Title: Implementation and Visualization of K-Nearest Neighbors (KNN) Algorithm for Classification.

**Introduction**

The **K-Nearest Neighbors (KNN)** algorithm is a simple, non-parametric, and lazy learning method used for classification and regression tasks. It works by finding the majority class or average value of the nearest data points in the feature space. In classification, the class label of a new sample is determined by the majority class of its nearest neighbors, based on a distance metric, such as Euclidean distance.

In this implementation, the KNN algorithm is applied to a 2D dataset, where each point is assigned a class label, and the algorithm predicts the class label of a new, unseen data point based on its proximity to the training data.

**Objective**

The objective of this report is to demonstrate the application of the **K-Nearest Neighbors (KNN)** algorithm using a custom Python implementation. The goal is to:

* Split the data into training and testing sets.
* Train a KNN classifier on the training data.
* Make predictions on new data points.
* Visualize the data points, the training data, and the predictions.
* Evaluate the classifier's performance using visual inspection.

**Algorithm**

The KNN algorithm follows these basic steps:

1. **Distance Calculation**: For a given test point, the algorithm computes the distance to all points in the training set. Here, Euclidean distance is used:

Euclidean Distance=∑i=1n(xi−yi)2\text{Euclidean Distance} = \sqrt{\sum\_{i=1}^{n} (x\_i - y\_i)^2}Euclidean Distance=i=1∑n​(xi​−yi​)2​

1. **Finding Neighbors**: Sort the distances in ascending order and select the kkk nearest neighbors to the test point.
2. **Voting Mechanism**: For classification tasks, the class label of the test point is determined by the majority class of the kkk nearest neighbors.
3. **Prediction**: The predicted class label is returned as the most common class among the kkk nearest neighbors.
4. **Conclusion**
5. In this report, we applied the K-Nearest Neighbors (KNN) algorithm to classify a new data point based on its proximity to training data points. By splitting the dataset into training and test sets, we were able to evaluate the KNN model on unseen data.
6. The model successfully predicted the class label for the new point based on the majority class of the 3 nearest neighbors. The visualization helped us understand how the KNN algorithm works by showing the spatial relationship between training data and predictions. The simplicity and effectiveness of KNN make it an ideal choice for basic classification problems, although it can be computationally expensive for large datasets.
7. Future work could involve optimizing the choice of kkk (number of neighbors) and experimenting with different distance metrics to improve model performance.