Cognixia

A Collabera LEARNING SOLUTIONS COMPANY

Introduction to Clustering

www.cognixia.com

Agenda

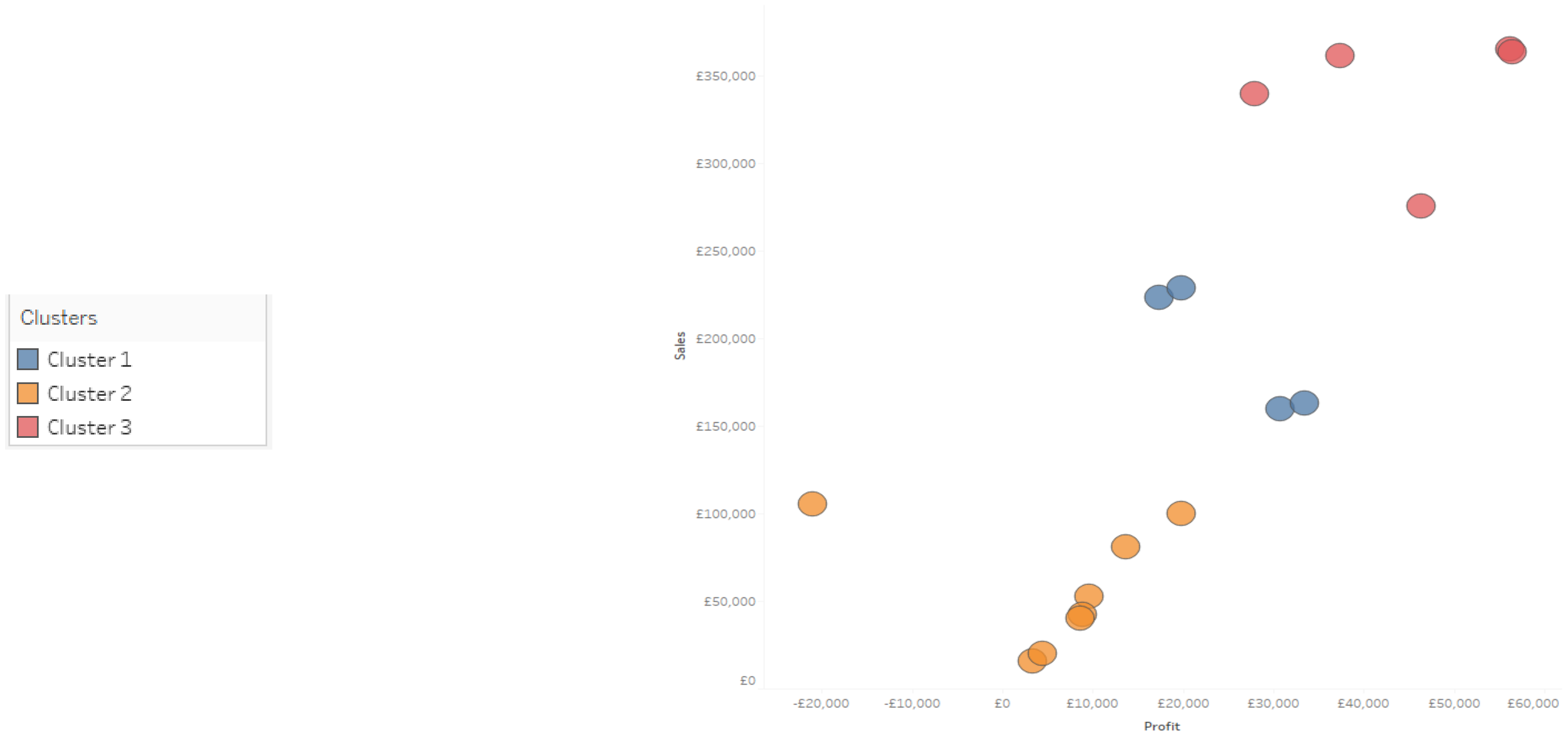
1. Types of Clustering
2. K-means Clustering
3. Agglomerative Clustering

