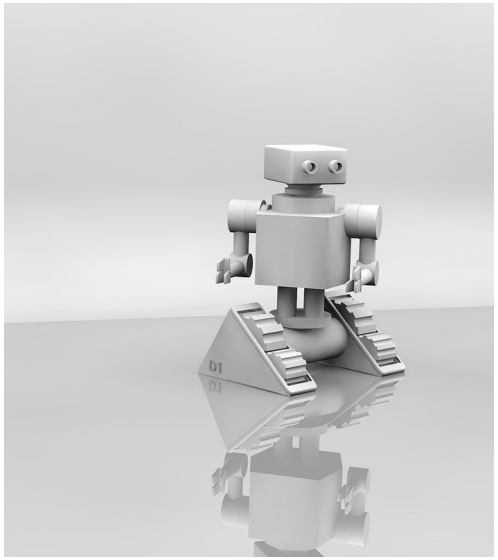


CPSC 481
Artificial Intelligence

Dr. Mira Kim
Mira.kim@fullerton.edu

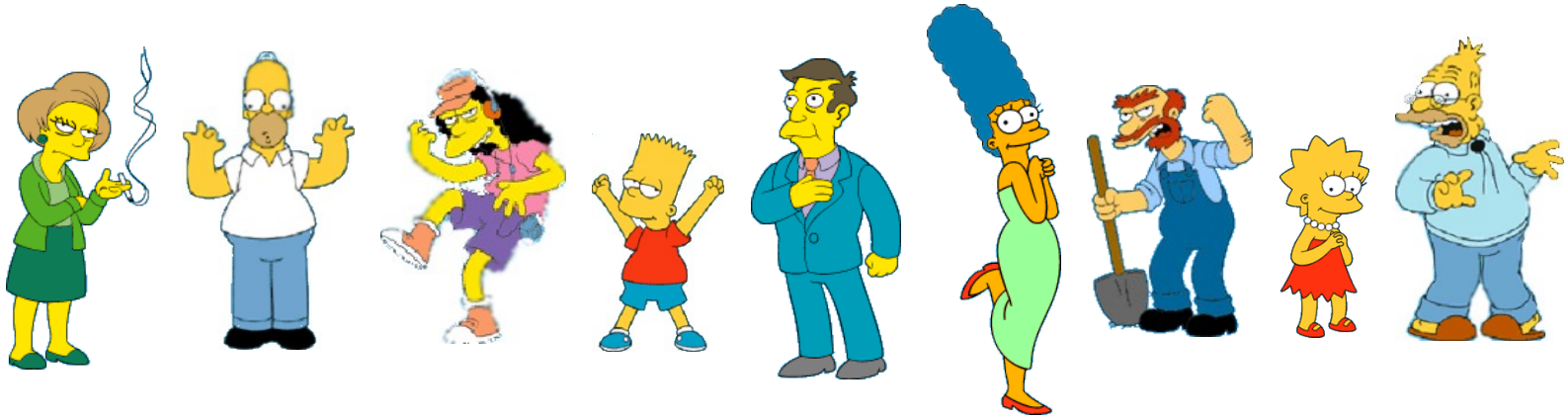
What we will cover this week

- Clustering algorithm
- K-means clustering
- K-means clustering – case study

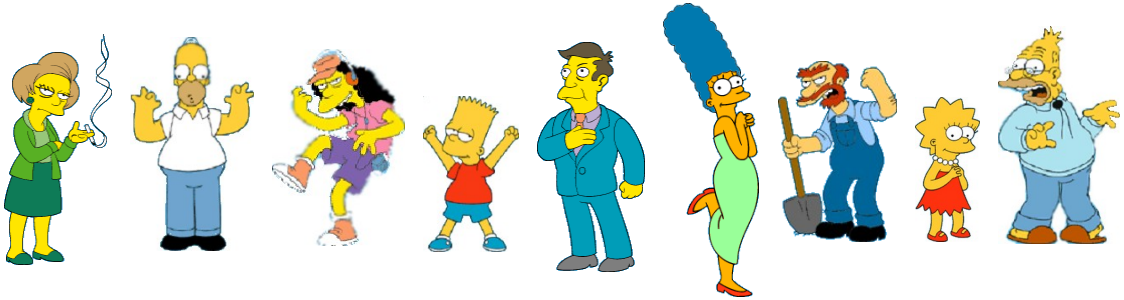
CLUSTERING ALGORITHMS (UNSUPERVISED LEARNING)

Motivation

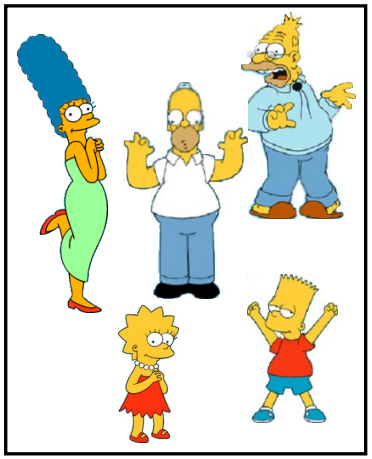
- What is a natural grouping of these objects?



Motivation

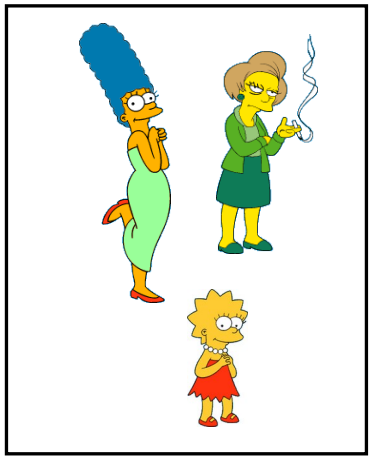


Clustering is subjective!

