multiple decision trees or multiple training models and combine the results of each decision tree to come to a specific answer. There are reasons for that like many models will be given random training datasets, so each model will view the model from a separate view and will create decision nodes or root nodes differently which will increase the accuracy of the model.

There are two types of Ensemble learning techniques:

A) BAGGING: Bagging or Bootstrap Aggregation is a method that creates many models or trees parallelly and independent of each other and combines the results of each tree to draw a conclusion.
B) BOOSTING: In Boosting, the models are trained one by one in sequential order. First a model is trained and the whichever data this model has misclassified is given higher weights for the next model so that the next model will learn more about misclassified data to improve the accuracy of the following models.
C) STACKING: This technique develops multiple weak learners and combine them by training a meta model and by using multiple predictions to give desired output.

7)

BAGGING	BOOSTING
Bagging constructs multiple models or trees parallelly and independent of each other.	Boosting constructs multiple trees simultaneously meaning one after one and the next tree will be dependent on the outcome and accuracy of the previous tree.
Bagging reduces variance.	Boosting reduces bias.
Each model or tree is given equal weight.	Each corresponding model or tree is given weight according to the performance of the previous tree.
Each model is given training dataset randomly like random sampling with replacement and Random sampling without replacement.	The next model is given more data samples which are misclassified by the previous model.
Bagging reduces overfitting problem.	Boosting reduces underfitting problem.
Bagging uses aggregation techniques like maximum voting or average.	Boosting uses aggregation techniques like weighted majority.
Example: Random Forest.	Example: AdaBoost.

8) First, we need to know what is Out Of Bag score in order to find the OOB Error. So, let's check it out. The out of bag score is a validation method used to test the Random Forest model. As you know, the Random Forest model randomly selects the samples either with replacement or without replacement. So, there is a high chance that some data samples will not be selected for training the dataset in any of the models. These left behind samples of each model are called as Out Of Bage samples which will be used to validate the model after it is trained to check how the model is performing.

In short, OOB score is the number of correctly predicted rows from the out of bag samples. Around 37% of data is available in Random Forest model as OOB sample. You can check the oob score in Random Forest model with the help of oob_score_ attribute.

Then oob error is calculated as :

OOB ERROR = 1 – OOB_SCORE

The oob score is not the exact as the accuracy score. Random Forest is very useful algorithm for this technique.

9) K-Fold Cross Validation is a machine learning technique used to determine the performance and accuracy of the model. First, the data is divided into k subsets of samples. Let's consider the value of k as 5. Hence, we have 5-folds cross validation. Now, we must train 5 models. Consider there are 1000 samples in the dataset. For the first model we will have the first 20% of the data as testing data i.e. first 200 samples and the rest of the data for training purpose. We will build the model and evaluate it. Check the accuracy score and test it. Afterwards for the second model we will consider second 20% of the data i.e. from row 201- 400 as our testing data and the rest of dataset will be model training dataset. Again, we will check the accuracy and evaluate our model using various metrics.

Then, for third model 401 to 600 rows will be testing data and rest is training data. For fourth it will be 601 to 800 testing data and for fifth model the testing data will be 801 to 1000. Each model accuracy score and other metrics will be combined to give you the overall score. You can also each individual model's score as well.

10) Hyperparameter tuning is a machine learning technique used to improve the accuracy of a model by training the model using the algorithm given. A hyperparameter is a parameter whose value is set before the learning process begins. For explanation of hyperparameter tuning I will consider Random Forest Algorithm. When we train the model with default parameters, you will get a decent accuracy score. But, for some datasets you will need to alter some of these parameters' values.

For Random Forest Algorithm, you can give a list of values for "max_depth" parameter like [10,20,30], criterion like ['gini','entropy'], max_features which considers only that much best features for training, "min_samples_split" represents the minimum no. of samples required for splitting or taking a decision, n_estimators is the most important hyperparameter required. It will specify the no. of trees you want to build for each combination.

Now, you make a dictionary set of these values and each hyperparameter can have multiple values. Now, the real process begins. You have two options to train the model. The first method is GridSearchCV also known as Grid Search. This technique will be given arguments such as the model object, hyperparameter dictionary which you created, cross validation number, etc. Then this technique will train the model using every possible combination of hyperparameters which we pass to the algorithm. This technique is computationally expensive as there will be hundreds of combinations from even a 10 set of values. Each model will evaluate itself and the accuracy score will be recorded. When every model is trained, it will note down the best accuracy score is from which parameters. These values will be given as the best estimator to us. Now, you can train the machine learning model using that particular parameters and the accuracy of the model will be improved.