Clustering

Unsupervised Learning

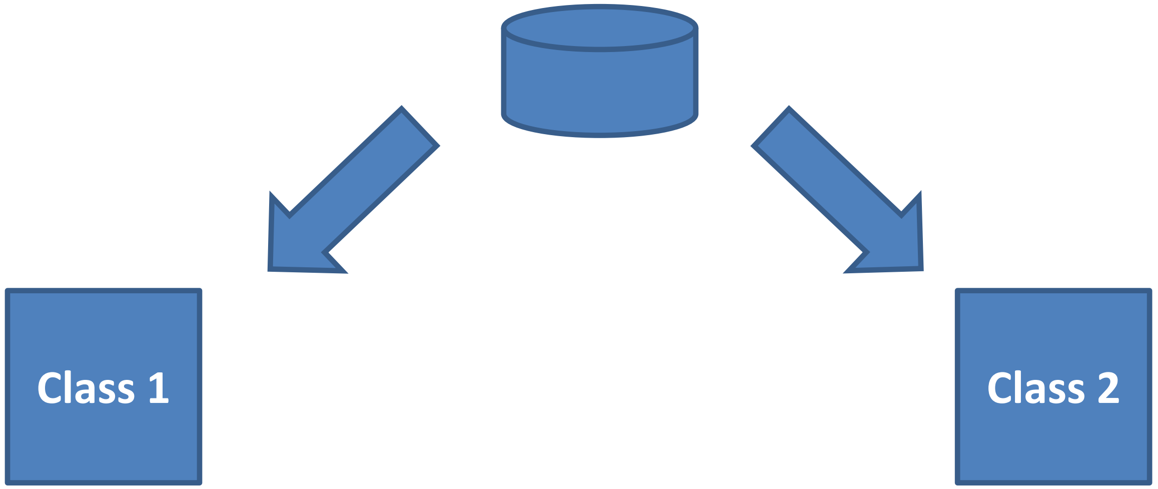
Sometimes the data is unstructured and it will be difficult to classify the data in different categories.

Here unsupervised learning helps to solve the problem. This learning is used to cluster the input data in classes on the basis of their statistical properties.



What is clustering?

Clustering means grouping of objects based on the information found in the data features



The goal is that objects in one group will be similar to one other and different from objects in another group

Clustering

- The objects in group 1 should be as similar as possible.
- But there should be much difference between an object in group 1 and group 2
- The attributes of the objects are allowed to determine which objects should be grouped together

Group 1

Group 2

Group 3

Group 4

Clustering

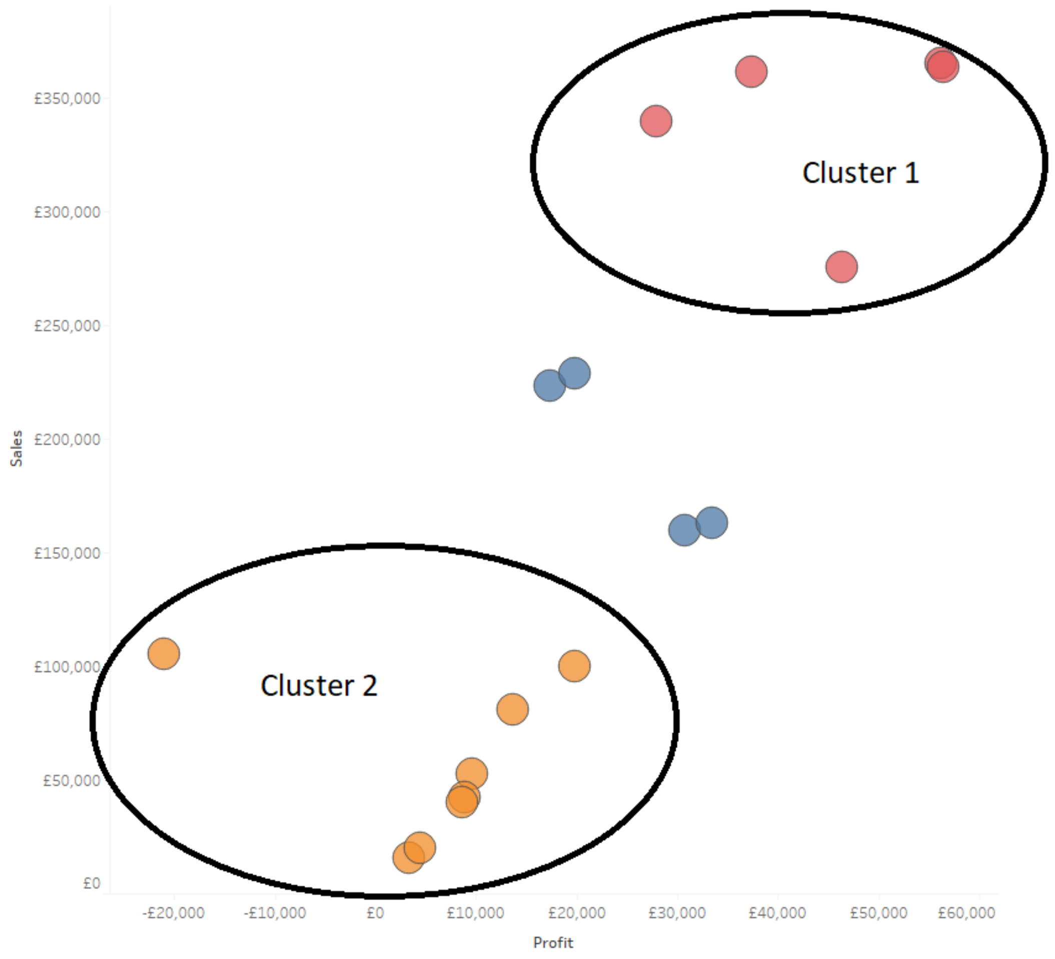
Basic concept of cluster analysis using two variables

