MACHINE LEARNING 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 is better measure of goodness of fit represents the goodness of fit of a regression model. The ideal value for r-square is 1. The closer the value of r-square to 1, the better is the model fitted.
R-square is a comparison of the residual sum of squares (SS_{res}) with the total sum of squares (SS_{tot}). The total sum of squares is calculated by summation of squares of perpendicular distance between data points and the average line.
R square is calculated by:

$$R^2 = 1 - SS_{res} / SS_{tot}$$

The goodness of fit of regression models can be analysed on the basis of the R-square method. The more the value of r-square near 1, the better is the model.

The value of R-square can also be negative when the model fitted is worse than the average fitted model.

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.

The Total SS (TSS or SST) tells you how much variation there is in the dependent variable.

$$\text{Total SS} = \Sigma(Yi - \text{mean of } Y)2.$$

The Explained SS tells you how much of the variation in the dependent variable your model explained.

$$\text{Explained SS} = \Sigma(Y\text{-Hat} - \text{mean of } Y)2.$$

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:

$$\text{Residual Sum of Squares} = \Sigma\, e2$$

The relationship between the three types of sum of squares can be summarized by the following equation:

$$TSS = ESS + RSS$$

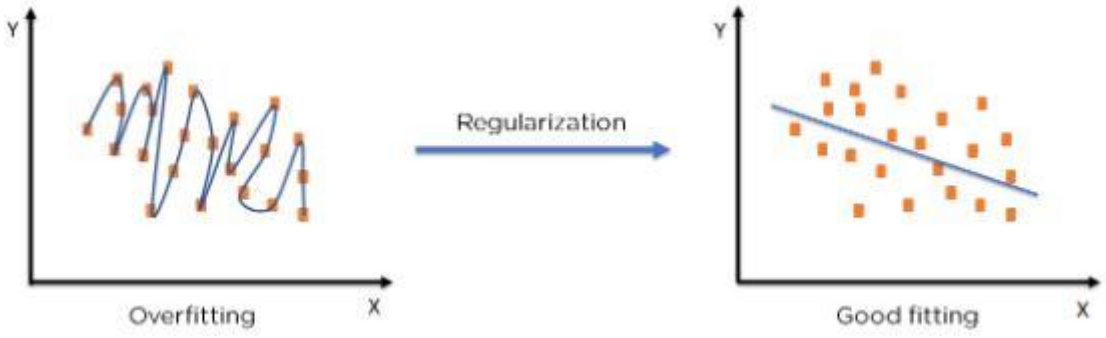
3. What is the need of regularization in machine learning?

To train the machine learning model, some data is been given to learn from. The process of plotting a series of data points and drawing the best fit line to understand the relationship between the variables is called Data Fitting. The model is best fit when it can find all necessary patterns in the data and avoid the random data points and unnecessary patterns called Noise.

A scenario where the machine learning model tries to learn from the details along with the noise in the data and tries to fit each data point on the curve is called Overfitting.

A scenario where a machine in the testing data nor predict or classify learning model can neither learn the relationship between variables a new data point is called Underfitting.

Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting.



Regularization on an over-fitted model

Using Regularization, the machine learning model can be fit appropriately on a given test set and hence reduce the errors in it.

4. What is Gini–impurity index?

The Gini Index is also known as Gini impurity. It is a measure of how mixed or impure a dataset is. The Gini impurity ranges between 0 and 1, where 0 represents a pure dataset and 1 represents a completely impure dataset (In a pure dataset, all the samples belong to the same class or category.

An attribute with a lower Gini index should be preferred i.e The lower the Gini impurity, the better the feature is for splitting the dataset.

Mathematically, The Gini Index is represented by

$$Gini\ impurity = 1 - \Sigma(p(i)^2)$$

Another commonly used formula is:

$$Gini\ impurity = 1 - \Sigma\left(p(i) * (1 - p(i))\right)$$

Where p(i) is the probability of a specific class and the summation is done for all classes present in the dataset.

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

- The training process is essentially building the tree. A key step is determining the "best" split. The procedure is as follows: split the data at each unique value in each feature, and choose the best one that yields the least disorder.
- Before building the tree, let's define decision node and leaf node. A decision node specifies the feature and value upon which it will split. It also points to its left and right children.
- A leaf node includes a dictionary similar to a counter object showing how many training examples for each class. This is useful to calculate the accuracy for training. In addition, it leads to the resulting prediction for each example that reaches this leaf.
- Given its structure, it is most convenient to construct the tree by recursion. The exit of recursion is a leaf node. This occurs when purity of the data cannot be increased through splitting. If a "best" split is found, this becomes a decision node. Similarly, it is recursively implemented to its left and right children.
- It results in training accuracy be 100% and the decision boundary is vague. Thus, the model is overfitting the training data. A decision tree will overfit the data if data is keep on splitting until the dataset couldn't be more pure. In other words, the model will correctly classify each and every example if it didn't stop splitting.

