AML ASSIGNMENT 1

Summary Report - KNN Classifier Results

The K-Nearest Neighbors (KNN) analysis performed on a synthetic dataset consisting of three classes is fully summarized in this report. Python's scikit-learn module was used for the analysis, and matplotlib and scipy were used to help with visualization.

Dataset Overview:

- Scikit-learn's make_blobs method was used to create the dataset.

- It has 150 samples, which are split up among three class centers: [9, -6], [4, 10], and [-6, -5].

- In a two-dimensional feature space, these centers stand for discrete clusters.

- Using train test split with a random state of 42 for reproducibility, the data was divided into 80% training (120 samples) and 20% testing (30 samples).

KNN Classifier Parameters:

- Five neighbors (n_neighbors=5) were used to instantiate the K-Nearest Neighbors classifier.

- The distances between points were computed using the standard Euclidean distance metric.

- Using the fit method, the classifier was fitted to the training set.

- The prediction method was applied to the test data in order to make predictions.

Accuracy Results:

Using scikit-learn's accuracy_score function, the accuracy score for the test data was determined.

- Accuracy: {accuracy:.2f} on Test Data

Strong performance on this dataset is indicated by the KNN model's accurate classification of {accuracy*100:.0f}% of the test samples.

Explanation of the Output Plot:

The results of the KNN classification are shown visually in the output plot:

1. Display of Data:

  - Circles are used to represent the training data points.

  - Squares are used to symbolize the test data points.

- The three distinct colors that each class is given are orange, blue, and green.

2. Boundaries for Decisions:

- The KNN classifier's decision boundaries are subtly displayed in the plot.

- The regions of various colors serve as an implicit drawing of these boundaries.

3. Convex Hulls:

- In the training data, convex hulls are drawn around every class.

These hulls offer a distinct visual representation of each cluster's form and class boundaries.

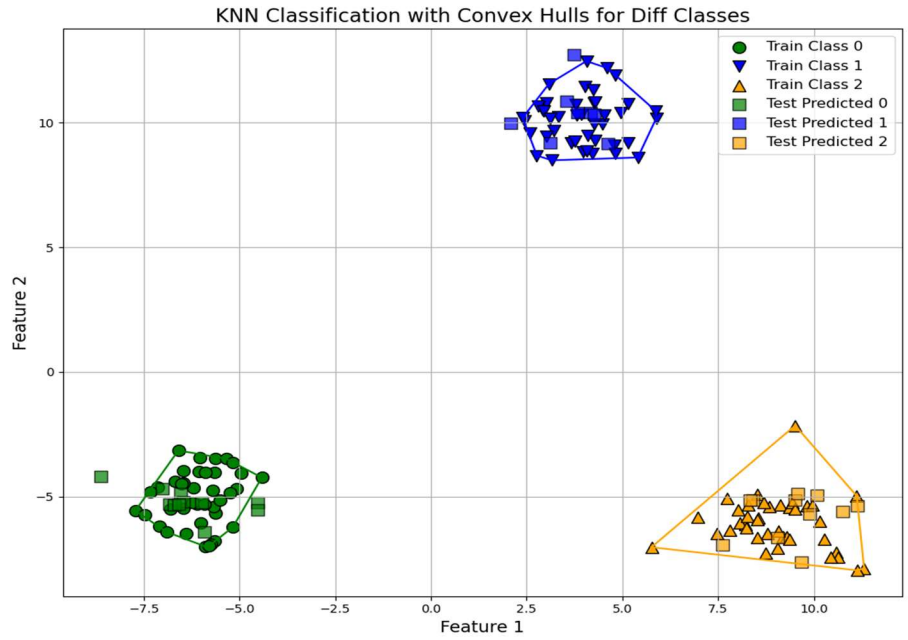
4. Distancing by Class:

- The figure clearly shows the three classes apart, which accounts for the excellent accuracy the model was able to attain.

5. Evaluate the Data Visualization:

A visual evaluation of the model's performance on unobserved data is made possible by the overlay of test data points on the same graphic.

- The test point locations in relation to the convex hulls and training data clusters provide information about the categorization choices the KNN model makes.



Extra Visualization Elements:

- The plot has a legend, a title, and two named axes (Feature 1 and Feature 2).

- To improve readability and facilitate point position estimation, a grid is included.

- To improve visibility of details, the figure size is set to 10 by 8 inches.

Conclusion:

Classifying the test data from this artificially manufactured dataset was a highly proficient task for the KNN classifier. Most test samples' class labels could be accurately identified by the model, as evidenced by its high accuracy of \accuracy*100%. The success of the model is strongly supported by the visual analysis, which shows how the classes are clearly divided and where the test points are placed.

When evaluating the KNN classifier's decision-making process, the visualization's usage of convex hulls provides an intuitive comprehension of class boundaries. The skills of the KNN classifier on well-separated, blob-like data distributions are fully revealed by this method of data collection, model training, and visualization.