Cluster 1:

High Profit and High Sales

Cluster 2:

Low Profit and Low Sales



Why Clustering?

- The goal of clustering is to determine the intrinsic grouping in a set of unlabelled data
- Organize the data into clusters to show internal structure of the data
- Sometimes partition is the goal

Group 1

Group 2

Group 3

Group 4

Types of Clustering

Clustering

Exclusive clustering

Here, an item belongs exclusively to one cluster, not various

K means does this type of clustering

Overlapping clustering

Here, an item belongs to multiple clusters

C means does this type of clustering

Hierarchical clustering

When two clusters have parent child relation

Hierarchical does this type of clustering

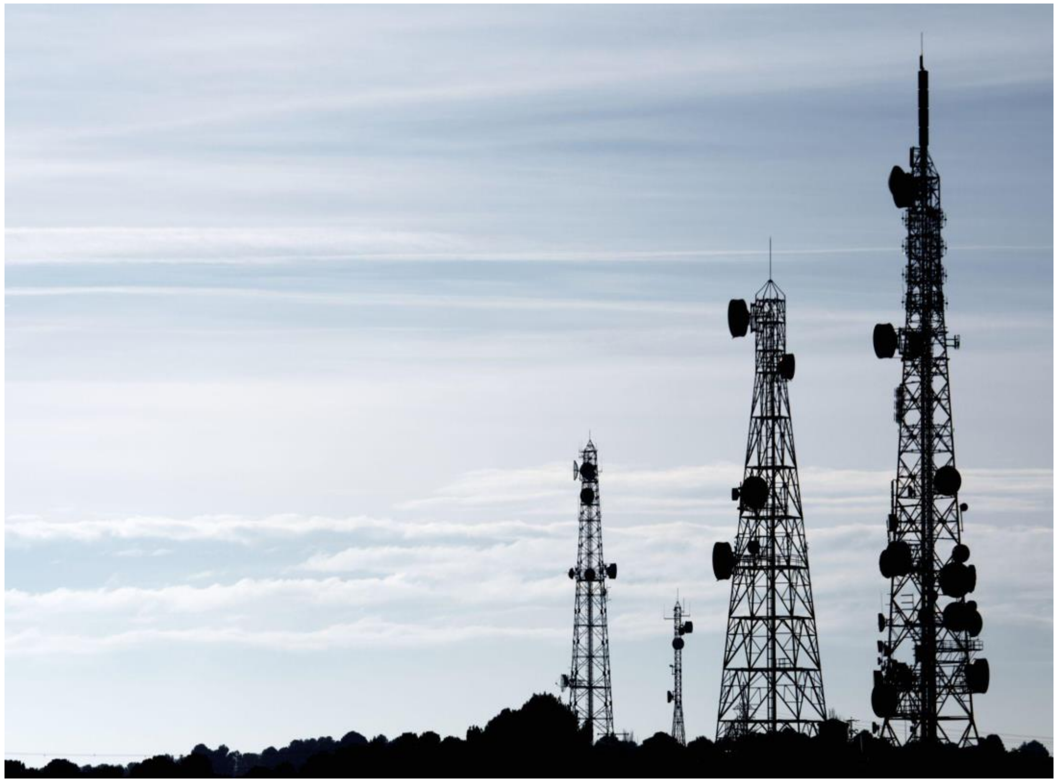
Business Problem

Telecom provider:

