Summary Report

Assignment-1

Introduction:

The aim of this study was to assess the performance of the K-Nearest Neighbors (KNN) algorithm on a newly generated simulated dataset. The dataset was designed with three distinct classes, allowing for the classification of data points based on their feature attributes.

Methodology:

1. **Data Generation**:
   - An artificial dataset was generated using the make_blobs function from the scikit-learn library. The dataset consisted of 150 samples and three distinct classes, with the class centers defined as [[2, 4], [6, 6], [1, 9]].
   - Each data point had two features (or dimensions) corresponding to the X and Y coordinates in a 2D plane.
   - A random state of 1 was set to ensure reproducibility in data generation.
2. **Train-Test Split**:
   - The dataset was split into training and testing sets using the train_test_split() function.
   - 80% of the data was allocated for training the model, while 20% was set aside for testing. A random state of 12 was used for consistent splitting of the data.
3. **KNN Classifier**:
   - A **K-Nearest Neighbors** classifier (KNeighborsClassifier()) was implemented with a default of 5 nearest neighbors (n_neighbors=5).
   - The model was trained using the training data, and predictions were made on both the training and testing datasets.
4. **Evaluation Metrics**:
   - The **accuracy score** was used as the primary metric for evaluating model performance.
   - The accuracy score was computed for both the training and testing sets using the accuracy_score() function. This provided insight into the model's ability to generalize and correctly classify new data.

5. **Data Visualization**:

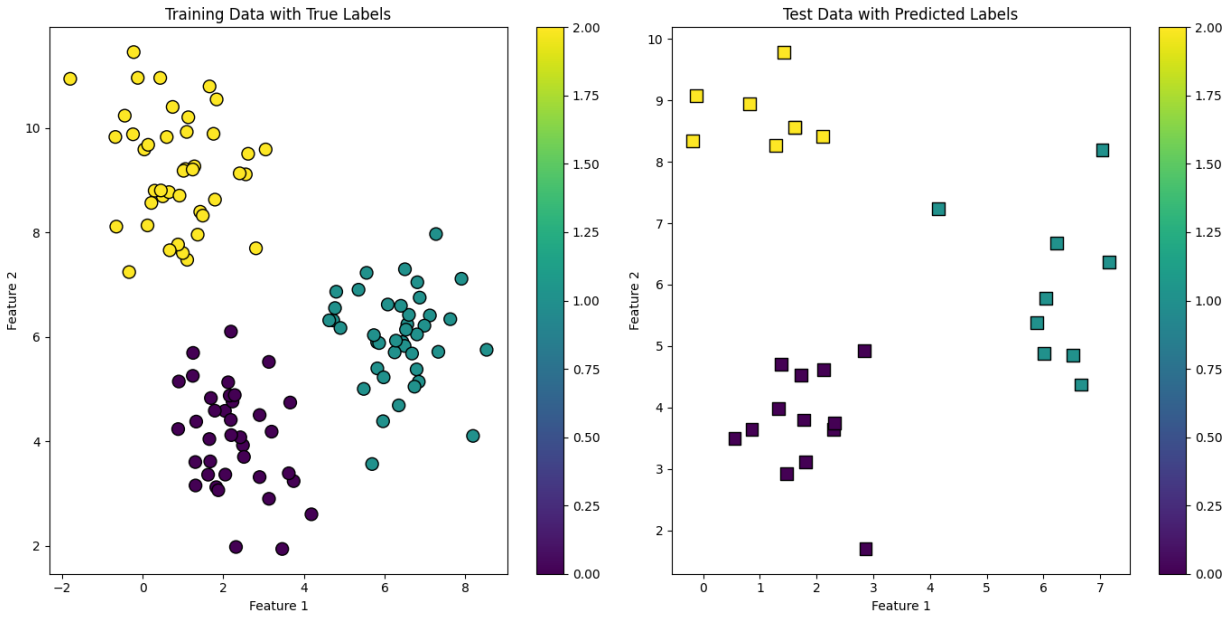
- To visualize the classification results, two scatter plots were created:
    - The first plot visualized the **training data** with the true class labels.
    - The second plot visualized the **test data** with the predicted class labels.
- Data points were color-coded by class to differentiate between the three clusters. The visualization helped in understanding the distribution of the classes and the model's performance in classifying them.

Results:

- Training Set Accuracy: The KNN classifier achieved an accuracy of 1.0 (100%) on the training data.
- Testing Set Accuracy: The accuracy on the testing data was 1.0(100%) demonstrating the model's effective generalization to new, unseen data.

Plots:

1.



Training Data with True Labels (Left Plot):