Second method is RandomizedSearchCV also known as Randomized Search which as the name suggests will only build models for a specific combination of parameters unlike the Grid Model which will build models for every combination. This technique is fast and cheap as well in terms of memory. The one downside of second method (RandomizedSearchCV) is that it may or may not improve the accuracy of the model as the best possible combination may not be trained as it selects only few combinations.

Hyperparameter tuning is used because it will improve the model performance, solve the overfitting problem, improve generalization which means model will adapt to real world or new data properly, customization of parameters of the algorithm to get better accuracy and to achieve robustness of the model.

11) Large learning rate causes the following issues:

A) Instability:

Large Learning Rate cause the function to fluctuate too much and prevent convergence to an optimal solution.

B) Unstable Gradients:

It may explode gradients by strengthening the gradients of a loss function making it difficult to converge for the algorithm.

C) Poor Generalization:

Training data can be fitted properly to a model which has a high learning rate but unseen or testing data can be improperly fitted or the model fails to predict properly.

D) Slow convergence:

High learning rate can slow down the convergence of the optimization algorithm because it will exceed the minimum of loss function. Then automatically it will take more time to converge.

12) No, we should not use Logistic Regression on non-linear data because it will lead to poor results. The reasons are as follows:

A) Linear Decision Boundary:

Logistic Regression can separate data using a straight lines or hyperplanes. So, complex decision boundaries may not be correctly drawn by Logistic Regression on non-linear data.

B) Underfitting:

As said earlier, the logistic regression model cannot understand complex boundary, so it will oversimplify the data and it will fail to understand the underlying patterns in the dataset.

C)

The Logistic Regression works on the assumption of that there is a linear relationship between target variable and independent variables. So, the model will give incorrect results.

13)

ADABOOST	GRADIENT BOOSTING
Adaboost gives more weights to the misclassified data in the next model or tree.	Gradient Boosting works on residuals error instead of assigning weights to the next model.
Adaboost can lead to overfitting as it captures noise and outliers.	Gradient Boosting is more robust to noise and outliers as it uses residual error instead of weights.
It is computationally expensive.	Gradient Boosting is less computationally expensive than AdaBoost
AdaBoost cannot change the Learning Rate.	Gradient Boosting can change the learning rate.
AdaBoost uses exponential loss function.	Gradient Boosting uses differential loss function.
AdaBoost is a faster algorithm	Gradient Boosting is a little bit slower than AdaBoost.

14) Bias is the difference between predictions of machine learning model and the correct values. When the model has high bias, it will lead to underfitting of data and the model does not capture any underlying patterns of the data. High bias leads to errors in training as well as testing data.

Variance is the variability of model predictions at a given data point. A model with high variance leads to overfitting of the data. Sych model will perform very well on training data but poorly on testing data as it fails to catch generalized patterns in the data.

A bias variance trade-off arises because the model cannot be too complex or too simple at a given time. Hence, we need to find balance between bias and variance. If a model has high bias and low variance then it becomes too simple and if it has low bias and high variance then it becomes too complex. You must find the best possible complexity to avoid this issue. The best fit occurs at the Trade-off point where the balance is possible.

15) 1) LINEAR KERNEL:

Linear Kernel is the dot product of two given vectors or datapoints in the original dimensions. This is used when data is linearly separable, and hyperplane can be easily drawn on it.

The mathematical equation is: $k(x, y) = x^T \cdot y$

Where x and y are the two vectors or datapoint.

$k(x, y) = \phi(x) \cdot \phi(y) = x^T \cdot y$

2) RBF:

Gaussian Kernel or Radial Basis Function is a kernel used for non-linear data and regression problems. The RBF kernel maps the given data into a n dimensional feature space using non-linear transformation. It is a nonlinear kernel function that maps the input data into a higher-dimensional feature space using a Gaussian function.

The mathematical equation is:

$k(x, y) = e^{\frac{-(x-y)^2}{2\sigma^2}}$

Where x and y are datapoints and σ is the constant.

If y is small RBF is similar to Linear SVM. The RBF performs a dot product in R^{∞} which means multi-dimensional feature space. Because it can handle n dimensional data, it clearly classify data points into different classes.

3) POLYNOMIAL KERNEL:

Polynomial kernel is used in machine learning to generate non-linear boundaries. It transfers the given data into a higher dimensional feature space. The mathematical equation is:

$k(x, y) = (x^T \cdot y + c)^d$

Where x and y are the datapoint/vectors, c is the constant and d is the degree of the polynomial.

The degree of the polynomial is instrumental in determining the degree of the non-linear data.