Solutions to Machine Learning Questions

1. Bias-Variance Tradeoff: The bias-variance tradeoff is a fundamental problem in machine learning. Bias measures the error due to incorrect assumptions in the learning algorithm, while variance measures the error due to sensitivity to small fluctuations in the training set. A model with high bias tends to oversimplify the problem, leading to underfitting. In contrast, a model with high variance captures noise in the data, leading to overfitting. The goal is to find the right balance to minimize total error.

2. Regularization Techniques: Common regularization techniques include L1 regularization (Lasso), which adds the absolute value of coefficients as a penalty term to the loss function, and L2 regularization (Ridge), which adds the squared value of coefficients as a penalty. Regularization helps prevent overfitting by penalizing large coefficients.

3. Cross-Validation: Cross-validation is a technique to assess the generalization performance of a model. It involves partitioning the data into training and validation sets multiple times to ensure the model performs well on unseen data. Techniques include k-fold cross-validation, where the dataset is split into 'k' groups, and each group is used as a validation set while the rest serve as training data.

4. Bagging vs. Boosting: Bagging (Bootstrap Aggregating) reduces variance by averaging predictions from multiple models trained on different data samples. Boosting, on the other hand, focuses on correcting the mistakes made by previous models by giving more weight to misclassified instances. Both techniques aim to improve the accuracy of ensemble models.

5. Linear Regression Assumptions: Key assumptions include linearity of the relationship between dependent and independent variables, homoscedasticity (constant variance of errors), independence of errors, and normality of error terms.

6. Decision Tree Splitting: A decision tree uses metrics like Gini impurity or information gain to determine the best split at each node. It evaluates all possible splits and selects the one that maximizes the separation of the classes.

7. Curse of Dimensionality: In machine learning, the curse of dimensionality refers to the exponential increase in data needed to maintain model accuracy as the number of features increases. High dimensionality can lead to overfitting and make distance-based algorithms like k-NN less effective.

8. Random Forest vs. Decision Tree: A single decision tree is prone to overfitting, while a Random Forest, which is an ensemble of multiple decision trees, reduces overfitting by averaging the predictions from individual trees trained on different subsets of the data.

9. Parametric vs. Non-Parametric Models: Parametric models make assumptions about the data distribution and have a fixed number of parameters, such as linear regression. Non-parametric models, like k-NN, do not assume a specific form for the function and can adapt to the data's complexity.

10. Feature Importance in Tree-Based Models: In models like Random Forest or Gradient Boosting, feature importance is measured by calculating the average reduction in impurity (e.g., Gini impurity or entropy) brought by the feature across all the trees in the ensemble.

11. Gradient Descent Variants: Gradient descent is an optimization algorithm used to minimize the loss function. Variants include:

- Stochastic Gradient Descent (SGD): Updates parameters for each training example, making it faster but noisier.
- Mini-Batch Gradient Descent: Uses a small batch of training examples, balancing between standard gradient descent and SGD.
- Batch Gradient Descent: Uses the entire dataset for each update, which is computationally expensive but stable.

12. How k-NN Works and Limitations: k-Nearest Neighbors (k-NN) classifies data based on the majority vote of the 'k' closest neighbors. Limitations include high computational cost for large datasets and poor performance with high-dimensional data due to the curse of dimensionality.