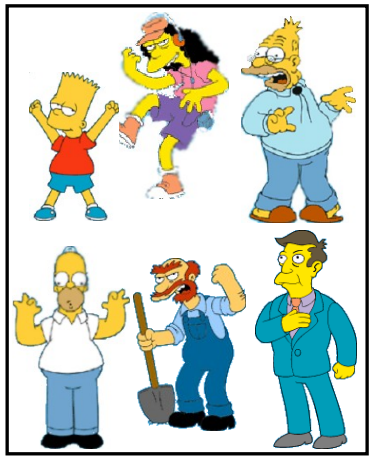


Simpson's Family

School Employees



Females



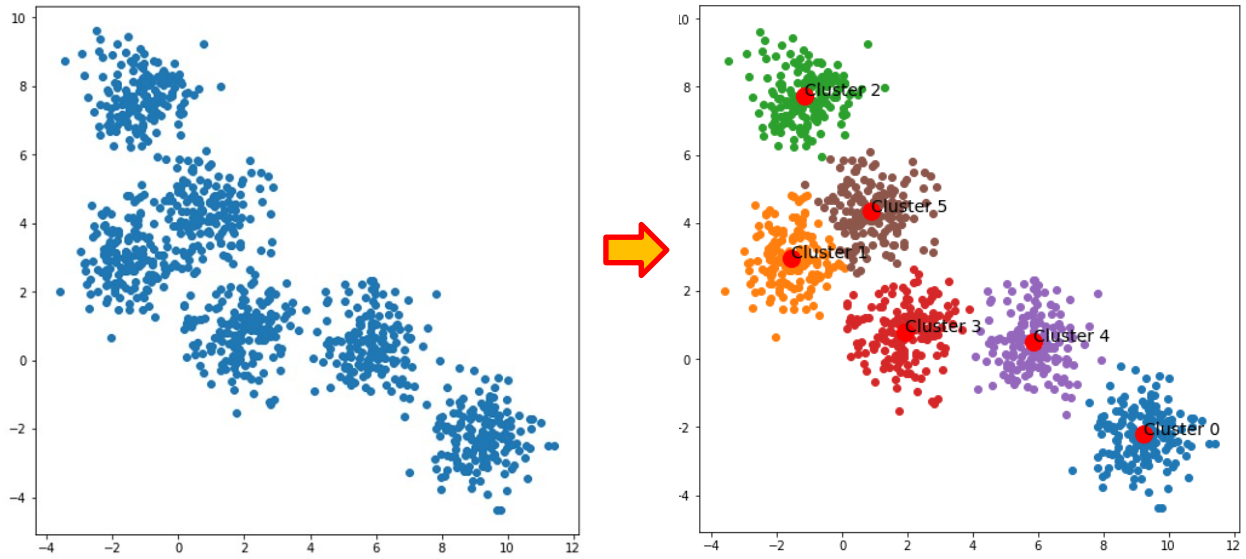
Males

What is Clustering?

- Clustering
  - The process of grouping a set of objects into classes/groups of similar objects ☾ Grouping
  - For clustering various documents,
    - Documents within a cluster should be similar.
    - Documents from different clusters should be dissimilar.
- Cluster
  - A collection of similar data objects

Goal of Clustering

- Given a set of data points, find clusters such that:
  - High Intra-cluster Similarity
    - Degree of closeness of the objects within the same cluster
  - Low Inter-cluster Similarity
    - similarity between different clusters

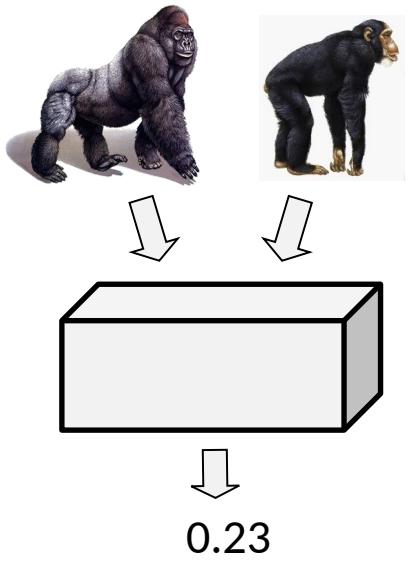


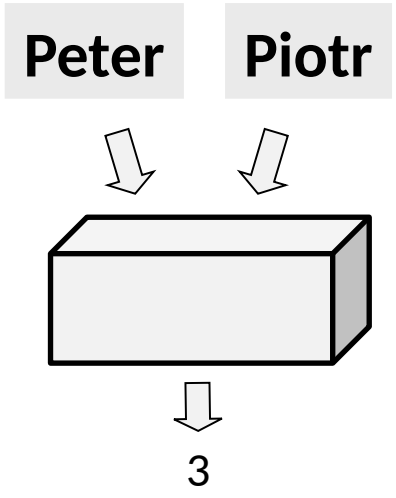
Applications of Clustering

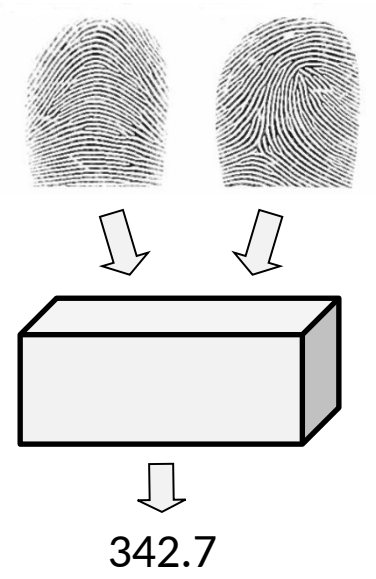
- Marketing
  - Help marketers discover distinct groups in their customer base, and then use this knowledge to develop targeted marketing programs
- Land Use
  - Identifying areas of similar land use in an earth observation database
- Insurance
  - Identifying groups of motor insurance policy holders with a high average claim cost
- Urban Planning
  - Identifying groups of houses according to t geographical location
- Many others

Similarity between Objects

- Definition
  - Let A and B be two objects from the universe of possible objects. The distance (dissimilarity) between the two objects is a real number denoted by *D(A, B)*.







Similarity & Dissimilarity Between Objects

- Euclidean Distance, d(i, j) for two objects
  - Length of Line Segment between the 2 points

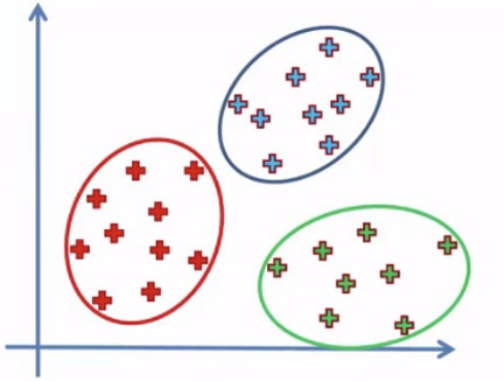
$$d(i,j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \ldots + |x_{ip} - x_{jp}|^2)}$$

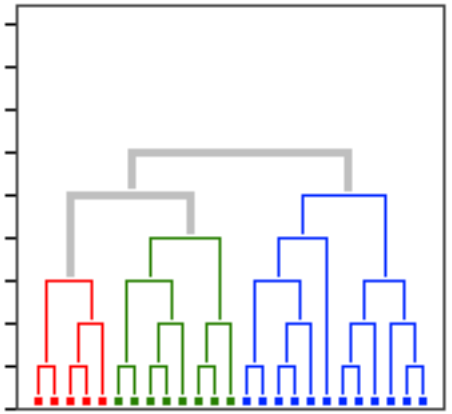
Properties of d(A, B)

- Symmetry
  - D(A, B) = D(B, A)
- Constancy of Self-Similarity
  - D(A, A) = 0
- Positivity (Separation)
  - D(A, B) = 0 IIf A=B
- Triangular Inequality
  - D(A, B) $\leq$ D(A, C) + D(B, C)

Types of Clustering Approaches

- Centroid-based Clustering
  - K-Means Clustering
    - Partitioned into k distinct clusters based on distance to the centroid of a cluster
- Connectivity-based Clustering
  - Hierarchical Clustering
    - Build nested clusters by merging or splitting them successively
    - Hierarchy of clusters is represented as a tree (dendrogram)





