1. In logistic regression, what is the logistic function (sigmoid function) and how is it used to

compute probabilities?

Ans:

Logistic regression is one of the most common machine learning algorithms used for binary classification. It predicts the probability of occurrence of a binary outcome using a logit function. It is a special case of linear regression as it predicts the probabilities of outcome using log function.

We use the activation function (sigmoid) to convert the outcome into categorical value. There are many examples where we can use logistic regression for example, it can be used for fraud detection, spam detection, cancer detection, etc.

It is a mathematical function having a characteristic that can take any real value and map it to between 0 to 1 shaped like the letter “S”. The sigmoid function also called a logistic function.

Y = 1 / 1+e -z

Sigmoid Function acts as an activation function in machine learning which is used to add non-linearity in a machine learning model, in simple words it decides which value to pass as output and what not to pass, there are mainly 7 types of Activation Functions which are used in machine learning and deep learning

2.When constructing a decision tree, what criterion is commonly used to split nodes, and

how is it calculated?

One of the most commonly used criteria for splitting nodes in a decision tree is the "Gini impurity" or "Gini index." The Gini impurity measures the degree or probability of a particular node being incorrectly classified when it is randomly labeled according to the distribution of labels in the node.

Mathematically, the Gini impurity for a node K classes can be calculated as follows:

Gini(t)=1-∑p(i∣t) ^2

p(i∣t) is the proportion of instances in node t that belong to class i.

To split a node in a decision tree based on the Gini impurity, the algorithm evaluates the impurity for each possible split on each feature and chooses the split that results in the lowest weighted sum of impurities for the resulting child nodes. This process continues recursively until a stopping criterion is met, such as reaching a maximum depth, having nodes with a minimum number of instances, or no further improvement in impurity reduction.

The split that minimizes the Gini impurity is selected as it leads to more homogeneous child nodes, improving the purity of the resulting subsets and thus making the classification more accurate.

3. Explain the concept of entropy and information gain in the context of decision tree

construction.

Ans: Entropy measures impurity in the data and information gain measures reduction in impurity in the data. The feature which has minimum impurity will be considered as the root node. Information gain is used to decide which feature to split on at each step in building the tree.

4. How does the random forest algorithm utilize bagging and feature randomization to

improve classification accuracy?

Ans:The random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

5. What distance metric is typically used in k-nearest neighbors (KNN) classification, and

how does it impact the algorithm's performance?

Ans:

k-Nearest neighbor classification

To classify an unknown instance represented by some feature vectors as a point in the feature space, the k-NN classifier calculates the distances between the point and points in the training data set. Usually, the Euclidean distance is used as the distance metric.

6. Describe the Naïve-Bayes assumption of feature independence and its implications for

classification.

Ans:

Assumption of Naive Bayes

Feature independence: The features of the data are conditionally independent of each other, given the class label. Continuous features are normally distributed: If a feature is continuous, then it is assumed to be normally distributed within each class.

7. In SVMs, what is the role of the kernel function, and what are some commonly used kernel

functions?

Ans:

“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.

8. Discuss the bias-variance tradeoff in the context of model complexity and overfitting.

Ans:

Bias Variance Tradeoff

If algorithms fit too complex (hypothesis with high degree equation) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as a Trade-off or Bias Variance Trade-off.

9. How does TensorFlow facilitate the creation and training of neural networks?

Ans:

1.Getting a batch of data to the model.

2.Asking the model to make a prediction.

3.Comparing that prediction with the "true" value.

4.Deciding how much to change each parameter so the model can make a better prediction in the future for that batch.

10. Explain the concept of cross-validation and its importance in evaluating model

performance.

Ans:

Cross-validation is a technique used to assess how well a predictive model will generalize to an independent dataset. It involves partitioning the original dataset into multiple subsets, training the model on a subset of the data (training set), and then evaluating it on the complementary subset (validation or testing set). This process is repeated multiple times, with different subsets used for training and validation in each iteration.

The most common type of cross-validation is k-fold cross-validation, where the data is divided into k equal-sized folds. The model is trained on

k−1 folds and evaluated on the remaining fold. This process is repeated k times, with each fold used as the validation set exactly once. The final performance metric is usually the average performance across all k iterations.

Cross-validation is essential for several reasons:

Better Estimate of Performance: By using multiple splits of the data, cross-validation provides a more reliable estimate of the model's performance compared to a single train-test split. It reduces the variance in the performance estimate, making it more robust and representative of the model's true generalization ability.

Reduces Overfitting: Cross-validation helps in identifying whether a model is overfitting or underfitting. If a model performs well on the training data but poorly on the validation data, it indicates overfitting, i.e., the model has learned noise in the training data rather than the underlying pattern.

Hyperparameter Tuning: Cross-validation is often used in hyperparameter tuning, where the goal is to find the optimal set of hyperparameters that results in the best model performance. By evaluating the model with different hyperparameter configurations across multiple cross-validation folds, one can choose the set of hyperparameters that generalizes well to unseen data.

Model Selection: Cross-validation can also be used for comparing different models. By applying the same cross-validation procedure to multiple models, one can determine which model performs better on average across different data splits.

