KNN-Related Questions

1. **What is K-Nearest Neighbors (KNN) and how does it work?**
   - KNN is a simple, non-parametric machine learning algorithm used for classification and regression. It works by identifying the `K` nearest neighbors of a data point in feature space using a distance metric (e.g., Euclidean distance). The prediction is made based on the majority class (classification) or average of neighbors (regression).
2. **What is the difference between KNN Classification and KNN Regression?**
   - **Classification:** Predicts a discrete class label by majority voting among neighbors.
   - **Regression:** Predicts a continuous value by averaging the values of neighbors.
3. **What is the role of the distance metric in KNN?**
   - The distance metric determines how neighbors are identified. Common metrics include Euclidean, Manhattan, and Minkowski distances. The choice affects accuracy and is dependent on the dataset.
4. **What is the Curse of Dimensionality in KNN?**
   - In high-dimensional spaces, data points become sparse, and the distance metric loses its discriminative power, making KNN less effective.
5. **How can we choose the best value of K in KNN?**
   - Use techniques like cross-validation. Smaller `K` values may result in overfitting, while larger values smooth predictions but can lead to underfitting.
6. **What are KD Tree and Ball Tree in KNN?**
   - KD Tree and Ball Tree are data structures used to speed up nearest-neighbor searches:
     - **KD Tree:** Organizes points in `k`-dimensional space using hyperplanes.
     - **Ball Tree:** Partitions space using hyperspheres.
7. **When should you use KD Tree vs. Ball Tree?**
   - **KD Tree:** Efficient for low-dimensional data.
   - **Ball Tree:** Performs better with high-dimensional data.
8. **What are the disadvantages of KNN?**
   - Computational cost during prediction.
   - Sensitivity to irrelevant features.
   - Poor performance in high-dimensional spaces.
9. **How does feature scaling affect KNN?**
   - KNN relies on distance measurements, so scaling features (e.g., using Min-Max or Standardization) ensures features contribute equally to distance calculations.

PCA-Related Questions

10. **What is PCA (Principal Component Analysis)?**
    - PCA is a dimensionality reduction technique that transforms data into a set of orthogonal components, capturing maximum variance.
11. **How does PCA work?**

- PCA computes the covariance matrix of data, finds eigenvalues and eigenvectors, and projects data onto the principal components.

12. **What is the geometric intuition behind PCA?**
    - PCA finds new axes (principal components) that maximize variance, effectively rotating the data to reduce redundancy.
13. **What is the difference between Feature Selection and Feature Extraction?**
    - **Feature Selection:** Selects a subset of existing features.
    - **Feature Extraction:** Creates new features by transforming data (e.g., PCA).
14. **What are Eigenvalues and Eigenvectors in PCA?**
    - Eigenvalues represent the variance captured by principal components.
    - Eigenvectors define the direction of principal components.
15. **How do you decide the number of components to keep in PCA?**
    - Use a Scree plot or set a threshold for cumulative explained variance (e.g., 95%).
16. **Can PCA be used for classification?**
    - Indirectly. PCA reduces dimensionality, which improves classification performance in algorithms sensitive to high-dimensional data.
17. **What are the limitations of PCA?**
    - Assumes linear relationships, sensitive to scaling, and can lose interpretability of original features.
18. **How do KNN and PCA complement each other?**
    - PCA reduces dimensionality, mitigating the curse of dimensionality in KNN, thereby improving performance.
19. **How does KNN handle missing values in a dataset?**
    - Typically, KNN requires imputation of missing values before training, such as using mean, median, or neighbors’ values.
20. **What are the key differences between PCA and Linear Discriminant Analysis (LDA)?**
    - **Objective:** PCA focuses on maximizing variance, while LDA maximizes class separability.
    - **Applicability:** PCA works for unsupervised tasks; LDA is used for supervised classification.