Types of Clustering Approaches

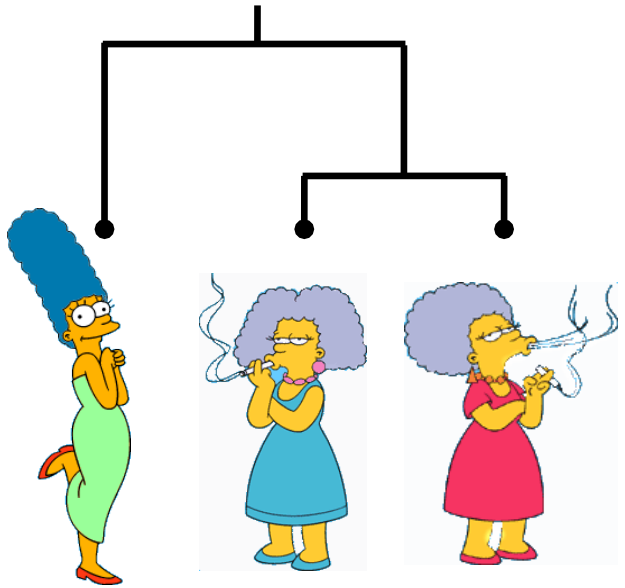
- Density-based Clustering
  - Guided by connectivity and density functions
  - DBSCAN
- Others

Hard vs. Soft Clustering

- Hard Clustering
  - Each data item belongs to exactly one cluster.
  - More common and easier to do
- Soft Clustering
  - A data item can belong to more than one cluster
  - Applicable to some applications like creating browsable hierarchies
    - Example) A pair of Sneakers can be in two clusters:
      - » Cluster of Sports Apparel
      - » Cluster of Shoes
    - Example 2) A news article called “The impact of AI on Modern Economy” can be in 3 clusters:
      - » Cluster of Economies, membership score of 0.7
      - » Cluster of Technology, score of 0.5
      - » Cluster of Politics, score of 0.1

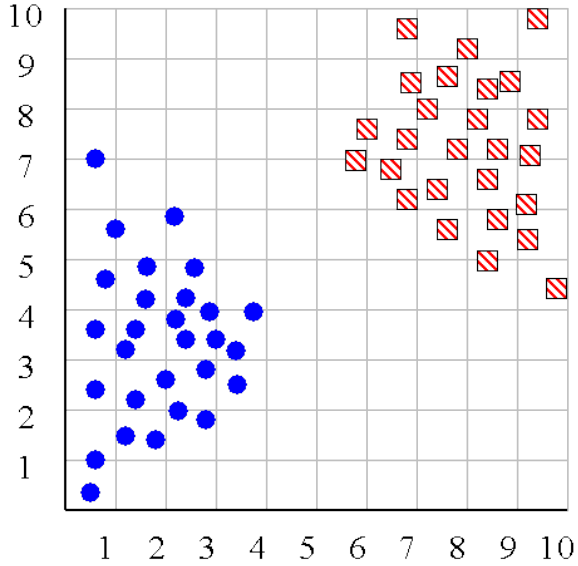
Dendogram

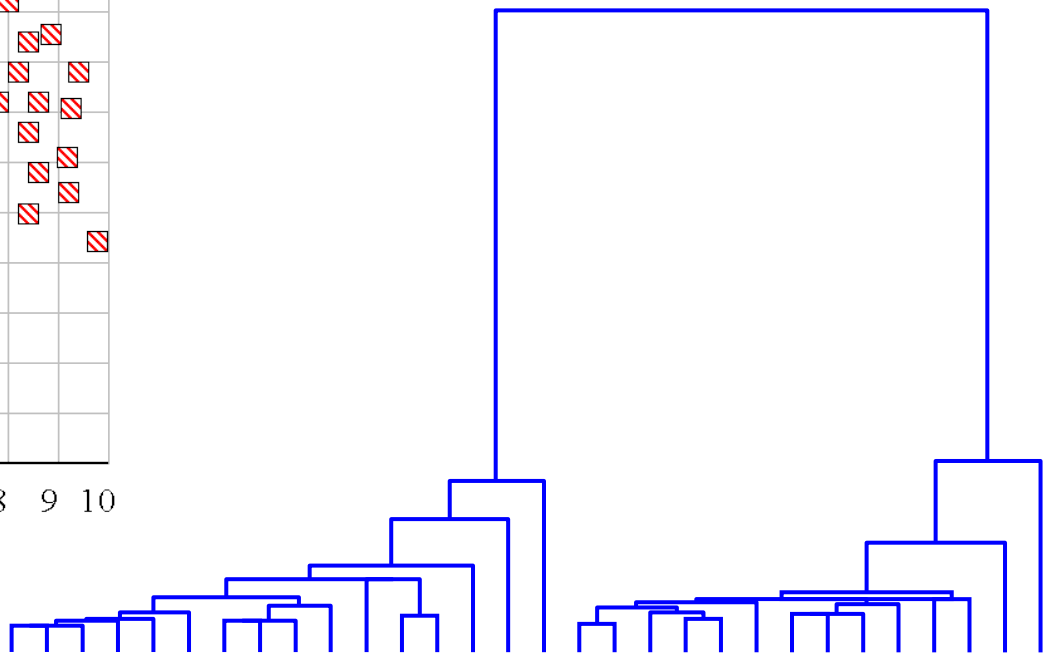
- Summarizing the Similarity
  - Represent the arrangement of clusters produced by hierarchical clustering
  - The similarity is represented as the height of the lowest internal node they share.



