MACHINE LEARNING

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?

Ans-> R-squared indicates the proportion of variance explained by the model and is intuitive, but it factors in the number of variables, potentially leading to over fitting. RSS measures the model fit directly by summing squared residuals, with a lower RSS indicating a better fit.

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

Ans-> Sum of Squares

A statistical tool that is used to identify the dispersion of data

What is Sum of Squares?

Sum of squares (SS) is a statistical tool that is used to identify the dispersion of data as well as how well the data can fit the model in regression analysis. The sum of squares got its name because it is calculated by finding the sum of the squared differences.

The sum of squares is one of the most important outputs in regression analysis. The general rule is that a smaller sum of squares indicates a better model, as there is less variation in the data.

In finance, understanding the sum of squares is important because linear regression models are widely used in both theoretical and practical finance.

Types of Sum of Squares

In regression analysis, the three main types of sum of squares are the total sum of squares, regression sum of squares, and residual sum of squares.

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.

2. Regression sum of squares (also known as the sum of squares due to regression or explained sum of squares)

The regression 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.

3. Residual sum of squares (also known as the sum of squared errors of prediction)

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.

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

TSS = SSR + SSE

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

Ans -> Regularization is one of the most important concepts of machine learning. It is a technique to prevent the model from over fitting by adding extra information to it.

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 over fitted. 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.

It mainly regularizes or reduces the coefficient of features toward zero. In simple words, "In regularization technique, we reduce the magnitude of the features by keeping the same number of features."

4. What is Gini–impurity index?

The Gini Index is a proportion of the impurity or inequality of a circulation, regularly utilized as an impurity measure in decision tree algorithms. With regards to decision trees, the Gini Index is utilized to determine the best feature to split the data on at every node of the tree.

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

Ans -> Enhancing Model Stability and Performance;

1. Overfitting and Decision Trees

Decision trees, by their very nature, are prone to overfitting, especially when they are deep. Overfitting occurs when a model captures noise or fluctuations in the training data that do not represent the underlying data distribution. In the context of decision trees, overfitting can mean creating too many branches based on outliers or anomalies in the training data.

Why is this problematic?

A tree that is too complex might achieve a perfect accuracy score on the training data but perform poorly on new, unseen data. Such a tree has low bias but high variance, and its predictions can be unstable.

2. Regularization as a Solution

Regularization techniques add constraints to the learning algorithm, reducing its freedom and, hence, its capacity to overfit. For decision trees, regularization is achieved by controlling the tree's depth and complexity.

Key regularization techniques for decision trees include:

Maximum Depth (max_depth): Setting a limit on how deep the tree can grow. This is one of the most straightforward regularization methods. A shallow tree will likely underfit, and a very deep tree will likely overfit, so finding a balanced depth is crucial.

Minimum Samples Split (min_samples_split): This hyperparameter ensures that a node must have a minimum number of samples before it can be split. This can prevent tiny splits that capture noise.

Minimum Samples Leaf (min_samples_leaf): This ensures that a terminal node (leaf) has a minimum number of samples. This can be particularly useful to prevent the final class decisions from relying on a tiny subset of the data.

Maximum Features (max_features): By setting a limit on the number of features considered for splitting, you can add a form of feature regularization.

Pruning: Some algorithms, like cost complexity pruning, allow for the removal of parts of the tree that do not provide power in predicting the target values.

3. Striking the Right Balance

The key challenge in regularization is striking a balance between underfitting and overfitting. Under-regularized trees might be too deep and capture noise (overfit), while over-regularized trees might be too shallow and miss important patterns (underfit).

Regularization parameters should ideally be determined using cross-validation or a separate validation dataset, ensuring that the settings generalize well to new data.

6. What is an ensemble technique in machine learning?

Ans -> 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.

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

Ans -> Bagging tries to solve over-fitting problem while Boosting tries to reduce bias. 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.

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

Ans -> Out-of-bag (OOB) error, also called out-of-bag estimate, is a method of measuring the prediction error of random forests, boosted decision trees, and other machine learning models utilizing bootstrap aggregating (bagging). Bagging uses subsampling with replacement to create training samples for the model to learn from.

9. What is K-fold cross-validation?

Ans -> K-fold cross-validation is a technique for evaluating predictive models. The dataset is divided into k subsets or folds. The model is trained and evaluated k times, using a different fold as the validation set each time. Performance metrics from each fold are averaged to estimate the model's generalization performance.

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

Ans -> Hyperparameter tuning is the process of selecting the optimal values for a machine learning model’s hyperparameters. Hyperparameters are settings that control the learning process of the model, such as the learning rate, the number of neurons in a neural network, or the kernel size in a support vector machine. The goal of hyperparameter tuning is to find the values that lead to the best performance on a given task.

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

Ans -> Learning Rate Selection: The choice of learning rate can significantly impact the performance of gradient descent. If the learning rate is too high, the algorithm may overshoot the minimum, and if it is too low, the algorithm may take too long to converge.

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

Ans -> Logistic regression is simple and easy to implement, but it also has some drawbacks. One of them is that it assumes a linear relationship between the input features and the output. This means that it cannot capture the complexity and non-linearity of the data.

13. Differentiate between Adaboost and Gradient Boosting.

Ans -> Adaboost is computed with a specific loss function and becomes more rigid when comes to few iterations. But in gradient boosting, it assists in finding the proper solution to additional iteration modeling problem as it is built with some generic features.

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

Ans -> In statistics and machine learning, the bias–variance tradeoff describes the relationship between a model's complexity, the accuracy of its predictions, and how well it can make predictions on previously unseen data that were not used to train the model.

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

Ans -> choosing an appropriate kernel function for a given dataset is crucial to building an effective Support Vector Machine (SVM) model. The choice of kernel function determines the transformation of the input data into a higher-dimensional space where it can be separated more easily by a linear decision boundary. Here are some guidelines for selecting the appropriate kernel function:

1. Linear kernel: If the data can be well-separated by a linear decision boundary, a linear kernel should be used. A linear kernel is the simplest and most computationally efficient kernel function, and it works well for low-dimensional datasets with a large number of features.

2. Polynomial kernel: If the data has polynomial features or contains interaction effects between the features, a polynomial kernel should be used. A polynomial kernel maps the input data to a higher-dimensional space using polynomial functions of the original features.

3. Radial basis function (RBF) kernel: If the data cannot be well-separated by a linear or polynomial decision boundary, an RBF kernel should be used. An RBF kernel is a popular choice for SVMs because it can capture complex nonlinear relationships in the data. However, choosing the appropriate value of the gamma hyperparameter is critical to prevent overfitting.