6. What is an ensemble technique in machine learning?

The ensemble methods in machine learning combine the insights obtained from multiple learning models to facilitate accurate and improved decisions. In learning models, noise, variance, and bias are the major sources of error. The ensemble methods in machine learning help minimize these error-causing factors, thereby ensuring the accuracy and stability of machine learning (ML) algorithms.

Example: Assume that an app is being developed an app for the travel industry. Before making the app public, crucial feedback is required on bugs and potential loopholes that are affecting the user experience.

The options for obtaining critical feedback:

1) Soliciting opinions from your parents, spouse, or close friends.

2) Asking your co-workers who travel regularly and then evaluating their response.

3) Rolling out the travel and tourism app in beta to gather feedback from non-biased audiences and the travel community.

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

- The Bagging technique is a simple way of combining predictions of the same kind, whereas boosting combines predictions that belong to different types.
- In Bagging, each model is created independent of the other, But in boosting new models, the results of the previously built models are affected.
- Bagging gives equal weight to each model, whereas in Boosting technique, the new models are weighted based on their results.
- In boosting, new subsets of data used for training contain observations that the previous model misclassified. Bagging uses randomly generated training data subsets.
- Bagging tends to decrease variance, not bias. In contrast, Boosting reduces bias, not variance.

- The bagging technique tries to resolve the issue of overfitting training data, whereas Boosting tries to reduce the problem of Bias.
- In Bagging, training data subsets are drawn randomly with a replacement for the training dataset. In Boosting, every new subset comprises the elements that were misclassified by previous models.
- Bagging is usually applied where the classifier is unstable and has a high variance. Boosting is usually applied where the classifier is stable and simple and has high bias.

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

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. The out-of-bag error is the average value of this error across all decision trees.

Example: Consider the dataset which records if it rains given the temperature and humidity:

S/N	TEMPERATURE	HUMIDITY	RAINED?
1	33	High	No
2	18	Low	No
3	27	Low	Yes
4	20	High	Yes
5	21	Low	No
6	29	Low	Yes
7	19	High	Yes

Assume that a random forest ensemble consisting of 5 decision trees DT1...DT5 is to be trained on the the dataset. Each tree will be trained on a random subset of the dataset. Assuming for DT1 that the randomly selected subset contains the first five samples of the dataset. Therefore, the last two samples 6 and 7 will be the out-of-bag samples on which DT1 will be validated. Continuing with the assumption, let the following table represent the prediction of each decision tree on each of its out-of-bag samples:

TREE	SAMPLE S/N	PREDICTION	ACTUAL	ERROR (ABS)
DT1	6	No	Yes	1
DT1	7	No	Yes	1
DT2	2	No	No	0
DT3	1	No	No	0
DT3	2	Yes	No	1

TREE	SAMPLE S/N	PREDICTION	ACTUAL	ERROR (ABS)
DT3	4	Yes	Yes	0
DT4	2	Yes	No	1
DT4	7	Yes	Yes	1
DT5	3	Yes	Yes	0
DT5	5	No	No	0

The out-of-bag error is the average error which is 0.5.
Only a subset of the decision trees in the ensemble is used in determining each error that is used to compute the out-of-bag score, it cannot be considered as accurate as a validation score on validation data. However, in cases such as this where the dataset is quite small and it is impossible to set aside a validation set, the out-of-bag error can prove to be a useful metric.

9. What is K-fold cross-validation?

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.

Cross-validation is primarily used to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

For K-fold Cross-validation:

1. Randomly divide a dataset into *k* groups, or "folds", of roughly equal size.

2. Choose one of the folds to be the holdout set. Fit the model on the remaining k-1 folds. Calculate the test MSE on the observations in the fold that was held out.

3. Repeat this process *k* times, using a different set each time as the holdout set.

4. Calculate the overall test MSE to be the average of the k test MSE's.

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

- In machine learning, a learning algorithm estimates model parameters for the given data set, then continues updating these values as it continues to learn. After learning is complete, these parameters become part of the model. For example, each weight and bias in a neural network is a parameter.

- Hyperparameters, on the other hand, are specific to the algorithm itself, so their values from the data cannot be calculated. Hyperparameters is used to calculate the model parameters. Different hyperparameter values produce different model parameter values for a given data set.