Dendogram

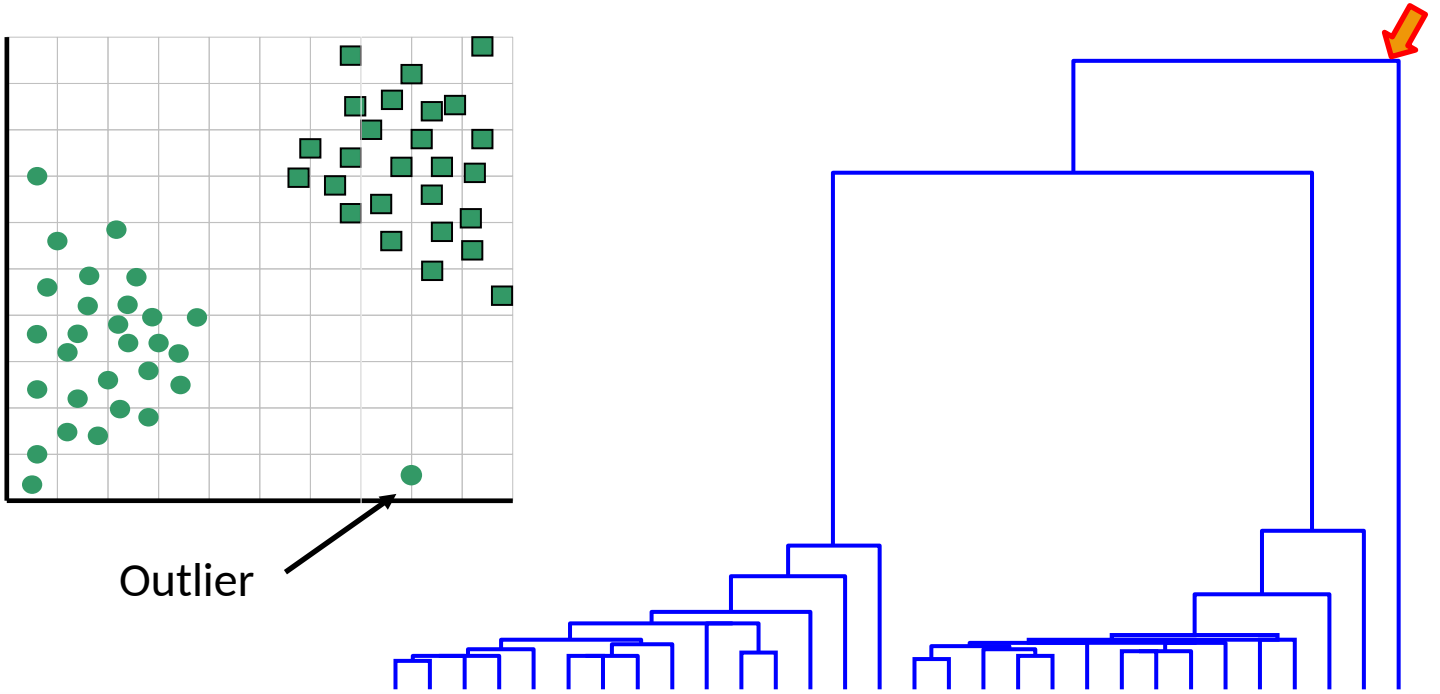
- Utilize the dendrogram to determine the “correct” number of clusters by looking at the “big jumps”





Dendogram

- Utilize the dendrogram to detect the outliers.
  - The single isolated branch is suggestive of a data point that is very different to all others.



K-MEANS CLUSTERING ALGORITHM

K-Means Clustering

- Algorithm to cluster n objects based on attributes into K partitions, where K < n and K is positive integer number.
- Attempt to find the centers of natural clusters in the data.
- Assumption
  - It assumes that the object attributes form a vector space.

How to Cluster?

- By partitioning n objects into K disjoint subsets S_j containing data points so as to minimize the sum-of-squares criterion.

  – where x_n is a vector representing the n^{th} data point and
    μ_j is the geometric centroid of the data points in S_j.

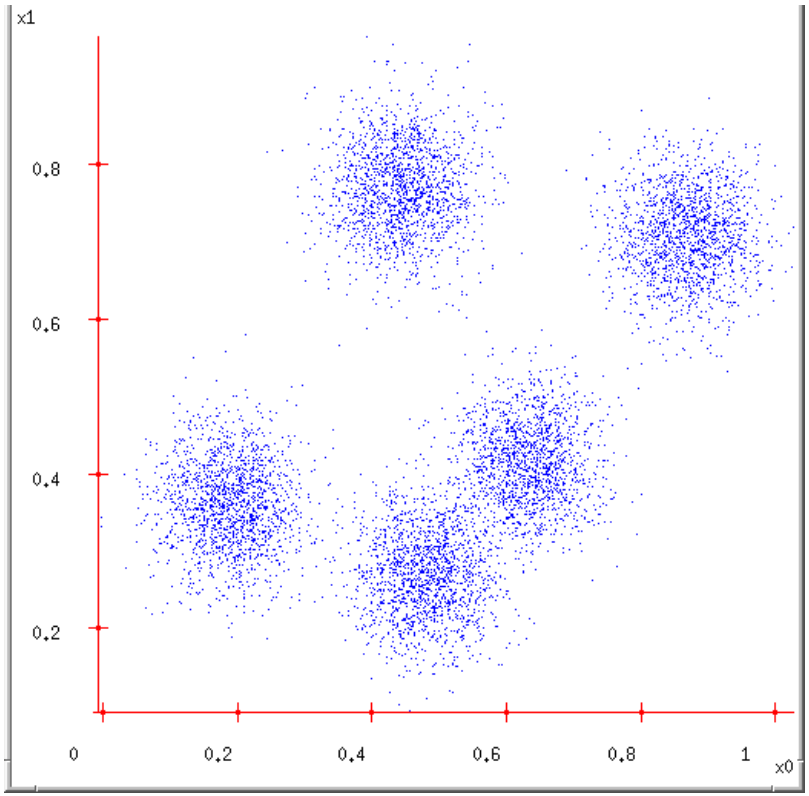
- The clustering is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

Computing Distance

- Distance Measure
  - How the similarity of two elements is calcula
- Euclidean Distance, 2-norm distance $d(x,y) = \sum_{i=1}^{p} |x_i - y_i|$
- Manhattan Distance. Taxicab Norm or 1-norm $d(x,y) = \max_{1 \le i \le p} |x_i - y_i|$
- Maximum Norm $d(x,y) = \sqrt[2]{\sum_{i=1}^{p} |x_i - y_i|^2}$
- Other Approaches
  - Mahala-nobis Distance
    - Corrects data for different scales and correlations in the variables.
  - Inner Product Space
    - The angle between two vectors can be used as a distance measure when clustering high dimensional data
  - Hamming Distance, Edit Distance
    - Measures the minimum number of substitutions required to change one member into another.

