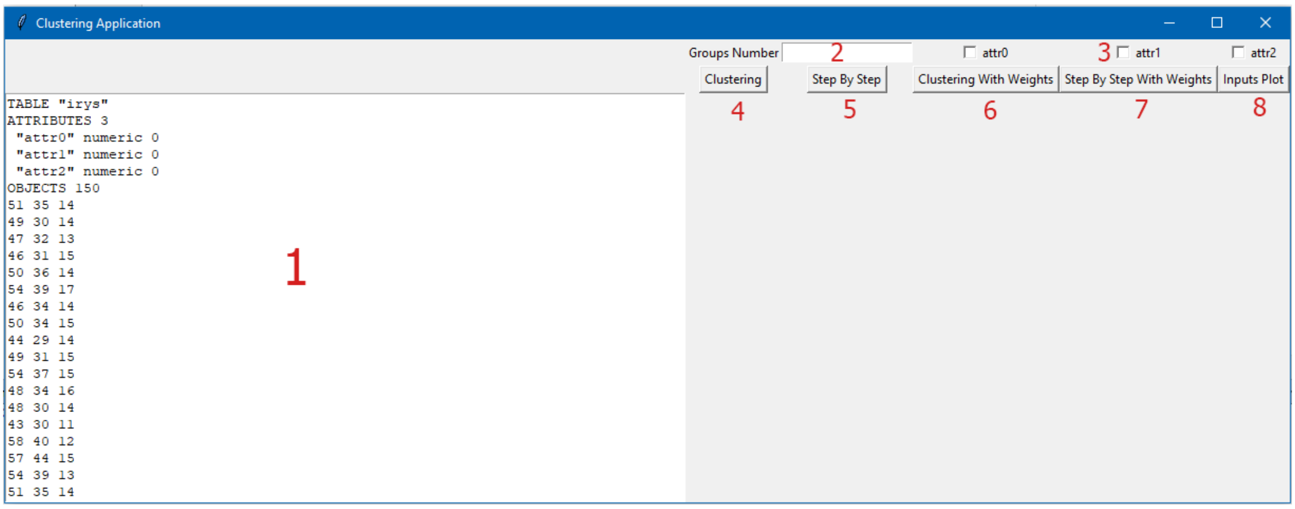
ARTIFICIAL INTELLIGENCE PROJECT

Data Clustering Application With K-Means Algorithm

Name: Sertac

Surname: Bazancir

1- User Interface



```
TABLE "irys"
ATTRIBUTES 3
 "attr0" numeric 0
 "attr1" numeric 0
 "attr2" numeric 0
OBJECTS 150
51 35 14
49 30 14
47 32 13
46 31 15
50 36 14
54 39 17
46 34 14
50 34 15
44 29 14
49 31 15
54 37 15
48 34 16
48 30 14
43 30 11
58 40 12
57 44 15
54 39 13
51 35 14
```

1) File content

2)Desired groups number

3)Attribute selection

4)Clustering and visualization without weights

5)Clustering and step by step visualization without weights

6)Clustering and visualization with weights

7)Clustering and step by step vizualization with weights

8)Visualization for input data

2- K-Means Clustering Algorithm

K-means clustering algorithm is a type of unsupervised learning algorithm. It means the algorithm determine the groups centers itself by number of groups which given by user. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. The algorithm inputs are the number of clusters K and the data set. The algorithm works by following steps;

1) Initial the centroids of each cluster. There are several approaches achieve it.

2) Calculate distance from each data to each centroids and calculate the groups for each data.

3) Update centroids by calculated groups.

4) If any centroid change go step 2

5) Finish

3- Data Set

Data set has 150 object and each object has 3 attribute.

attr0 is an integer number

attr1 is an integer number

attr2 is an integer number

4- Implementation Of K-Means Algorithm

4.A- Initializing The Centroids

Initializing the centroids is very important to implement the algroithm successfully. I have calculated centroids by following algorithm;

k is groups number

centroids is an array which stores the centroids.

least is an array which stores smallest values by attributes. least[0] is the smallest value by attr0 in data set.

big is an array which stores greates values by attributes. big[0] is the greatest value by attr0 in data set.

difference is an array which stores the differences between each centroid by attributes. difference[0] is difference between centroids[i][0] and centroids[i+1][0].

Difference values are calculated by this equation;

```
difference[i] = (big[i]-least[i])/(groups_number-1)
```

Centroids are calculated by this equation;

```
centroids[i][0] = least[0]+((i)*difference[0])
centroids[i][1] = least[1]+((i)*difference[1])
centroids[i][2] = least[2]+((i)*difference[2])
```

4.B- Grouping Datas

In this step, calculate distances from each object to centroids and the groups of object is determined. I use Euclidean Distance for achieve it. It calculates the distance by following equation;

```
distance = np.sqrt(((datas[i][0] - centroids[j][0])**2)*attribute_0.get()*weight_1 + ((datas[i][1] - centroids[j][1])**2)*attribute_1.get()*weight_2 + ((datas[i][2] - centroids[j][2])**2)*attribute_2.get()*weight_3)
```

attribute_0, attribute_1 and attribute_2 is refer to which attributes select for clustering operation.

weight_1, weight_2, weight_3 is refer to clustering operation calculate with weights or without weights and if clustering operation calculate with weight how much is weight values.

After the distances are calculated, the nearest centroid is found and the group of object is determined.

4.C- Update Centroids

Firstly the old centroids copying another array with this command.

```
old_centroids = deepcopy(centroids)
```

After counts and sum arrays are defined. Elements of counts and sum arrays calculated by following equation;

```
sum[groups[i]][j] = sum[groups[i]][j]+datas[i][j]
             counts[groups[i]] += 1
```

After the new centroids are calculated by following equation;

```
centroids[i][j] = sum[i][j] / counts[i]
```

Function return true if old centroid equal to new centroids and it means algorithm will finish. If old centroids not equal to new centroids function return false and it means algorithm go to step 2.