CS 4407-01 Data Mining and Machine Learning

Instructor: Professor Shabia Shabir

Name: Anonymos

Programming Assignment Unit 4

**Problem Overview**

This assignment involves using the k-Nearest Neighbors (kNN) algorithm to classify objects into two distinct classes, A and B. Each class contains three training cases represented as coordinates (X, Y) in a two-dimensional space.

The training cases for Class A are represented by the following coordinates:

* A1 = (0, 0)
* A2 = (1, 1)
* A3 = (2, 2)

The training cases for Class B are represented by the following coordinates:

* B1 = (6, 6)
* B2 = (5.5, 7)
* B3 = (6.5, 5)

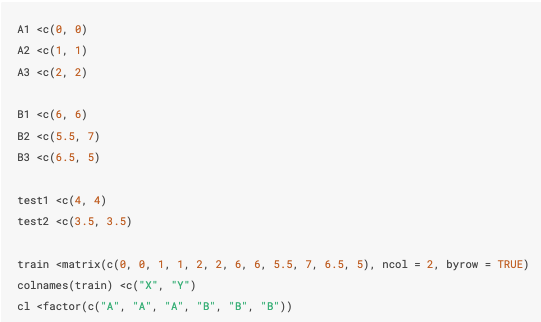
These data points are plotted and visualized using the R programming language. Subsequently, classification labels (`cl`) are constructed, and test cases such as (4, 4) and (3.5, 3.5) are classified to determine which class they belong to. Furthermore, the effect of changing the number of neighbors (k) on classification results is examined. Finally, multiple test cases (T1 = (4, 4), T2 = (3, 3), T3 = (5, 6), T4 = (7, 7)) are processed together to determine which class each case belongs to.

Through this assignment, the following objectives are achieved:

1. Understanding the basic mechanics of the kNN algorithm.
2. Visualizing and interpreting classification results using kNN.
3. Evaluating the impact of k values (number of neighbors) on classification accuracy.

**Data Preparation**

To use the kNN algorithm, training data for Classes A and B is prepared. Each class contains three object instances (cases), which are defined as two-dimensional coordinates. The given data is prepared using the following code:



**Responses to Each Task**

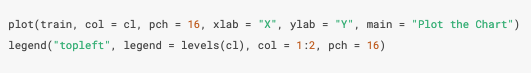
**1. Visualization of Training Data**

Before running the kNN algorithm, the distribution of training data is visualized to understand the characteristics of the dataset. The training data for Classes A and B are plotted on a two-dimensional chart, and the points for each class are visually distinguished.

To plot the data, the `plot()` function in R is used, and the points are color-coded based on the classification labels (`cl`).

Visualization Code

The following R script is used to implement the plot:



* `train`: Training data coordinates
* `col = cl`: Colors points based on class labels
* `pch = 16`: Specifies the shape of points
* `legend`: Displays a legend for Classes A and B

Visualization Result

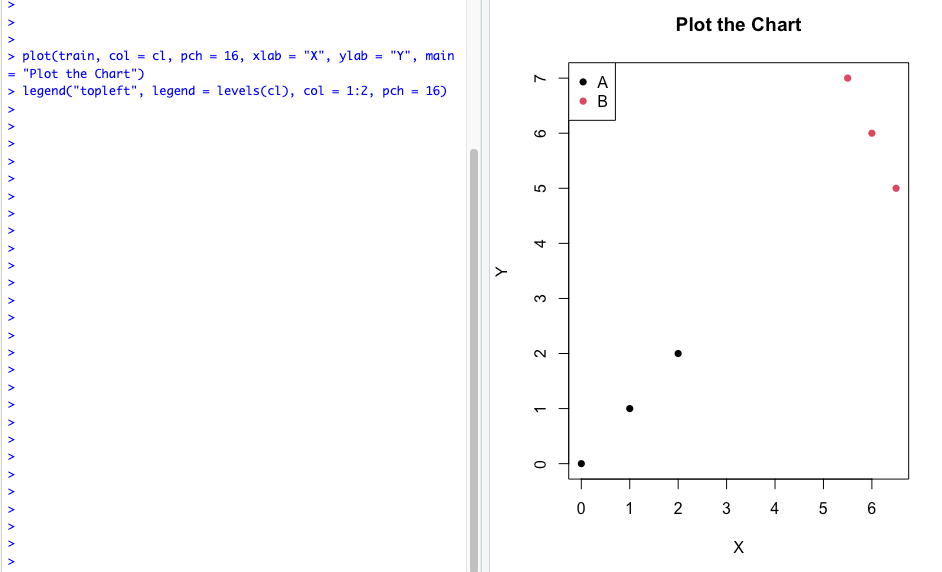


Figure 1: Visualization of Training Data

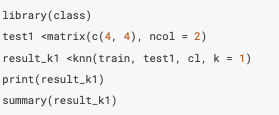
From the plotted results, Classes A and B are clearly located in distinct regions of the two-dimensional space, with each class’s data points clustered in separate areas.

