Task 6 - KNN Classification Interview Q&A

1. How does the KNN algorithm work?

KNN predicts the class of a data point by looking at the 'K' nearest neighbors in the training data. It assigns the class that is most common among those neighbors.

2. How do you choose the right K?

You try different K values (e.g., from 1 to 20) and pick the one that gives the best performance on validation data. Small K values may overfit; large K values may underfit.

3. Why is normalization important in KNN?

Because KNN is based on distance, and features on larger scales (like salary) can dominate smaller ones (like age). Normalization ensures all features contribute equally.

4. What is the time complexity of KNN?

Training time is O(1), but prediction time is O(n × d), where n = number of training samples, and d = number of features. It's slow at prediction time.

5. What are pros and cons of KNN?

Pros: Simple, no training required, supports multi-class.
Cons: Slow prediction, sensitive to noise, needs feature scaling, memory-intensive.

6. Is KNN sensitive to noise?

Yes. A noisy or mislabeled neighbor can lead to wrong predictions, especially with small K. Using a larger K or cleaning the data helps.

7. How does KNN handle multi-class problems?

KNN naturally supports multi-class classification. It looks at the K nearest neighbors and picks the most frequent class among them.

8. What's the role of distance metrics in KNN?

KNN uses distance to find neighbors. Common metrics include Euclidean, Manhattan, and Minkowski. The choice affects performance depending on the data type and scale.