Que1. 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 1. R squared is a better measure of goodness of fit mode in regression because R Square is calculated by the sum of squared of prediction error divided by the total sum of the square which replaces the calculated prediction with mean. R Square value is between 0 to 1 and a bigger value indicates a better fit between prediction and actual value.R Square is a good measure to determine how well the model fits the dependent variables. However, it does not take into consideration of overfitting problem

Que 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.

Ans2 . 1. Total sum of squares: 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:

Total SS = Σ(Yi – mean of Y)^2.

- y_i – the value in a sample
- $\bar{y}$ – the mean value of a sample

2. Explained sum of square: The explained sum of squares describes how well a regression model represents the modeled data. A higher regression sum of squares indicates that the model does not fit the data well.

ESS = Σ(Y-Hat – mean of Y)^2.

- $\hat{y}_i$ – the value estimated by the regression line
- $\bar{y}$ – the mean value of a sample

3. Residual sum of square: The residual sum of squares essentially measures the variation of modeling errors. In other words, it depicts how the variation in the dependent variable in a regression model cannot be explained by the model. Generally, a lower residual sum of squares indicates that the regression model can better explain the data, while a higher residual sum of squares indicates that the model poorly explains the data.

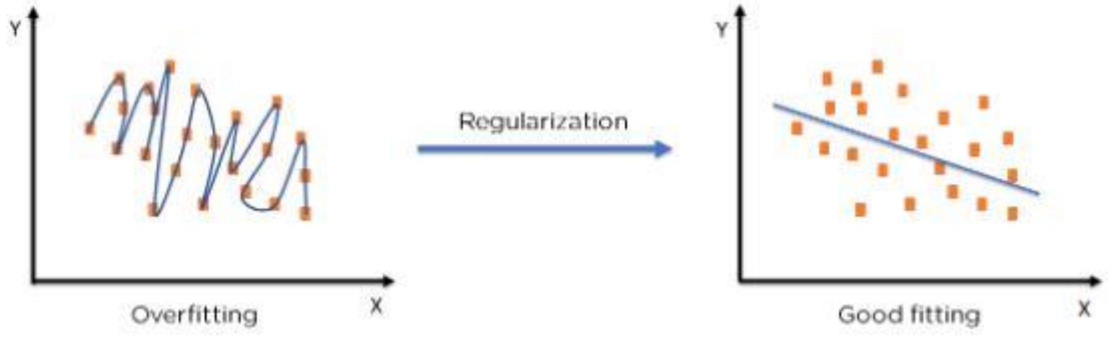
RSS = Σ(Yi – Y-Hat)^2.

- y_i – the observed value
- $\hat{y}_i$ – the value estimated by the regression line

TSS = ESS+RSS

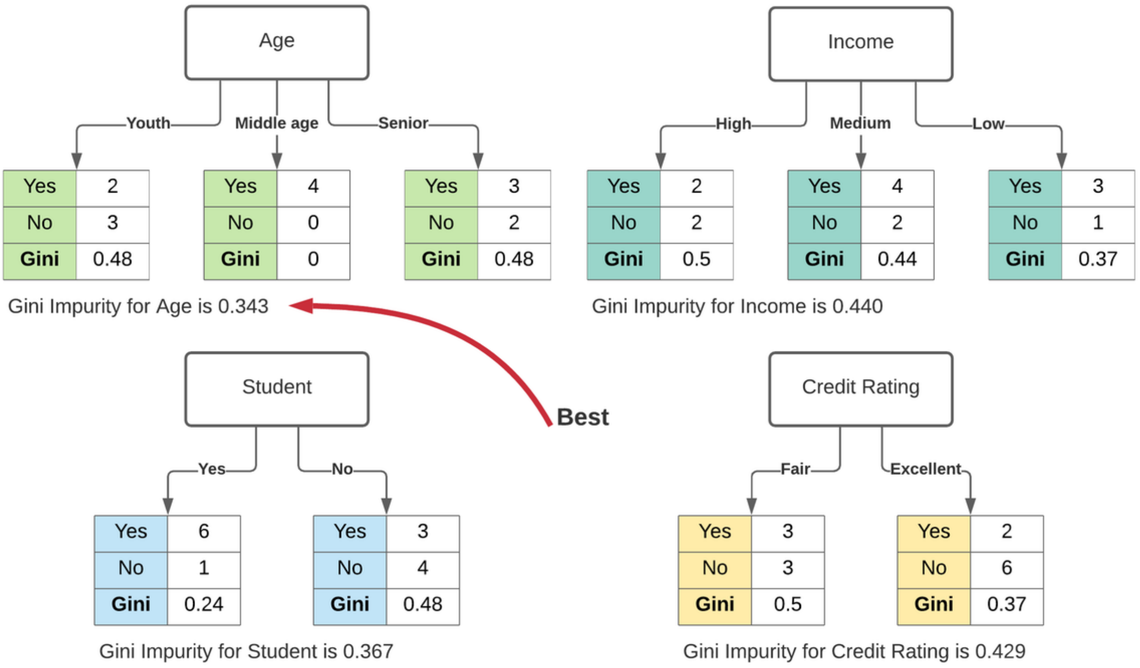
Que3. What is the need of regularization in machine learning?