- 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 describes the learning of algorithm optimization and the loss based on the input data and tries to find an optimal solution within the given setting.

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

Gradient Descent is known for minimizing errors between actual and expected results. In machine learning, optimization is the task of minimizing the cost function parameterized by the model's parameters. The main objective of gradient descent is to minimize the convex function using iteration of parameter updates.

Challenges with the Gradient Descent

1. Local Minima and Saddle Point:

For convex problems, gradient descent can find the global minimum easily, while for non-convex problems, it is sometimes difficult to find the global minimum, where the machine learning models achieve the best results. Whenever the slope of the cost function is at zero or just close to zero, this model stops learning further. Apart from the global minimum, there occur some scenarios that can show this slop, which is saddle point and local minimum.

2. Vanishing and Exploding Gradient

If the model is trained with gradient descent and backpropagation, there can occur two more issues other than local minima and saddle point.

Vanishing Gradients: It occurs when the gradient is smaller than expected. During backpropagation, this gradient becomes smaller that causing the decrease in the learning rate of earlier layers than the later layer of the network. Once this happens, the weight parameters update until they become insignificant.

Exploding Gradient: It is just opposite to the vanishing gradient as it occurs when the Gradient is too large and creates a stable model. Further, in this scenario, model weight increases, and they will be represented as NaN. This problem can be solved using the dimensionality reduction technique, which helps to minimize complexity within the model.

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

- Logistic Regression is a statistical technique used to model the relationship between a dependent variable and one or more independent variables. The dependent variable in a Logistic Regression model is binary, meaning it can take on only two values (e.g.

yes/no, true/false). The independent variables, on the other hand, can be continuous or categorical. Logistic Regression models the relationship between the dependent variable and independent variables by using the logistic function to produce a probability that the dependent variable is a certain value, which can then be thresholded to make a final binary prediction.

- Example: Suppose you are tasked with predicting whether a customer will purchase a product based on their age and salary. In this case, the dependent variable is "purchase", which can be either "yes" or "no". The independent variables are "age" and "salary". The Logistic Regression model will use the logistic function to produce a probability that the customer will purchase the product based on their age and salary, which can then be thresholded to make a final binary prediction.
- Logistic Regression is not suitable for classification of non-linear data as it is complex to find the relationships between the dependent variable and independent variables.

13. Differentiate between Adaboost and Gradient Boosting.

Features	Gradient boosting	Adaboost
Model	It identifies complex observations by huge residuals calculated in prior iterations	The shift is made by up-weighting the observations that are miscalculated prior
Trees	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	The trees are called decision stumps.
Classifier	The classifiers are weighted precisely and their prediction capacity is constrained to learning rate and increasing accuracy	Every classifier has different weight assumptions to its final prediction that depend on the performance.
Prediction	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.	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.
Short-comings	Here, the gradients themselves identify the shortcomings.	Maximum weighted data points are used to identify the shortcomings.

Loss value	Gradient boosting cut down the error components to provide clear explanations and its concepts are easier to adapt and understand	The exponential loss provides maximum weights for the samples which are fitted in worse conditions.
Applications	This method trains the learners and depends on reducing the loss functions of that week learner by training the residues of the model	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

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

In machine learning, 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.

- The bias error is an error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).
- The variance is an error from sensitivity to small fluctuations in the training set. High variance may result from an algorithm modeling the random noise in the training data (overfitting).

The bias–variance tradeoff is a central problem in supervised learning. Ideally, id is preferable to choose a model that both accurately captures the regularities in its training data, but also generalizes well to unseen data; which is typically impossible to do both simultaneously.

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

The kernel functions are used to map the original dataset (linear/nonlinear) into a higher dimensional space with view to making it linear dataset.

- Radial Basis Function (RBF): It is used to perform transformation when there is no prior knowledge about data and It is used to perform transformation when there is no prior knowledge about data. It is a general-purpose kernel; used when there is no prior knowledge about the data.
- Polynomial Kernel: Polynomial features are derived features from given features in the data set. For example, a data set with a single feature x and let's find polynomial features with degree 3, then polynomial features will be x, x^2, x^3. If other features are also available, then, each of them would be converted similarly. 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. It is popular in image processing.

Equation is:

$$k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} \cdot \mathbf{x_j} + 1)^d$$

where d is the degree of the polynomial.

- Linear Kernel: used when 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. One of the examples where there are a lot of features, is **Text Classification**, as each alphabet is a new feature.

Usually linear and polynomial kernels are less time consuming and provides less accuracy than the rbf or Gaussian kernels.