ASSIGNMENT – 5

MACHINE LEARNING

Q1. 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?

Ans.:-R-squared is generally considered a better measure than RSS.

This is because R-squared provides an overall measure of the proportion of variance in the dependent variable that is explained by the model, whereas RSS only measures the magnitude of the residuals. Additionally, R-squared is a standardized measure and ranges from 0 to 1, making it easy to compare the fit of different models. In contrast, the magnitude of the RSS value depends on the scale of the dependent variable and can't be easily compared across models.

Q2. 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.

Ans.:-1)Total sum of squares(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:

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

yi=Observed dependent variable

$\bar{y}$=Mean of the dependent variable

2)Explained Sum of Squares(ESS):
The Explained SS tells you how much of the variation in the dependent variable your model explained.
Explained SS = Σ(Y-Hat – mean of Y)2.

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

yi=Observed dependent variable

$\hat{y}$=Mean of the dependent variable

3)Residual Sum of Squares(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:
Residual Sum of Squares = Σe^2

$e = yi-\hat{y}i$

Qu3. What is the need of regularization in machine learning?

Ans.:-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.

Qu4. **What is Gini–impurity index?**

Ans.:-Gini impurity measures how often a randomly chosen element of a set would be incorrectly labeled if it were labeled randomly and independently according to the distribution of labels in the set. It is calculated by multiplying the probability that a given observation is classified into the correct class and sum of the probabilities when that particular observation is classified into the wrong class.
Gini impurity value lies between 0 and 1. o being no impurity and 1 denoting random distributor. The node for which the Gini impurity is least is selected as the root node to split.

Qu5. Are unregularized decision-trees prone to overfitting? If yes, why?

Ans.:- Decision trees are prone to overfitting.
Especially when a decision 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.
An example of this could be predicting if the Boston Celtics will beat the Miami Heat in tonight's basketball game. The first level of the tree could ask if the Celtics are playing home or away. The second level might ask if the Celtics have a higher win percentage than their opponent, in this case the Heat. The third level asks if the Celtic's leading scorer is playing? The fourth level asks if the Celtic's second leading scorer is playing. The fifth level asks if the Celtics are traveling back to the east coast from 3 or more consecutive road games on the west coast. While all of these questions may be relevant, there may only be two previous games where the conditions of tonight's game were met. Using only two games as the basis for our classification would not be adequate for an informed decision. One way to combat this issue is by setting a max depth. This will limit our risk of overfitting; but as always, this will be at the expense of error due to bias. Thus if we set a max depth of three, we would only ask if the game is home or away, do the Celtics have a higher winning percentage than their opponent, and is their leading scorer playing. This is a simpler model with less variance sample to sample but ultimately will not be a strong predictive model.

Qu6. What is an ensemble technique in machine learning?

Ans.:-Ensemble technique has a similar underlying idea where we aggregate predictions from a group of predictors, which may be classifiers or regressors and most of the times the prediction is better than the one obtained using a single predictor such algorithms are called ensemble methods and such predictors are called ensembles.
Example- In a game show instead of taking judgment from judges the opinion poll is considered and the contestants getting maximum vote from audience is will win the game.

In this technique basically the average or mean is considered thereby we see variance decreases when we use average of all the predictors.
Ensemble method take multiple small models and combine their predictions to obtain a more powerful predictive power. There are you very popular ensemble technique such as Bagging Boosting and stacking.

Qu7. What is the difference between Bagging and Boosting techniques?

Ans.:-Differences between Bagging and Boosting are as below;

i)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.

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

iii)In Baggiing each model receives equal weight whereas in Boosting models are weighted according to their performance.

iv)In Bagging each model is built independently whereas in Boosting new models are influenced by performance of previously built models.

v)In Bagging different training data subsets are randomly drawn with replacement from the entire training dataset. In Boosting every new subsets contains the elements that were misclassified by previous models.

vi)Bagging tries to solve over-fitting problem while Boosting tries to reduce bias.

vii)If the classifier is unstable (high variance), then we should apply Bagging. If the classifier is stable and simple (high bias) then we should apply Boosting.

viii)Bagging is extended to Random forest model while Boosting is extended to Gradient boosting.