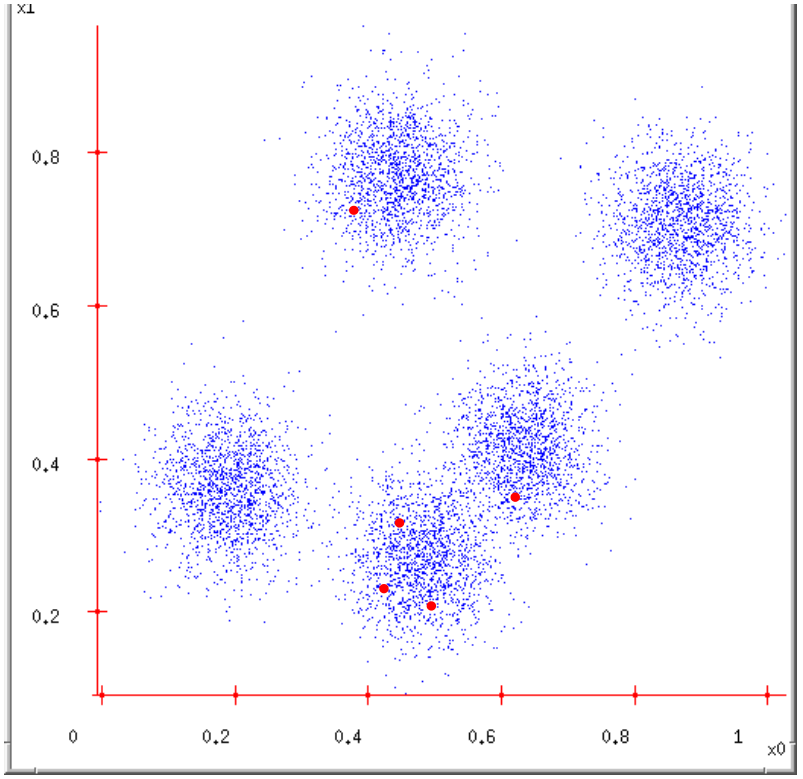
K-means - Initialization

- Ask user how many clusters they'd like.
  - e.g., k=5



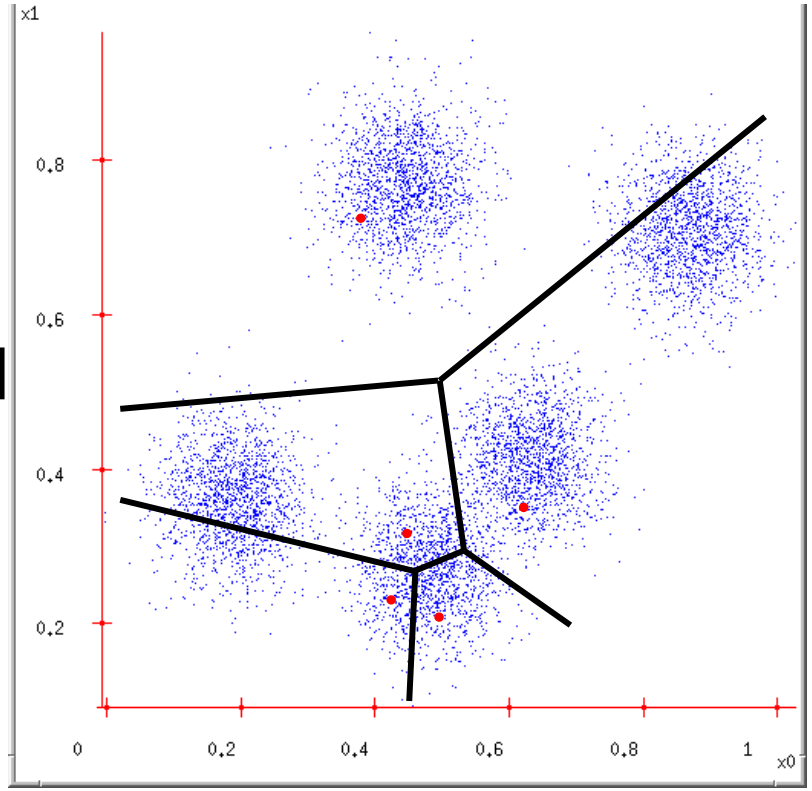
K-means - Assignment

- Ask user how many clusters they'd like.
  - e.g., k=5
- Randomly guess k centroids.



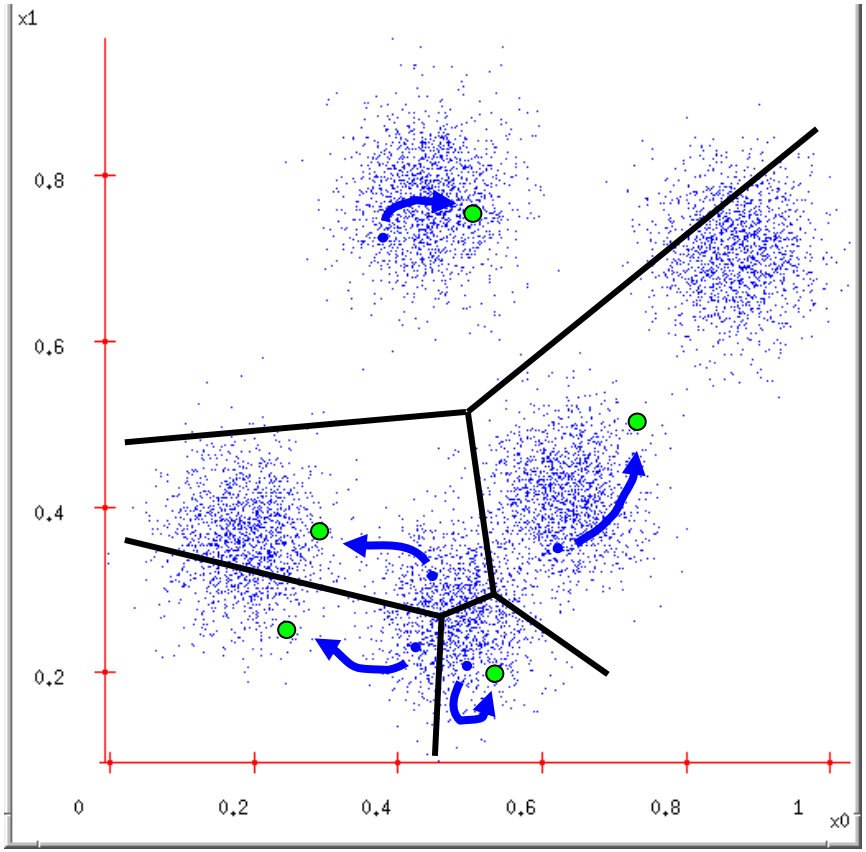
K-means – Find the center

- Ask user how many clusters they'd like.
  - e.g., k=5
- Guess k centroids.
- Compute distance from every point from centroid and cluster them accordingly.



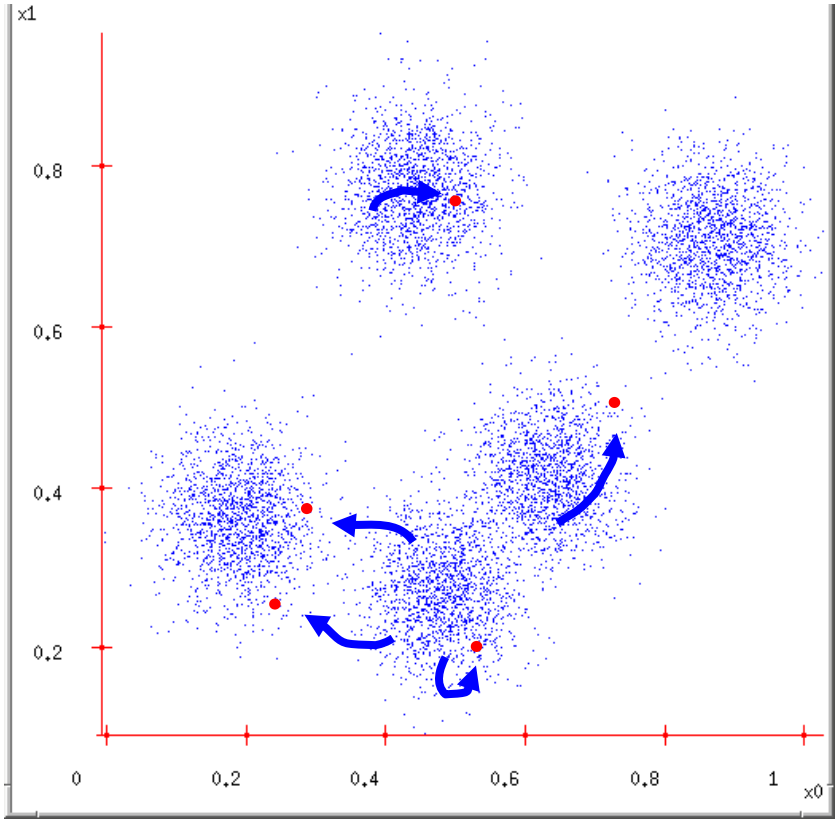
K-means – Adjust centroids

- Ask user how many clusters they'd like.
  - e.g., k=5
- Randomly guess k centroids.
- Compute distance from every point from centroid and cluster them accordingly.
- Adjust centroids so they become the center for given clusters