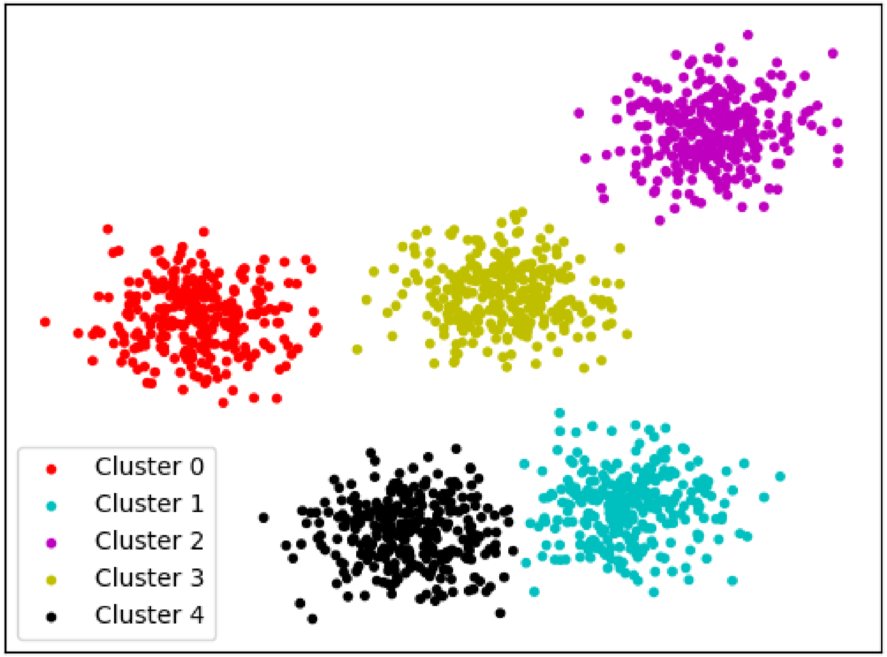
How can you cluster pre-paid customers to identify patterns in terms of money spent in recharging, sending SMS, and browsing the internet?

Pain points and Actions

- Use unused data to get the relevant information
- Segment customers by their buying pattern
- Use clustering algorithms to create different customer segments.
- The classification would help the company target specific clusters of customers for specific campaigns.



What is Clustering?