Xi)In Bagging base classifiers are trained parallelly, were as in Boosting base classifiers are trained sequentially.
x)Example: The Random forest model uses Bagging and the AdaBoost uses Boosting techniques.

Qu8. What is out-of-bag error in random forests?

Ans.:-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.

Qu9. What is K-fold cross-validation?

Ans.:-K-fold cross validation helps to generalize the machine learning model which results in better prediction on unknown data, this significantly reduces underfitting as we are using most of the data for training(fitting) and also it significantly reduces overfitting, as most of the data is being used for validation set.
K-fold cross validation input data is divided into K number of folds, so the model will get trained and tested for K times. But for every iteration we use one fold as test data and rest for all training data; here for every iteration, data in training and test fold changes which adds to the effectiveness of this method

Qu10. What is hyper parameter tuning in machine learning and why it is done?

Ans.:-In machine learning model several parameters that need to be learned from the data by training a model with existing data we can fit the model parameters however there are parameters those are called hyperparameters cannot be directly learned from the regular training process. They are usually fixed before the actual training starts and these parameters have important properties of the model, such as complexity and how fast it should learn.
Hyperparameter tuning is, when we basically try to find those sets and values of hyperparameters which will give us model with maximum accuracy.

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

Ans.:- 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.

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

Ans.:-No we cannot use Logistic Regression for classification of Non-Linear Data.
Non-linear problems can't be solved with logistic regression because it has a linear decision surface.

Qu.13. Differentiate between Adaboost and Gradient Boosting.

Ans.:-1)The main difference therefore is that Gradient Boosting is a generic algorithm to find approximate solutions to the additive modeling problem, while AdaBoost can be seen as a special case with a particular loss function (Exponential loss function). Hence, gradient boosting is much more flexible.
2)AdaBoost can be interrupted from a much more intuitive perspective and can be implemented without the reference to gradients by reweighting the training samples based on classifications from previous learners.
3)In Adaboost, shortcomings are identified by high-weight data points while in Gradient Boosting, shortcomings of existing weak learners are identified by gradients.
4)Adaboost is more about 'voting weights' and Gradient boosting is more about 'adding gradient optimization'.

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

Ans.:-For any model, we have to find the perfect balance between Bias and Variance. This just ensures that we capture the essential patterns in our model while ignoring the noise present it in. This is called Bias-Variance Tradeoff. It helps optimize the error in the model and keeps it as low as possible.
An optimized model will be sensitive to the patterns in data, but at the same time will be able to generalize to new data. In this, both the bias and variance should be low so as to prevent overfitting and underfitting.

Q15. Give short description each of Linear, RBF, Polynomial kernels used in SVM.

Ans.-i)Linear SVMs use a linear decision boundary to separate the data points of different classes. When the data can be precisely linearly separated, linear SVMs are very suitable. This means that a single straight line (in 2D) or a hyperplane (in higher dimensions) can entirely divide the data points into their respective classes. A hyperplane that maximizes the margin between the classes is the decision boundary.

ii)RBF kernels are the most generalized form of kernelization and is one of the most widely used kernels due to its similarity to the Gaussian distribution. The RBF kernel function for two points X_1 and X_2 computes the similarity or how close they are to each other. This kernel can be mathematically represented as follows:
$K=(X1,X2)=exp(- ||X_1 - X_2||/2\sigma^2)$
where,
1. ‘σ’ is the variance and our hyperparameter
2. $||X_1 - X_2||$ is the Euclidean (L_2-norm) Distance between two points X_1 and X_2

iii)Polynomial kernel SVM is a type of SVM that uses a polynomial function to transform the input data into a higher dimensional space. The polynomial kernel function takes the dot product of the input data points and adds a constant to the result, which is raised to a power specified by the degree parameter of the function. The result of this transformation is a set of new features that capture the non-linear relationships between the input data.
Polynomial kernel SVM works bY finding the hyperplane that best separates the data on the transformed space. The hyperplane is chosen such that it maximizes the margin between the classes, which is the distance between the hyperplane and the closest data points from each class. The hyperplane is also chosen to minimize the classification error on the training data.