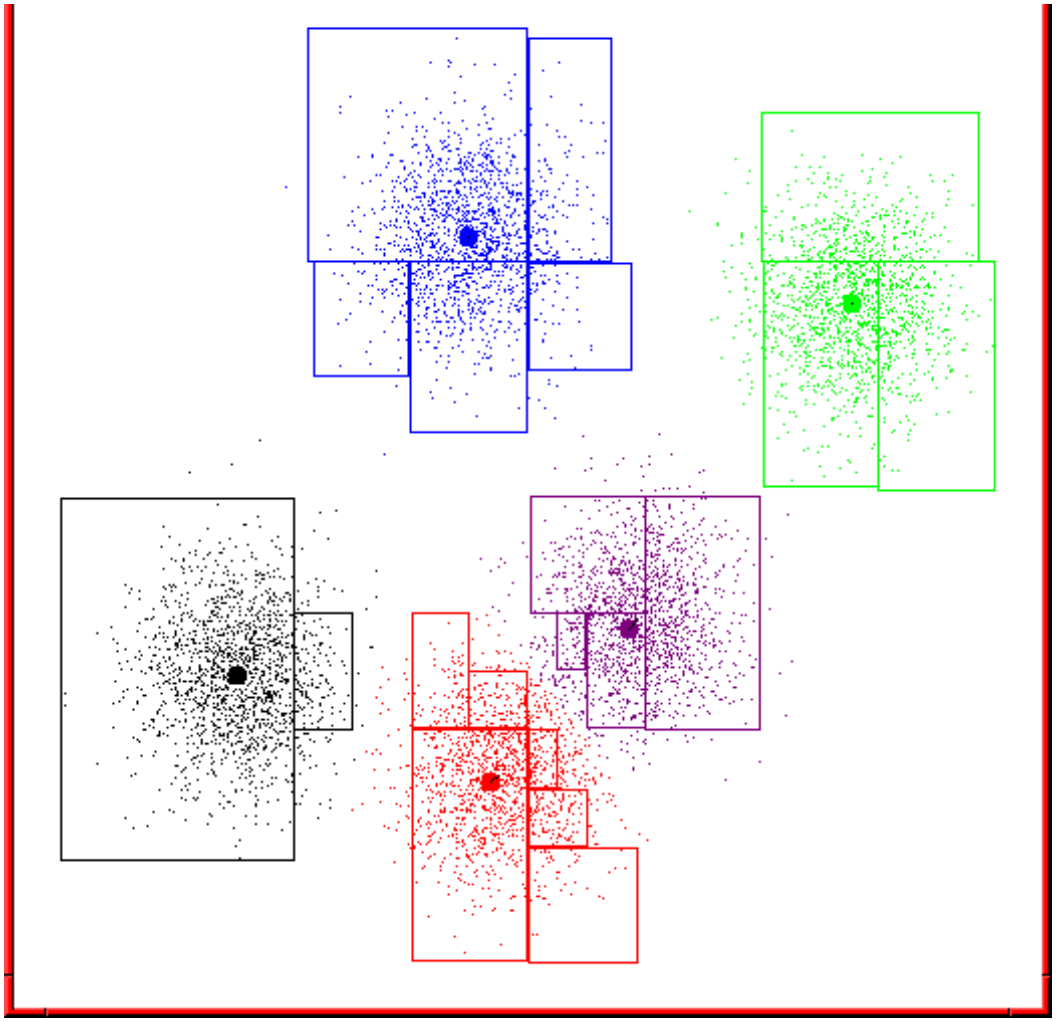


K-means - Iteration

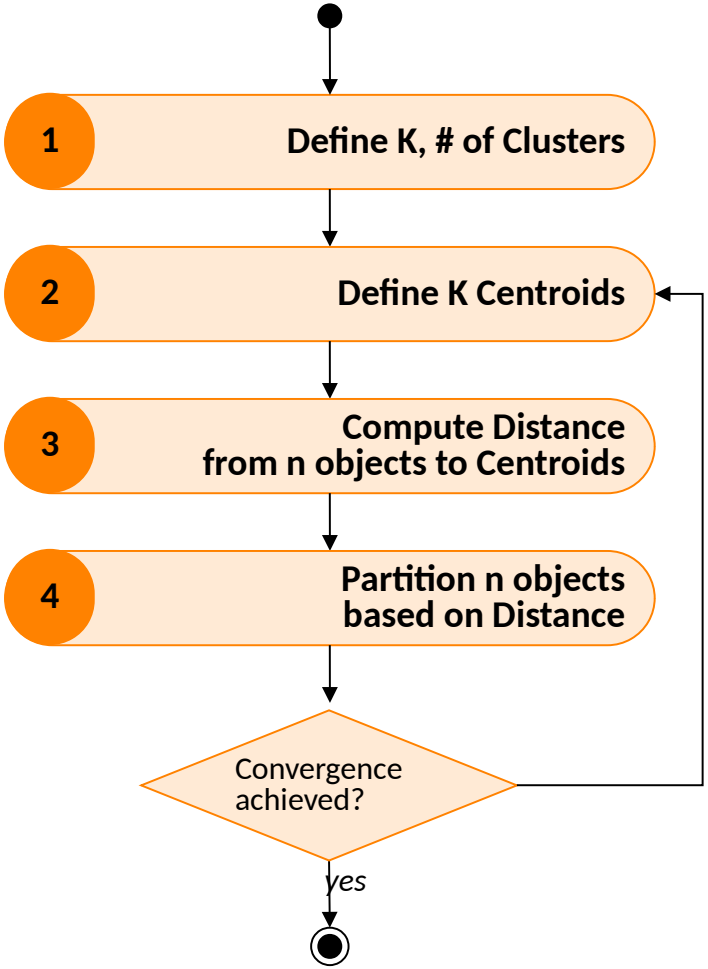
- Ask user how many clusters they'd like.
  - e.g., k=5
- Randomly guess k centroids.
- Each datapoint finds out which Center it's closest to.
- Adjust centroids so they become the center for given clusters
- …Repeat until terminated!



