Machine Learning Worksheet 5

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 ?

Answer- The R-squared statistic provides a measure of fit. It takes the form of a proportion—the proportion of variance explained—and so it always takes on a value between 0 and 1. In simple words, it represents how much of our data is being explained by our model.

Residual for a point in the data is the difference between the actual value and the value predicted by our linear regression model.Using the residual values, we can determine the sum of squares of the residuals also known as Residual sum of squares or RSS.

$$RSS = \sum (y_i - \hat{y}_i)^2$$

The lower the value of RSS, the better is the model predictions. Or we can say that – a regression line is a line of best fit if it minimizes the RSS value

So R-squared gives the degree of variability in the target variable that is explained by the model or the independent variables. If this value is 0.7, then it means that the independent variables explain 70% of the variation in the target variable.

If we had a really low RSS value, it would mean that the regression line was very close to the actual points. This means the independent variables explain the majority of variation in the target variable. In such a case, we would have a really high R-squared value.

R-squared = (TSS-RSS)/TSS

Where ,

TSS = Total Sum of Squares

RSS= Residual sum of squares

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?

Answer- **The Total Sum of Squares (TSS)-Total variation in target variable is the sum of squares of the difference between the actual values and their mean.**

$$TSS = \sum (y_i - \bar{y})^2$$

Residual sum of squares or RSS-Residual for a point in the data is the difference between the actual value and the value predicted by our linear regression model.

$$Residual = actual - predicted = y\_\hat{y}$$

Using the residual values, we can determine the sum of squares of the residuals also known as **Residual sum of squares** or RSS.

$$RSS = \sum (y_i - \hat{y}_i)^2$$

ESS (Explained Sum of Squares)- 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.

TSS = ESS + RSS.

Q3- What is the need of regularization in machine learning?

Answer- When training a machine learning model, the model can be easily overfitted or under fitted. **To avoid this, we use regularization in machine learning to properly fit the model to our test set. Regularization techniques help reduce the possibility of overfitting and help us obtain an optimal model.**

Q4- What is Gini–impurity index?
Answer4- 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.

Q5- Are unregularized decision-trees prone to overfitting? If yes, why?
Answer- Decision trees are prone to overfitting, especially when a 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.

Q6- What is an ensemble technique in machine learning?
Answer- Ensemble methods are techniques that aim at improving the accuracy of results in models by combining multiple models instead of using a single model. The combined models increase the accuracy of the results significantly. This has boosted the popularity of ensemble methods in machine learning. Ensemble methods are ideal for regression and classification, where they reduce bias and variance to boost the accuracy of models.
The most popular ensemble methods are Boosting, Bagging, and Stacking.

Q7- What is the difference between Bagging and Boosting techniques?
Answer-

Bagging	Boosting
1- Training Data subsets are drawn randomly with replacement from the entire dataset 2- Attempts to tacklethe overfitting isuue. 3- Bagging every model receives an equal weight 4- Bagging object to decrease variance not bias 5- Every model built independently	1-Each new subsets contains the components that were misclassified by previous models 2-Bossting tries to reduce biasness 3- models are weighted by their performances 4- Boosting Objective is to decrease Bias not variance 5-New models are affected by the performance of the previously developed model

---

Q8- What is out-of-bag error in random forests?
Answer- 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.

Q9- What is K-fold cross-validation?
Answer- Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

Q10- 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.
Hyperparameters are important because they directly control the behaviour of the training algorithm and have a significant impact on the performance of the model is being trained

---

Q11- What issues can occur if we have a large learning rate in Gradient Descent?
Answer-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. in order for Gradient Descent to work, we must set the learning rate to an appropriate value. This parameter determines how fast or slow we will move towards the optimal weights. If the learning rate is very large **we will skip the optimal solution**

Q12- 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. Robust and efficient implementations are readily available (e.g. scikit-learn) to use logistic regression as a linear classifier.

Q13- Differentiate between Adaboost and Gradient Boosting?
Answer-

AdaBoost or Adaptive Boosting	**Gradient Boost**
1-is the first Boosting ensemble model. The method automatically adjusts its parameters to the data based on the actual performance in the current iteration. Meaning, both the weights for re-weighting the data and the weights for the final aggregation are re-computed iteratively.	**1-is** a robust machine learning algorithm made up of Gradient descent and Boosting. The word 'gradient' implies that you can have two or more derivatives of the same function. Gradient Boosting has three main components: additive model, loss function and a weak learner
2-Adaptive Boosting or AdaBoost, it minimises the exponential loss function that can make the algorithm sensitive to the outliers	**2-**Gradient Boosting, any differentiable loss function can be utilised. Gradient Boosting algorithm is more robust to outliers than AdaBoost.
3-AdaBoost is the first designed boosting algorithm with a particular loss function.	**3-**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.
4-the shifting is done by up-weighting observations that were misclassified before	**4-**dentifies the difficult observations by large residuals computed in the previous iterations

---

Q14-What is bias-variance trade off in machine learning?
Answer-Bias is the simplifying assumptions made by the model to make the target function easier to approximate. Variance is the amount that the estimate of the target function will change given different training data. Trade-off is tension between the error introduced by the bias and the variance
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.**

Q15-Give short description each of Linear, RBF, Polynomial kernels used in SVM?
Answer- Kernel Function is a method used to take data as input and transform it into the required form of processing data. "Kernel" is used due to a set of mathematical functions used in Support Vector Machine providing the window to manipulate the data. So, Kernel Function generally transforms the training set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces

- **Linear Kernel**: used when data is linearly separable.
- **RBF-Same as Gaussian kernel function, adding radial basis method to improve the transformation.**
- **Polynomial Kernel:** It represents the similarity of vectors in the training set of data in a feature space over polynomials of the original variables used in the kernel**.**