11. What techniques can be employed to handle overfitting in machine learning models?

Ans:  
Cross-Validation: As discussed earlier, cross-validation helps in assessing the model's performance on unseen data and can provide insights into whether the model is overfitting.

Train with More Data: Increasing the size of the training dataset can help the model to generalize better by exposing it to more diverse examples of the underlying pattern.

Feature Selection/Engineering: Selecting relevant features and removing irrelevant or redundant ones can simplify the model and reduce overfitting. Feature engineering techniques like dimensionality reduction (e.g., PCA) can also help in capturing the most important aspects of the data.

Dropout: Dropout is a regularization technique commonly used in neural networks. During training, randomly selected neurons are ignored or "dropped out" with a certain probability. This helps in preventing co-adaptation of neurons and encourages the network to learn more robust features.

Data Augmentation: Increasing the diversity of the training data by applying transformations like rotation, translation, scaling, or adding noise can help in regularizing the model and improving its generalization performance.

12. What is the purpose of regularization in machine learning, and how does it work?

Ans:

Regularization is a technique used in machine learning to prevent overfitting and improve the generalization performance of models. Overfitting occurs when a model learns to fit the training data too closely, capturing noise or random fluctuations in the data rather than the underlying pattern. Regularization methods add a penalty term to the model's objective function, discouraging overly complex models that are prone to overfitting.

Overfitting: Regularization helps in controlling the complexity of the model, making it less likely to fit the noise in the training data and more likely to generalize well to unseen data.

Improving Generalization: By reducing overfitting, regularization techniques help in improving the model's ability to generalize to new, unseen data, resulting in better performance on test or validation datasets.

13. Describe the role of hyper-parameters in machine learning models and how they are tuned

for optimal performance.

Ans:

Hyperparameters are parameters that are not learned from the data during the training process but are set before training begins. They control aspects of the learning process, such as the complexity of the model, the learning rate, the regularization strength, etc. The role of hyperparameters is crucial as they directly impact the performance of the model, including its ability to generalize to new, unseen data.

Grid Search: Grid search is a brute-force technique where a predefined grid of hyperparameter values is specified, and the model is trained and evaluated for each combination of hyperparameters. The combination that results in the best performance on a validation set is selected as the optimal set of hyperparameters. While simple and straightforward, grid search can be computationally expensive, especially for models with many hyperparameters or large search spaces.

Random Search: Random search is an alternative to grid search where hyperparameters are sampled randomly from predefined distributions. Instead of exhaustively searching over all combinations, random search explores the search space more efficiently by sampling hyperparameters randomly. While it may not guarantee finding the optimal hyperparameters, random search is often more computationally efficient than grid search and can be effective, especially in high-dimensional search spaces.

Bayesian Optimization: Bayesian optimization is an iterative optimization technique that builds a probabilistic model of the objective function (model performance) and uses it to decide which hyperparameters to explore next. By leveraging information from previous evaluations, Bayesian optimization guides the search towards promising regions of the hyperparameter space, making it more efficient than random search or grid search. It is particularly useful when the evaluation of the objective function (model training and validation) is computationally expensive.

14. What are precision and recall, and how do they differ from accuracy in classification

evaluation?

Ans:

Precision: Precision measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive (true positives + false positives).

Precision answers the question: "Of all the instances predicted as positive, how many are actually positive?"

Precision is particularly useful when the cost of false positives is high, and we want to minimize the number of false positives.

Recall: Recall, also known as sensitivity or true positive rate (TPR), measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances (true positives + false negatives).

Recall answers the question: "Of all the actual positive instances, how many were predicted as positive?"

Recall is particularly useful when the cost of false negatives is high, and we want to minimize the number of false negatives.

Accuracy: Accuracy measures the overall correctness of the model and represents the proportion of correctly classified instances (both true positives and true negatives) out of all instances.

Accuracy answers the question: "Of all instances, how many were classified correctly?"

Accuracy is not suitable for imbalanced datasets because it can be high even when the model performs poorly on the minority class.

15. Explain the ROC curve and how it is used to visualize the performance of binary classifiers.

Ans:

True Positive Rate (TPR):

TPR measures the proportion of positive instances (actual positives) that are correctly classified as positive by the classifier.

False Positive Rate (FPR):

FPR measures the proportion of negative instances (actual negatives) that are incorrectly classified as positive by the classifier.

ROC Curve:

The ROC curve is created by plotting TPR (sensitivity) on the y-axis against FPR (1-specificity) on the x-axis for different threshold values.

Each point on the ROC curve represents a sensitivity-specificity pair corresponding to a particular threshold setting of the classifier.

The diagonal line (y = x) represents the ROC curve of a random classifier, which has no discriminative ability. A classifier above this line is considered better than random.

Area Under the ROC Curve (AUC-ROC):

The AUC-ROC represents the area under the ROC curve and provides a single scalar value summarizing the classifier's performance across all possible threshold settings.

AUC-ROC ranges from 0 to 1, where 0 indicates a poor classifier (performs worse than random guessing), and 1 indicates a perfect classifier (perfect discrimination between positive and negative classes).