K-means continues…

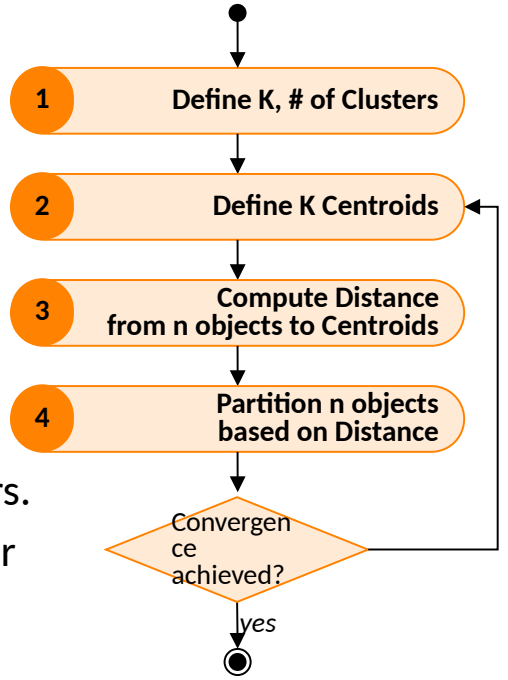


Process of K-Mean Clustering

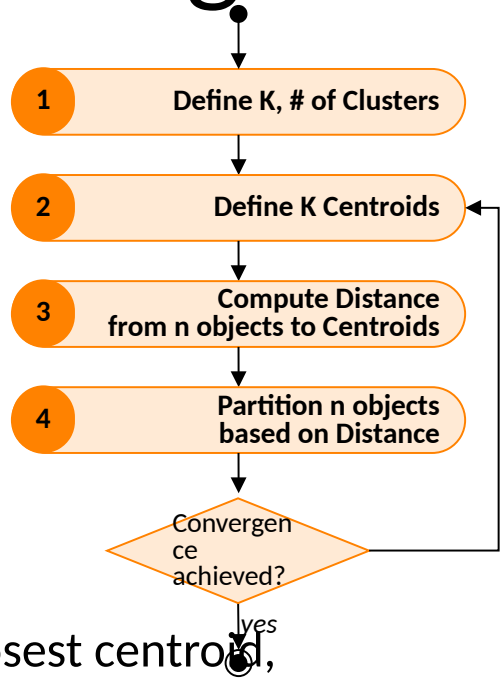


Process of K-Mean Clustering

- Step 1
  - Enter the value of 'K', the number of clusters.
- Step 2
  - Define K number of centroids.
  - Initially,
    - May assign the training samples randomly
    - Or, may define systematically
      - Take the first k training samples as single-element clusters.
      - Assign each remaining training sample (n-K) to the cluster with the nearest centroid.
  - For Subsequent Iterations,
    - Compute the new centroids from the clusters.



Process of K-Mean Clustering



- Step 3
  - Compute the distances from each of n objects to the centroids.
- Step 4
  - Partition n objects to the cluster with the closest centroid.
    - If an object is not currently in the cluster with the closest centroid, move this sample to the cluster with the closed centroid.
- Decision Node
  - Repeat steps 2, 3, and 4 until the convergence is achieved.

K-MEANS CLUSTERING - CASE STUDY

Wholesale Customer Clustering

- Target System
  - is to make customer groups based on their types of purchase
- Clustering Algorithm to Apply
  - K-Means Clustering

Dataset of Wholesale Customer

- Source
  - UCI Machine Learning Repository
- ‘Wholesale Customer Dataset’
  - Collected from wholesales in Lisbon, Portugal
    - Annual spending money considering purchase item division for each customer
  - Collection of 440 Customers’ Data
  - 8 Features
    - Annual spending from Fresh, Milk, Grocery, Frozen, Detergent Paper, and Delicatessen
    - Customer’s Channel
      - Horeca (Hotel/Restaurant/Cafe) or Retail channel
    - Customer’s Region
      - Lisnon, Oporto or other

Dataset of Wholesale Customer

Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
2	3	12669	9656	7561	214	2674	1338
2	3	7057	9810	9568	1762	3293	1776
2	3	6353	8808	7684	2405	3516	7844
1	3	13265	1196	4221	6404	507	1788
2	3	22615	5410	7198	3915	1777	5185
2	3	9413	8259	5126	666	1795	1451
2	3	12126	3199	6975	480	3140	545
2	3	7579	4956	9426	1669	3321	2566
1	3	5963	3648	6192	425	1716	750
2	3	6006	11093	18881	1159	7425	2098
2	3	3366	5403	12974	4400	5977	1744
2	3	13146	1124	4523	1420	549	497
2	3	31714	12319	11757	287	3881	2931
2	3	21217	6208	14982	3095	6707	602
2	3	24653	9465	12091	294	5058	2168
1	3	10253	1114	3821	397	964	412
2	3	1020	8816	12121	134	4508	1080
1	3	5876	6157	2933	839	370	4478
2	3	18601	6327	10099	2205	2767	3181
1	3	7780	2495	9464	669	2518	501

Library for K-Means Clustering

- Scikit-Learn
  - ML Library for Python
- Hyper-parameters
  - n_cluster: 4
    - The Number of Clusters to form and the number of centroid to generate
  - init: ‘k-means++’
    - Default, Method for initializing centroid
    - “K-means++” selects initial cluster centers for k-mean clustering in a smart way to speed up convergence.
  - n_init: 10
    - Default, Number of time the k-means algorithm will be run with different centroid seeds.
  - max_iter: 300
    - Default, Maximum number of iterations of the k-means algorithm for a single run.

Evaluation Metrics Applied

- Silhouette Score
  - Used to evaluate the quality of clusters generated by clustering algorithms such as K-Means.
  - It measures the distance as a mean nearest-cluster distance.

    - Let a_i be mean intra-cluster distance from instance *i* to other instances.
    - Let b_i be the mean nearest-cluster distance from instance *i* to other instances.
    - Range: -1..1
      - As result is closed to 1, well-separated clusters
      - As result is closed to 0, overlapping clusters
      - As result is closed to -1, poor clustering

Step 1. Loading Dataset

- To load dataset
  - To load dataset from local .CSV file

```
def load_dataset(self, path):
    """
    To load dataset from local
    :param path: string, dataset path
    :return:
    """
    self.data = pd.read_csv(path)    # To load data
    print(self.data.head())
    print(self.data.info())
```

Step 2. Preprocessing Dataset

- To normalize whole values in each feature
  - To normalize range of values to 0..1

```
def preprocess_dataset(self):
    """
    To normalize whole data from 0 to 1
    :return:
    """
    mms = MinMaxScaler()
    d = mms.fit_transform(self.data)    # To normalize data from 0 to 1
    self.data = pd.DataFrame(data=d, columns=self.data.columns) # To change type
    from npadrray to dataframe
    print(self.data.info())
    print(self.data.head())

    self.data.sample(frac=1)    # To shuffle data
```