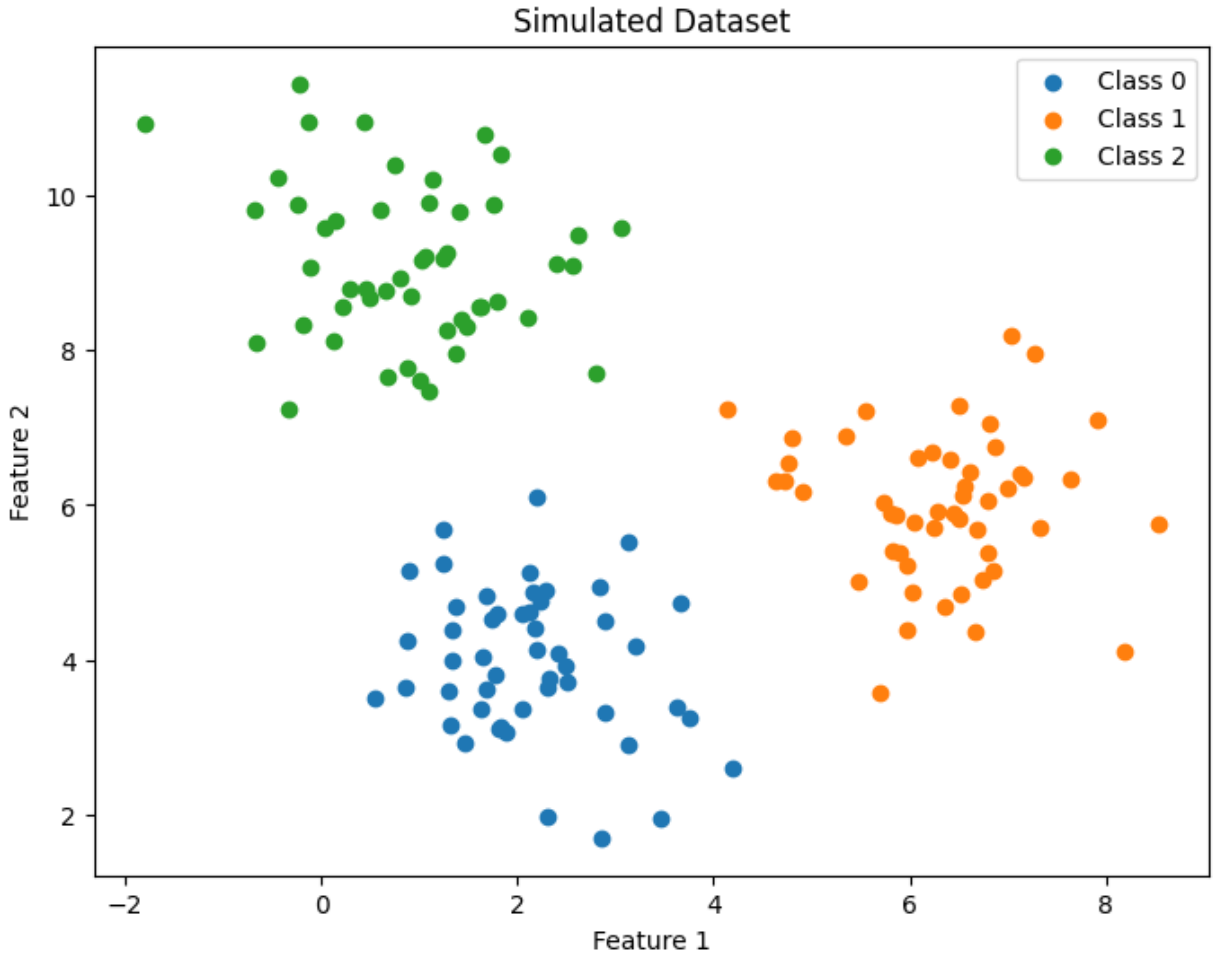
- **Plot Description**: This scatter plot displays the **training data points** classified according to their **true class labels**. Each point represents a sample from the training set, with the color of the point corresponding to its actual class label.

- **Colors**: The color map (viridis) is used to differentiate the three classes visually. Each class has a unique color, making it easy to see how the samples are grouped.
- **Axes**: The x-axis (Feature 1) and y-axis (Feature 2) represent the two features of the dataset.
- **Edge Color**: The points have a black edge (edgecolor='k') to make them stand out more clearly against the background.
- **Marker**: Circular markers (marker='o') are used for the training data.
- **Purpose**: This plot shows how the actual classes in the training set are distributed, helping to understand the underlying structure of the data.

Test Data with Predicted Labels (Right Plot):

- **Plot Description**: This scatter plot shows the **test data points**, where each point is colored according to the **predicted class label** from the KNN classifier.
- **Colors**: As in the training plot, the colors represent the class labels, but here the classes are predicted by the KNN model.
- **Axes**: The x-axis and y-axis again represent the two features of the dataset (Feature 1 and Feature 2).
- **Edge Color**: Similar to the training plot, the points have a black edge to highlight each point.
- **Marker**: Square markers (marker='s') are used to differentiate the test data from the training data.
- **Purpose**: This plot illustrates how well the KNN classifier performed on the test set, showing the predicted classifications of the test samples.

2.



Plot Description: This scatter plot visualizes the **entire simulated dataset** used in the K-Nearest Neighbors (KNN) analysis. Each point in the plot represents a data sample, and the points are color-coded according to their class labels.

Colors and Classes:

- The dataset consists of three distinct classes (Class 0, Class 1, and Class 2). Each class is represented by a different color, making it easy to see the grouping and distribution of the data points.
- The for loop ensures that data points from each class are plotted in a different color, with the condition labels == i selecting the points corresponding to class i.
- The label=f'Class {i}' part adds a legend entry for each class, making it easier to identify which color corresponds to which class.

Axes:

- The x-axis (Feature 1) and y-axis (Feature 2) represent the two features of the dataset, where each point's position is determined by its values for these two features.

Legend:

- The plot includes a legend to indicate which color corresponds to which class. This helps distinguish between the different classes visually.

Purpose:

- This plot provides an overview of the **distribution of the data points** across the three classes. It helps to understand how well-separated or overlapped the classes are, which is crucial for classification tasks.
- The spread and grouping of the points give insights into the complexity of the classification problem, and how well a model like KNN might perform on the dataset.

Conclusion:

The study successfully replicated the performance of the KNN algorithm using a newly generated simulated dataset. The high accuracy scores on both the training and testing sets demonstrate the KNN algorithm's effectiveness in classifying data points based on their features.