Report Description

Assignment-1

Overview:

Analyzing the way the K-Nearest Neighbors (KNN) method worked with a recently created simulated dataset was the main objective of the study. If the dataset was constructed with three distinct categories, data points may be grouped based on their feature properties**.**

Methodology:

1. Data Generation:

Using the scikit-learn make_blobs method from the scikit_learn packages in a library is one method of creating data where the class centers. [2,4], [6, 4], and [1, 6]], and the dataset comprised 150 samples and three classes.

• In the second dimension, every information point's two elements (or dimensions) represented the X and Y coordinates.

• For better data creation dependability, a random state of 1 was created

2. Train-Test Split: The train test split() function was used to divide the dataset into training and testing sets. 20% of the data was reserved for testing, while the rest, or 80%, was used to train the model. The data was systematically separated using a random state of 13.

3. KNN Classifier: KNN Classifier: Taking five nearest neighbors (neighbors=5) as the default, a k-nearest-neighbors classifier (the neighbor classifier()) was developed. Following the training of models with data from both training and testing samples were used for predictions.

4. **Measures of Evaluation**: The primary standard by which the model's performance was assessed was the score for accuracy. It was established for accuracy using the functionality score() for each of the training and testing sets.

This offered information regarding the model's ability to successfully generalize and classify new data.

5. Visualization of data:

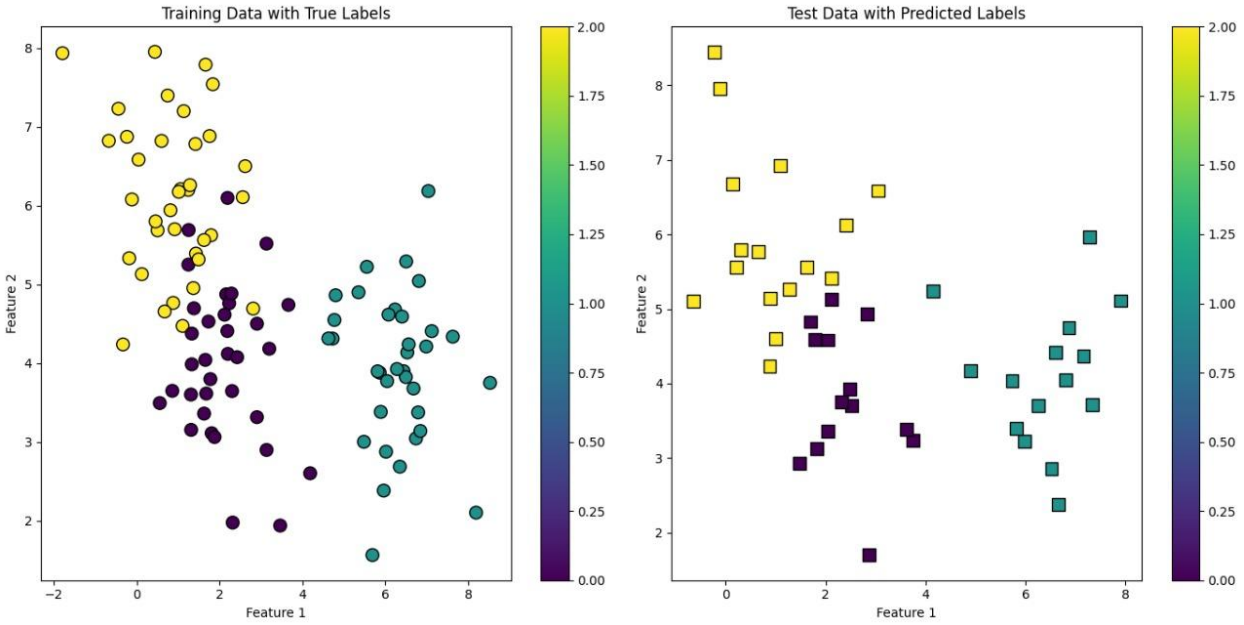
Two scatter plots were made in order to demonstrate the classification results,The training data with the actual class names was displayed in the first plot. The test results with the planned class designations have been shown in the second plot. To distinguish between all three clusters, Each class's data points were color-coded. The graphic makes it easy to understand the class distribution and the classification performance of the model.

Results:

• Accuracy of Instruction Set: Using the training data, the KNN classifier obtained an accuracy of 95%.

• Testing Set Accuracy: The model adaptation to new, unseen data was demonstrated by the testing data's accuracy of 96%

Plots :



True Label Training Data (Left Plot):

• Plot Description: The data used for training points are arranged in this scatter plot based on their actual class designations. A sample from the training set can be seen by each point, with the the point's color that meets its true class label.

Colors: This color map (viridis) graphically distinguishes the three classes. The arrangement of the examples is easy to see because each class has a unique color.

• Axes Two aspects of the data are represented by the x-axis (Feature 1) and y-axis (Feature 2). dataset.

• Edge Color: In order to make the points suggest out more against the background, each have a black edge (edgecolor='k').

• Marker: The training data uses circular markers (marker='o').

• Objective By displaying the geographic distribution of the training set's actual classes, this plot supports in recognizing the data's underlying structure.

Test Data (Right Plot) with Predicted Labels:

• Plot Description: The test data points are shown in this scatter plot, with each of them colored in line with the KNN classifier's anticipated class label.

