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

The logistic function, or sigmoid function, maps input values to probabilities between 0 and 1. It's computed as $1/(1+e^{-z})$, where z is a linear combination of input features. In logistic regression, it's used to model the probability of a binary outcome.

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

The commonly used criterion to split nodes in a decision tree is Gini impurity or information gain. Gini impurity measures the probability of incorrectly classifying a randomly chosen element if it was randomly labeled, while information gain measures the reduction in entropy or uncertainty after a split.

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

Entropy in decision trees measures the impurity or uncertainty of a dataset. Information gain quantifies the effectiveness of a feature in reducing entropy after a split. The goal is to maximize information gain or minimize entropy at each split to construct an optimal decision tree.

4. How does the random forest algorithm utilize bagging and feature randomization to improve classification accuracy?

Random forest employs bagging (bootstrap aggregation) by training multiple decision trees on random subsets of the training data. Feature randomization further enhances diversity among trees by considering random subsets of features for each split. This ensemble approach reduces overfitting and improves classification accuracy.

5. What distance metric is typically used in k-nearest neighbours (KNN) classification, and how does it impact the algorithm's performance?

The Euclidean distance metric is typically used in k-nearest neighbors (KNN) classification. It measures the straight-line distance between two points in the feature space. The choice of distance metric impacts KNN's performance, with different metrics being suitable for different types of data.

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

Naïve Bayes assumes that features are conditionally independent given the class label. This simplifies the calculation of probabilities and makes the algorithm computationally efficient. However, this assumption may not hold true in practice, leading to potential inaccuracies in classification.

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

The kernel function in SVMs computes the inner product of feature vectors in a high-dimensional space. It maps input data into a higher-dimensional space where linear separation may be possible. Commonly used kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels.

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

The bias-variance trade off refers to the balance between a model's bias (error from overly simplistic assumptions) and variance (sensitivity to fluctuations in the training data). Increasing model complexity reduces bias but increases variance, leading to overfitting.

Finding the optimal trade off is crucial to prevent overfitting and achieve good generalization performance.

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

TensorFlow facilitates the creation and training of neural networks by providing a comprehensive library for building computational graphs, automatic differentiation for optimizing model parameters, GPU acceleration for faster training, and high-level APIs like Keras for simplified model development.

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

Cross-validation involves partitioning the dataset into multiple subsets, training the model on a subset, and evaluating it on the remaining data. This process is repeated multiple times, and the average performance is used to assess the model's generalization ability. Cross-validation helps prevent overfitting and provides a more reliable estimate of model performance.

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

Techniques to handle overfitting include increasing the size of the training dataset, reducing model complexity, using regularization techniques such as L1/L2 regularization, applying early stopping during training, incorporating dropout, and using ensemble methods like random forests or gradient boosting.

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

Regularization in machine learning is used to prevent overfitting by adding a penalty term to the model's loss function. This penalty discourages overly complex models by penalizing large parameter values. Regularization techniques like L1 (Lasso) and L2 (Ridge) regularization help control model complexity and improve generalization performance.

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

Hyperparameters are settings that control the behavior of a machine learning algorithm. They are not learned from the data but are set before training. Hyperparameters are tuned using techniques like grid search, random search, or Bayesian optimization to find the combination that maximizes the model's performance on a validation set.

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

Precision measures the proportion of true positive predictions among all positive predictions made by the classifier, while recall measures the proportion of true positive predictions among all actual positive instances in the dataset. Accuracy, on the other hand, measures the overall correctness of the classifier's predictions.

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

The ROC (Receiver Operating Characteristic) curve is a graphical representation of the true positive rate (TPR) against the false positive rate (FPR) for different classification thresholds. It visualizes the tradeoff between sensitivity and specificity and helps assess the performance of binary classifiers across various threshold values.