SUMMARY REPORT

This report summarizes the results of applying the KNN algorithm to two datasets: the Iris dataset (a well-known dataset in machine learning) and a simulated dataset generated using the make_blobs function. The objective was to assess the accuracy of the KNN classifier on both datasets.

The Iris dataset contains 150 samples from three classes of the Iris plant: Setosa, Versicolor, and Virginica. Each sample includes four features: sepal length, sepal width, petal length, and petal width. The dataset was split into training (80%) and testing (20%) sets using train_test_split with a random seed of 12 to ensure reproducibility. A KNN classifier was first initialized with the default parameters (n_neighbors=5, weights=uniform, metric=minkowski, p=2 for Euclidean distance). The classifier was trained using the training data and evaluated on both the training and testing sets.

The classifier achieved an accuracy of 0.975 (97.5%) on the training set and 0.967 (96.7%) on the testing set using the default settings. Additionally, a second KNN model was created with specific parameters such as algorithm=auto, leaf_size=30, metric=minkowski, p=2, n_neighbors=5, and weights=uniform. Although, the accuracy of this optimized model was not explicitly evaluated in the code provided. Therefore, the performance results discussed in this report primarily refer to the default KNN model.

An artificial dataset was generated using the make_blobs function, which creates clusters of data points. The simulated dataset consists of three distinct classes centered around points [2, 4], [6, 6], and [1, 9`. The dataset includes 150 samples distributed across the three classes. The dataset was split into training (80%) and testing (20%) sets using train_test_split with a random seed of 23. A KNN classifier was initialized with default parameters and trained on the training set. Predictions were made on the test set, and the accuracy score was calculated. The classifier achieved a perfect accuracy of 1.0 (100%) on the test set, indicating that the KNN algorithm was highly effective in classifying the simulated dataset. A scatter plot of the dataset was generated, showing the distribution of the data points and their respective classes. The three classes were well-separated, contributing to the high accuracy of the classifier.

The KNN algorithm demonstrated high accuracy on both the Iris dataset and the simulated dataset. For the Iris dataset, the default KNN model performed very well, achieving around 96.7% accuracy on the test set. While an additional set of parameters was used to create an optimized KNN model, its performance was not explicitly compared in this analysis. For the simulated dataset, the KNN model achieved perfect accuracy, which is expected given the clear separation between the three classes in the data.

The choice of parameters in the KNN algorithm (such as n_neighbors, weights, and metric) significantly impact the model performance. This analysis demonstrates the effectiveness of the KNN algorithm in classifying data with clear boundaries between classes, as well as its robustness in dealing with real-world datasets like the Iris dataset.