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



Que 4. What is Gini–impurity index?

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.



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

Yes Unregularized Decision-trees prone to overfitting becauae Overfitting happens when any learning processing overly optimizes training set error at the cost test error. While it’s possible for training and testing to perform equality well in cross validation, it could be as the result of the data being very close in characteristics, which may not be a huge problem. In the case of decision tree’s they can learn a training set to a point of high granularity that makes them easily overfit. Allowing a decision tree to split to a granular degree, is the behavior of this model that makes it prone to learning every point extremely well — to the point of perfect classification — ie: overfitting.

Que 6. What is an ensemble technique in machine learning?

Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would. This has been the case in a number of machine learning competitions, where the winning solutions used ensemble methods. In the popular Netflix Competition, the winner used an ensemble method to implement a powerful collaborative filtering algorithm. Another example is KDD 2009 where the winner also used ensemble methods. You can also find winners who used these methods in Kaggle competitions.

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

Bagging: The simplest way of combining predictions that

belong to the same type.

Aim to decrease variance, not bias.

Each model receives equal weight.

Each model is built independently.

Different training data subsets are selected using row sampling with replacement and

random sampling methods from the entire training dataset.

Bagging tries to solve the over-fitting problem.

Boosting : A way of combining predictions that belong to the different types.
Aim to decrease bias, not variance.
Models are weighted according to their performance.
New Models are influenced by the performance of previous built models.
Every new subset contains the elements that were misclassified by previous models.
Boosting tries to reduce bias.

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

The RandomForestClassifier is trained using *bootstrap aggregation*, where each new tree is fit from a bootstrap sample of the training observations zi=(xi,yi). The *out-of-bag* (OOB) error is the average error for each zi calculated using predictions from the trees that do not contain zi in their respective bootstrap sample. This allows the RandomForestClassifier to be fit and validated whilst being trained.

Que 9. What is K-fold cross-validation?

k-fold cross-validation is one of the most popular strategies widely used by data scientists. It is a *data partitioning strategy* so that you can effectively use your dataset to build *a more generalized model*. The main intention of doing any kind of machine learning is to develop a more generalized model which can perform well on *unseen data*. One can build a perfect model on the training data with 100% accuracy or 0 error, but it may fail to generalize for unseen data. So, it is not a good model. It overfits the training data. Machine Learning is all about *generalization* meaning that model’s performance can only be measured with data points that have never been used during the training process. That is why we often split our data into a training set and a test set.

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

In machine learning, hyperparameter optimization[1] or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.

The same kind of machine learning model can require different constraints, weights or learning rates to generalize different data patterns. These measures are called hyperparameters, and have to be tuned so that the model can optimally solve the machine learning problem. Hyperparameter optimization finds a tuple of hyperparameters that yields an optimal model which minimizes a predefined loss function on given independent data.[2] The objective function takes a tuple of hyperparameters and returns the associated loss.[2] Cross-validation is often used to estimate this generalization performance.[

Que11. What issues can occur if we have a large learning rate in Gradient Descent?

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. If it is too small we will need too many iterations to converge to the best values. So using a good learning rate is crucial.

In simple language, we can define learning rate as how quickly our network abandons the concepts it has learned up until now for new ones.

Que 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, i.e. when the classes can be separated in the feature space by linear boundaries. That can be remedied however if we happen to have a better idea as to the shape of the decision boundary…

Logistic regression is known and used as a linear classifier. It is used to come up with a hyper*plane* 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.

Que 13. Differentiate between Adaboost and Gradient Boosting.

AdaBoost

AdaBoost or Adaptive Boosting 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.

In practice, this boosting technique is used with simple classification trees or stumps as base-learners, which resulted in improved performance compared to the classification by one tree or other single base-learner.

Gradient Boosting

Gradient Boost 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.

The technique yields a direct interpretation of boosting methods from the perspective of numerical optimisation in a function space and generalises them by allowing optimisation of an arbitrary loss function.

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

In statistics and 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–variance dilemma or bias–variance problem is the conflict in trying to simultaneously minimize these two sources of error that prevent supervised learning algorithms from generalizing beyond their training set:[1][2]

- 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 decomposition is a way of analyzing a learning algorithm's expected generalization error with respect to a particular problem as a sum of three terms, the bias, variance, and a quantity called the *irreducible error*, resulting from noise in the problem itsel