PW SKILLS
KNN ASSISGNMENT 2

1. Difference between Euclidean and Manhattan Distance:
   - Euclidean distance measures the straight-line distance between two points in a Euclidean space, considering the square root of the sum of squared differences along each dimension.
   - Manhattan distance, also known as the L1 norm, calculates the distance by summing the absolute differences along each dimension, resembling the distance traveled along city block paths.
   - The main difference lies in how they calculate distance: Euclidean considers the direct line between points, while Manhattan measures distance along axes.
   - This difference can affect KNN's performance because Euclidean distance gives equal weight to all dimensions, making it sensitive to features with larger magnitudes. In contrast, Manhattan distance is more robust to differences in feature scales, which can affect how KNN perceives data proximity.

2. Choosing the Optimal Value of K:
   - The optimal value of K in KNN significantly impacts its performance. A small K can lead to overfitting, while a large K may cause oversmoothing of decision boundaries.
   - Techniques for determining the optimal K include cross-validation, where different K values are tested and evaluated on validation sets, or grid search, where a range of K values is systematically tested to find the one with the best performance.
   - Domain knowledge and understanding the underlying data distribution can also help in selecting an appropriate K value.

3. Impact of Distance Metric Choice:
   - The choice of distance metric affects how KNN defines proximity between data points, thereby influencing classification or regression outcomes.
   - Euclidean distance is sensitive to feature scales and assumes data distribution is spherical, making it suitable for continuous, normally distributed data.

- Manhattan distance is robust to scale differences and works well with high-dimensional or sparse data, as it doesn't assume spherical distribution.
- Situations where one might choose one over the other include when dealing with highly dimensional data or when feature scales differ significantly.

4. Common Hyperparameters and Tuning:
- Common hyperparameters in KNN include K (number of neighbors), distance metric, and optional weighting of neighbors.
- K affects the bias-variance tradeoff; smaller K values increase model complexity, potentially leading to overfitting, while larger K values may introduce oversmoothing.
- Distance metric choice influences how KNN measures proximity and thus impacts classification or regression results.
- Hyperparameters can be tuned using techniques like grid search or randomized search, where different combinations are tried and evaluated based on cross-validation performance.

5. Effect of Training Set Size:
- The size of the training set directly affects KNN's performance. A small training set may lead to overfitting, as there may not be enough data to capture the underlying patterns adequately.
- Conversely, a large training set can improve generalization but may increase computational costs.
- Techniques to optimize training set size include using techniques like cross-validation to assess model performance with different training set sizes and selecting the largest feasible training set without compromising computational resources.

6. Drawbacks of KNN and Overcoming Them:
- Drawbacks of KNN include computational inefficiency with large datasets, sensitivity to irrelevant features, and the need for proper scaling and feature selection.
- To address computational inefficiency, techniques like KD-trees or ball trees can be used to speed up nearest neighbor searches.
- Sensitivity to irrelevant features can be mitigated by feature selection techniques or using distance metrics that down-weight less informative features.

- Proper feature scaling ensures that all features contribute equally to distance calculations, reducing the impact of feature magnitudes on the results.

- Dimensionality reduction methods like PCA or feature engineering can help reduce the curse of dimensionality and improve KNN's performance.