Step 2. Preprocessing Dataset

- Result of Normalization

Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
1.0	1.0	0.112940	0.130727	0.081464	0.003106	0.065427	0.027847
1.0	1.0	0.062899	0.132824	0.103097	0.028548	0.080590	0.036984
1.0	1.0	0.056622	0.119181	0.082790	0.039116	0.086052	0.163559
0.0	1.0	0.118254	0.015536	0.045464	0.104842	0.012346	0.037234
1.0	1.0	0.201626	0.072914	0.077552	0.063934	0.043455	0.108093
1.0	1.0	0.083907	0.111706	0.055218	0.010535	0.043896	0.030204
1.0	1.0	0.108098	0.042809	0.075148	0.007478	0.076842	0.011306
1.0	1.0	0.067554	0.066732	0.101566	0.027020	0.081276	0.053463
0.0	1.0	0.053144	0.048922	0.066708	0.006574	0.041961	0.015582
1.0	1.0	0.053527	0.150293	0.203477	0.018638	0.181805	0.043700
1.0	1.0	0.029987	0.072818	0.139808	0.071905	0.146335	0.036316
1.0	1.0	0.117193	0.014556	0.048719	0.022927	0.013374	0.010305
1.0	1.0	0.282760	0.166987	0.126691	0.004306	0.094993	0.061076
1.0	1.0	0.189161	0.083779	0.161452	0.050457	0.164217	0.012495
1.0	1.0	0.219799	0.128127	0.130291	0.004421	0.123824	0.045161
0.0	1.0	0.091397	0.014419	0.041152	0.006114	0.023540	0.008531
1.0	1.0	0.009068	0.119290	0.130614	0.001791	0.110352	0.022466
0.0	1.0	0.052368	0.083085	0.031581	0.013378	0.008990	0.093346
1.0	1.0	0.165834	0.085400	0.108820	0.035829	0.067705	0.066291
0.0	1.0	0.069346	0.033223	0.101976	0.010584	0.061606	0.010388

Step 3. Clustering Instances

- To cluster dataset using K-Means algorithm
  - To set the number of clusters to 4
  - To use *fit_predict()* method for clustering
    - Fits the model to the data
    - Predicts the cluster labels for the same data

```
def cluster(self, n_cluster=4):
    """
    To cluster dataset
    :param n_cluster:
    :return: array, a list of cluster ids
    """
    self.km = KMeans(n_clusters=n_cluster)  # To define model
    result = self.km.fit_predict(self.data)     # To cluster

    return result
```

Step 3. Clustering Instances

- Result of Clustering
  - To cluster instances (whole customers information) as 4 clusters
  - To show the cluster ID of each instance

```
[0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 1 1 0 0 0 1 1 0 1 1 1 1 1 1 0 1
 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 1 0 1 0 1 1 1 1 1 0
 0 1 1 0 1 1 1 0 0 1 0 0 0 1 1 1 1 1 0 1 0 1 0 1 1 1 0 0 0 1 1 1 0 0 0 0 1
 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
 1 1 1 1 1 1 1 0 0 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 0 1 0 1 1 1 1 1 1 1 1 1
 1 1 1 0 0 1 1 1 0 1 1 2 3 2 2 3 3 2 2 2 3 2 3 2 3 2 3 2 2 3 2 3 2 3 2 2 2
 2 3 2 2 3 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 3 2 2 2 2 2 2 2
 2 2 2 2 2 3 2 3 2 3 2 2 2 2 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 3 2 3
 2 3 3 2 3 3 3 3 3 3 3 2 2 3 2 2 3 2 2 3 2 2 2 3 2 2 2 2 2 2 2 2 2 2 3 2
 3 3 3 2 2 2 2 0 0 1 0 1 1 0 0 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1
 0 1 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
 0 0 1 1 1 1 1 1 0 0 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1]
```

Step 4. Evaluating Result

- To compute silhouette score
  - To use *silhouette_score()* method in scikit learn library
    - The average silhouette score for whole instances is returned.

```
def compute_silhouette_score(self, clusters):
    """
    To compute silhouette score considering input # of clusters
    :param clusters: nparray, clustering results
    :return: double, the score
    """
    score = silhouette_score(self.data, clusters)     # To compute silhouette
    score
    return score    # To return silhouette score
```

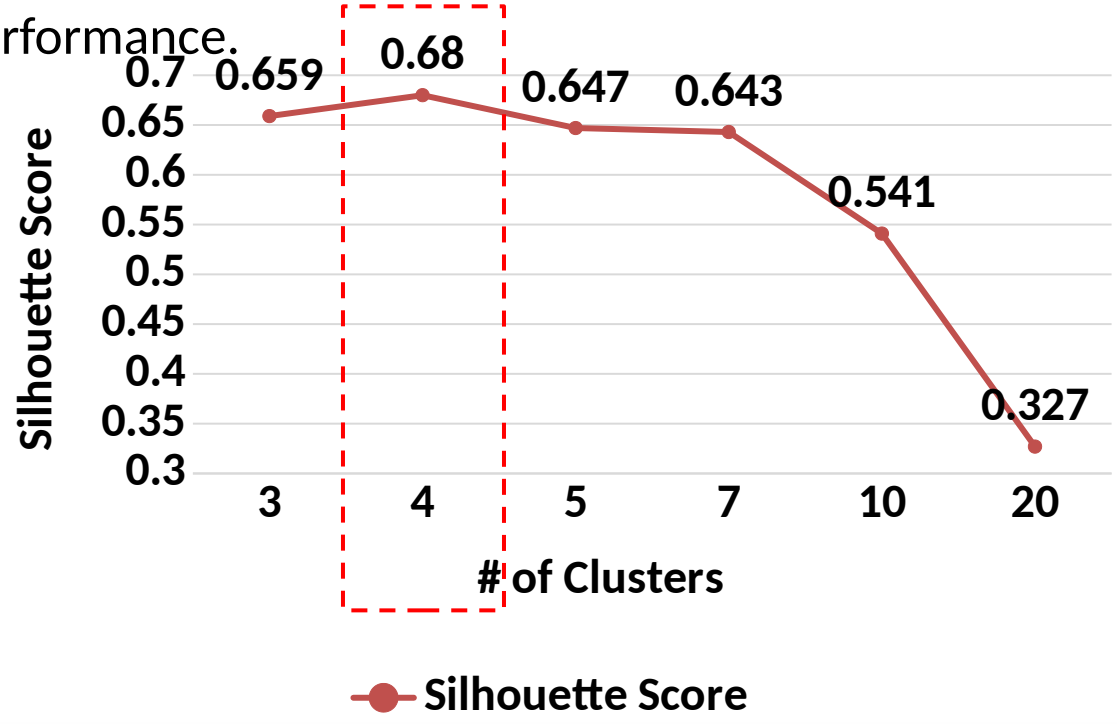
  - Result
    - Silhouette Score for 0.680

Step 5. Fine-tuning Model

- Apply clustering with different numbers of clusters
  - Case 1. 3 clusters
  - Case 2. 5 clusters
  - Case 3. 7 clusters
  - Case 4. 10 clusters
  - Case 5. 20 clusters

Step 5. Fine-tuning Model

- Result of Evaluation
  - To evaluate cluster results using silhouette score
    - The result clustered by 4 centroids shows the highest performance.



References

- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 3rd Edition