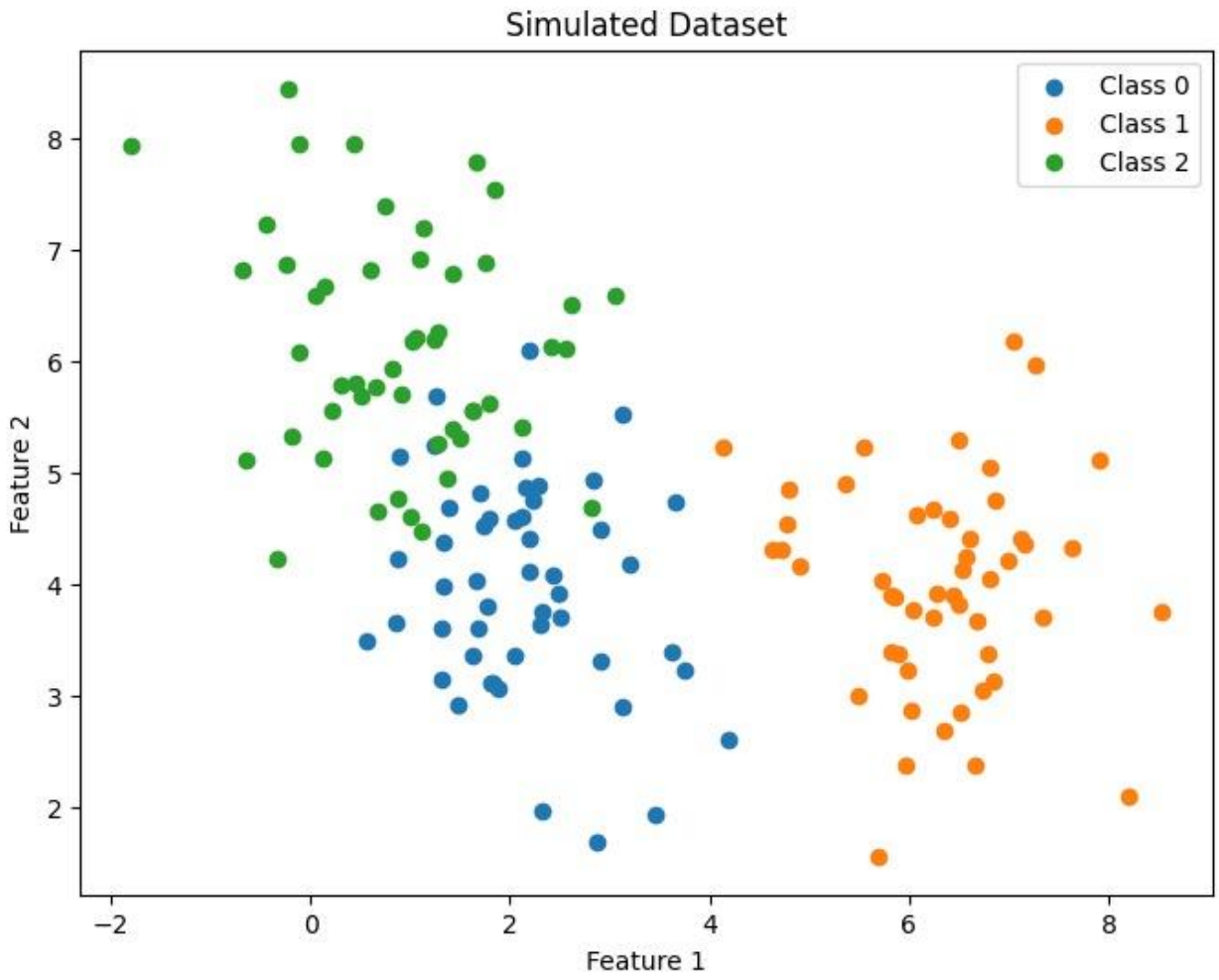
• Colors: Similar to the exercise plot, the colors represent the class labels; however, in this instance, the classes are predicted using KNN

• Axes: once more, the dataset's two features (Feature 1 and Feature 2) are denoted the KNN model by the x and y axes.

• Edge Color: To draw attention to each point, the points have a black border, much like in the training plot.

• Marker: To distinguish across test and training data, square markers (marker='s') are utilized.

• Goal: This plot displays the planned classifications of the test and shows how effectively the test set was handled by the KNN classifier.



Plot Description: The entire simulated dataset used in the K-Nearest Neighbors (KNN) study is shown in this scatter plot. A data sample is portrayed by each plot point, and it is color-coded based on its class identification.

Colors and Classes:

This means that there are three different classes in the dataset: Class 0, Class 1, and Class 2. The data points are easily grouped and divided because each class is represented by a distinctive hue.

• Each class's data points are presented in a distinct color thanks to the for loop, that selects the points that match to class I using the condition labels == i.

• Each class now has a legend item added by the label=f'Class {i}' section, thus making it simpler for figuring out which color goes with which class.

Axes: • The two distinct characteristics of the dataset are denoted by the x-axis (Feature 1) and y-axis (Feature 2), where the values for these two features define the position of each point**.**

Legend : For demonstrating which color goes with which class, the plot contains a legend. This aids in the visual differentiation of the various classes.

The **goal** of this plot is to give a general picture of how the data points are distributed throughout the three classes. Understanding the degree of class separation or overlap is important for identifying issues. The distribution and clustering of the points offer details about the intricacy of the classification problem and the potential performance of a model such as KNN on the dataset.

In **summary**, the study effectively reproduced the KNN algorithm's efficiency employing a previously created simulated dataset. The KNN algorithm's extremely high precision scores on both the training and testing sets show how well it classifies data items based on their qualities**.**