**2. Classification with k = 1**

To understand the basic behavior of the kNN algorithm, k = 1 is set, and the test case (4, 4) is classified. In this case, classification is based on the class of the single nearest training point.

Implementation of Classification

The kNN algorithm is executed in R using the following code:



Classification Result

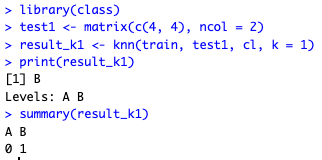


Figure 2: Classification Result for Test Case 1

The results indicate that test case (4, 4) is classified into Class B.

Verification with Euclidean Distance

To validate the classification, the Euclidean distances from test case (4, 4) to each training point are calculated:

Class A:

Distance(A1) = √((4-0)² + (4-0)²) = √32 ≈ 5.66

Distance(A2) = √((4-1)² + (4-1)²) = √18 ≈ 4.24

Distance(A3) = √((4-2)² + (4-2)²) = √8 ≈ 2.83

Class B:

Distance(B1) = √((4-6)² + (4-6)²) = √8 ≈ 2.83

Distance(B2) = √((4-5.5)² + (4-7)²) = √6.25 ≈ 2.50

Distance(B3) = √((4-6.5)² + (4-5)²) = √6.25 ≈ 2.50

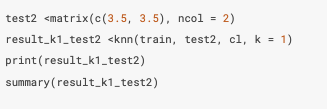
The nearest points are B2 and B3, both at a distance of 2.50. Hence, the kNN algorithm classified (4, 4) into Class B.

**3. Classification of a New Test Case**

Next, the test case (3.5, 3.5) is classified. The kNN algorithm is run with k = 1, and the classification results are validated.

Implementation of Classification

The classification is executed using the following code:



Classification Result

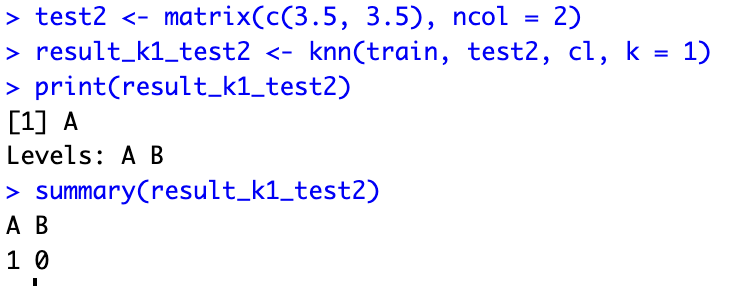


Figure 3: Classification Result for Test Case 2

The results indicate that the test case (3.5, 3.5) is classified into Class A.

Verification with Euclidean Distance

The Euclidean distances from test case (3.5, 3.5) to each training point are calculated:

Class A:

Distance(A1) = √((3.5-0)² + (3.5-0)²) = √24.5 ≈ 4.95

Distance(A2) = √((3.5-1)² + (3.5-1)²) = √12.5 ≈ 3.54

Distance(A3) = √((3.5-2)² + (3.5-2)²) = √4.5 ≈ 2.12

Class B:

Distance(B1) = √((3.5-6)² + (3.5-6)²) = √12.5 ≈ 3.54

Distance(B2) = √((3.5-5.5)² + (3.5-7)²) = √13.25 ≈ 3.64

Distance(B3) = √((3.5-6.5)² + (3.5-5)²) = √14.25 ≈ 3.77

The nearest point is A3 at a distance of 2.12. Hence, (3.5, 3.5) is classified into Class A.

**4. Classification with k = 3 and Multiple Test Cases**

Multiple test cases are classified simultaneously. The test cases include:

T1 = (4, 4)

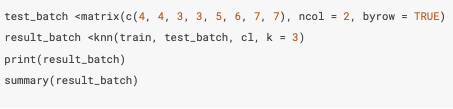
T2 = (3, 3)

T3 = (5, 6)

T4 = (7, 7)

Implementation of Classification

The classification is performed using the following code:



Classification Result

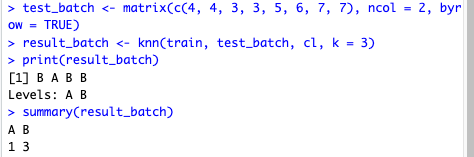


Figure 4: Classification Results for Multiple Test Cases

Among the four test cases, one is classified into Class A, and three are classified into Class B.

Word Count: 807

References

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. New York, NY: Springer.