Department of Computer Engineering
University of Peradeniya
CO544: Machine Learning and Data Mining

Lab 04: Clustering and Association Rule Learning

E/20/420 : WANASINGHE J.K.

Exercise 01

1. Import the iris dataset from scikit-learn. Convert it into an unlabeled dataset by removing the class attribute.

```
# Load the iris dataset
iris = load_iris()
X = iris.data  # Features only, no labels
```

2. Use the Elbow method to identify the best value for k (minimizing WCSS).

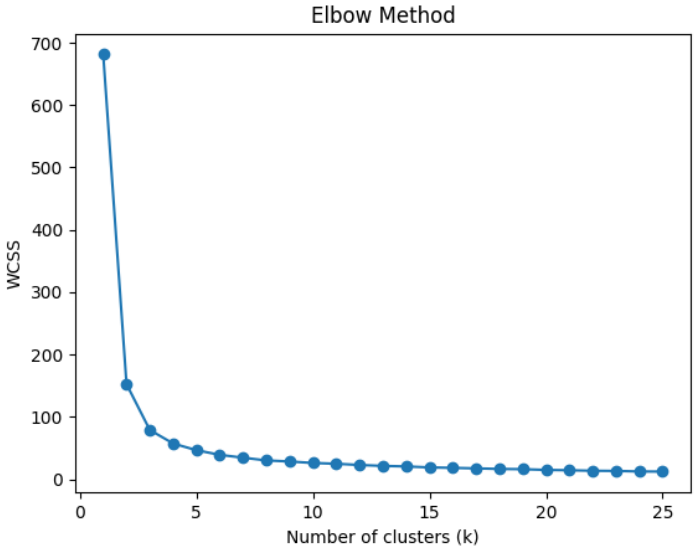

Figure 1:ELBOW METHOD PLOT

The best value for k is **3**, as this is where the plot forms an elbow and adding more clusters does not significantly reduce the WCSS further

3. Fit the K-Means algorithm with the k found in part (b).

4. Explain the output of:

```
kmeans.cluster_centers_
```

where kmeans is your fitted KMeans object.

```
print(kmeans.cluster_centers_)
[[5.88360656 2.74098361 4.38852459 1.43442623]
 [5.006      3.428      1.462      0.246     ]
 [6.85384615 3.07692308 5.71538462 2.05384615]]
```

- `kmeans.cluster_centers_` returns a 2D array where each row represents the coordinates (in feature space) of a cluster center.
- For Iris, it's a 3x4 array (3 clusters, 4 features per sample).
- These centers represent the "mean" feature values for each cluster and can be interpreted as the prototype or most typical sample for each group

5. Visualize the data points and cluster centers in a 3D plot using the first three features as axes.

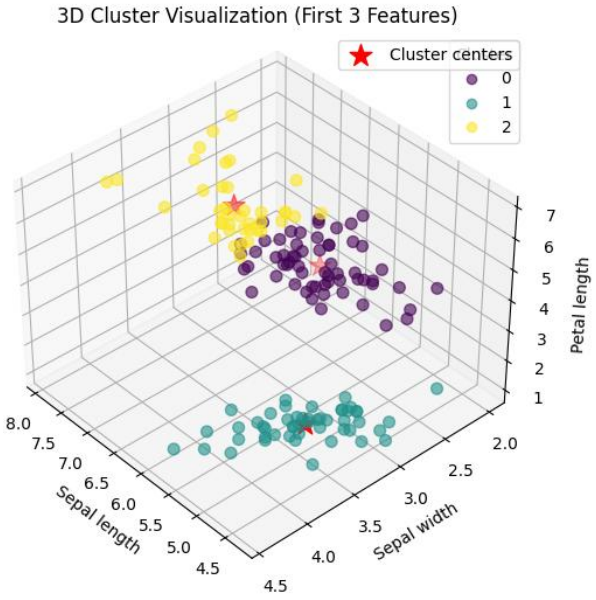

Figure 2: 3D Cluster Visualization

Exercise 02

1. Import the provided groceries.csv dataset.
2. Explore the dataset and build the frequent-item DataFrame.
3. Apply the Apriori algorithm to find itemsets with support > 8%.

	support	itemsets
0	0.080529	(bottled beer)
1	0.110524	(bottled water)
2	0.082766	(citrus fruit)
3	0.193493	(other vegetables)
4	0.088968	(pastry)
5	0.183935	(rolls/buns)
6	0.108998	(root vegetables)
7	0.093950	(sausage)
8	0.098526	(shopping bags)
9	0.174377	(soda)
10	0.104931	(tropical fruit)
11	0.255516	(whole milk)
12	0.139502	(yogurt)

4. Generate association rules using the lift metric.

```
Empty DataFrame
Columns: [antecedents, consequents, antecedent support, consequent support, support,
confidence, lift, representativity, leverage, conviction, zhangs_metric, jaccard,
certainty, kulczynski]
Index: []
```

5. Select one rule and interpret it in your own words.

- **No association rules were found** that meet the default constraints that were set

6. How many rules satisfy both lift > 4 and confidence > 0.8?

```
Number of rules with lift > 4 and confidence > 0.8: 0
```

Based on the given constraints, the answer is 0.