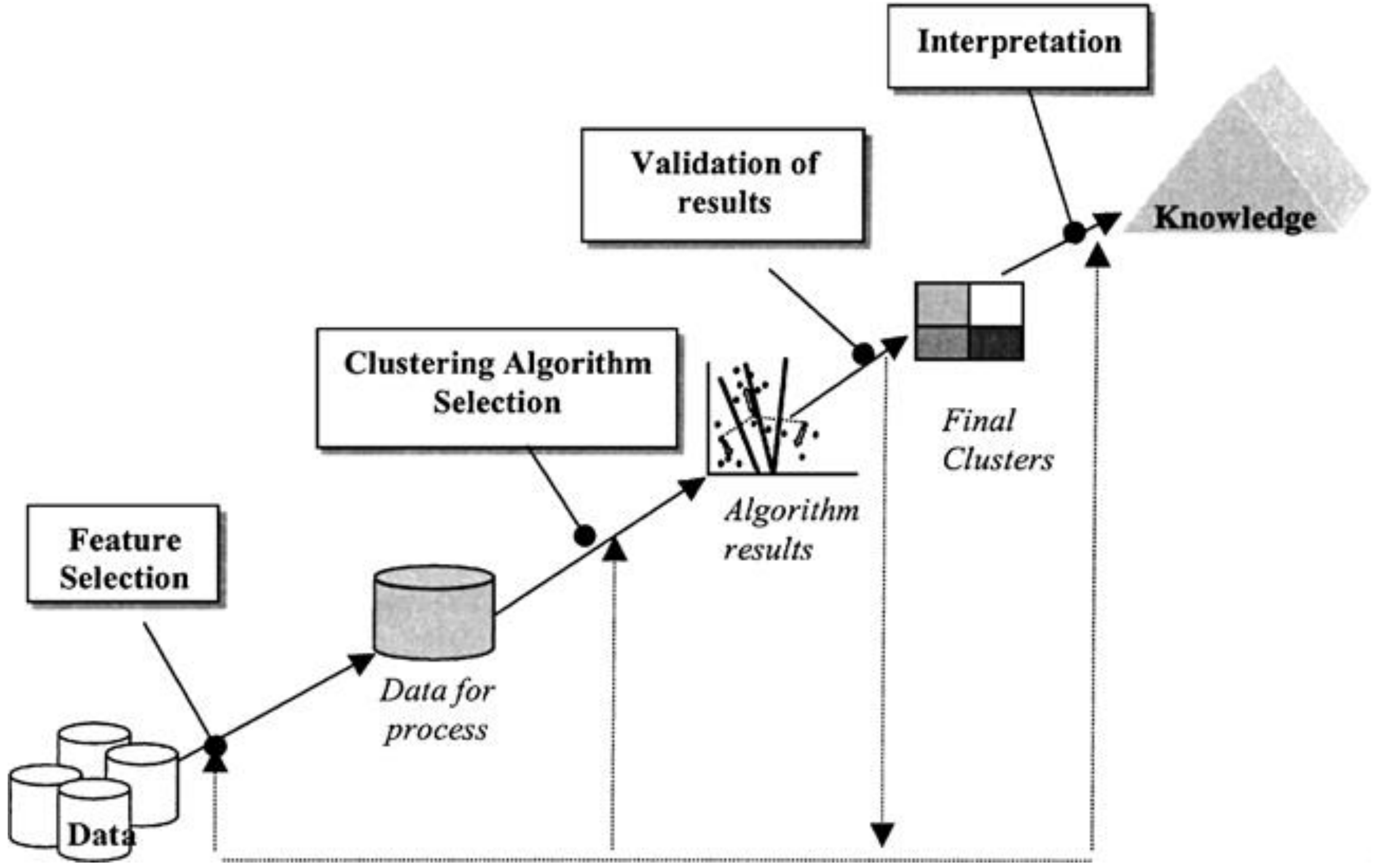


Clustering

- *Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups.*

- *Helps to segregate groups with similar traits and assign them into clusters.*
- *Get hidden insights*
- *Identify target customers for business campaigns.*

Phases of Clustering

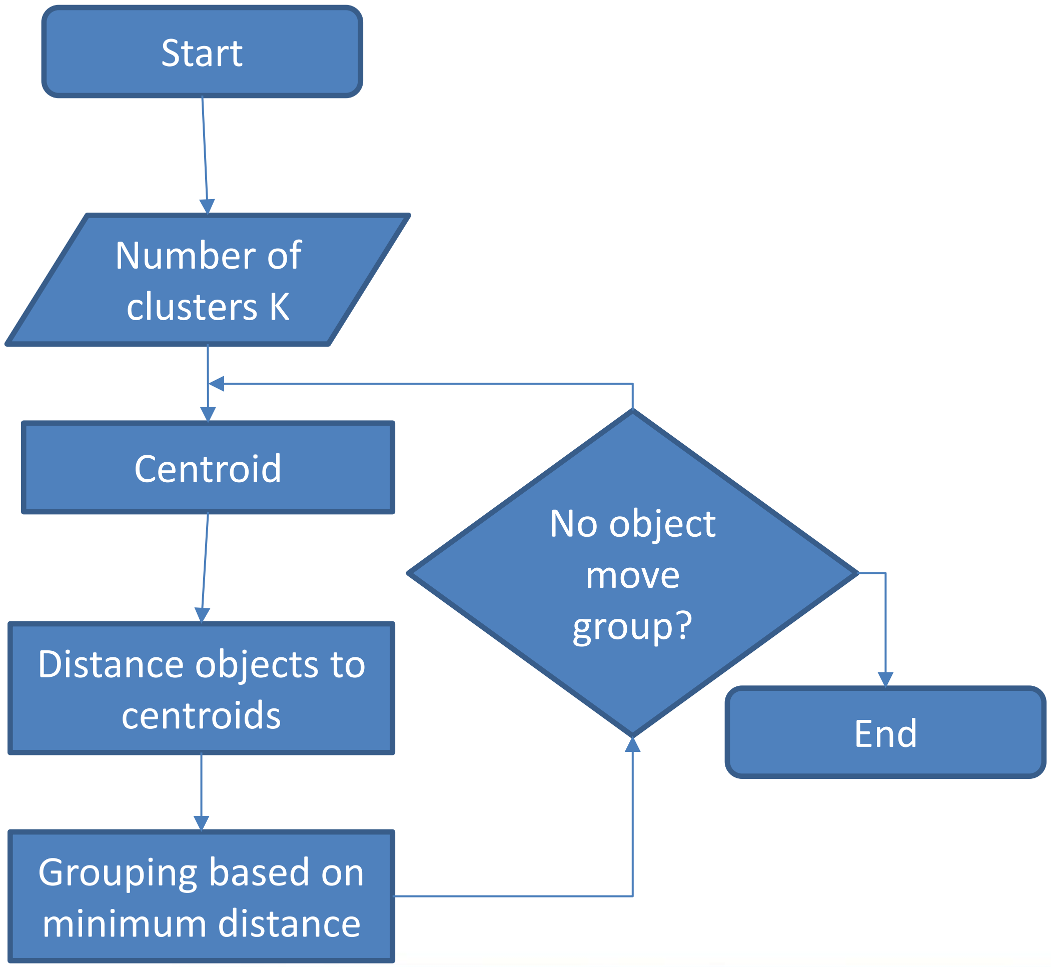


Types of Clustering Algorithms

K-means Clustering:

K-means is one of the simplest unsupervised learning algorithms that solves the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster.

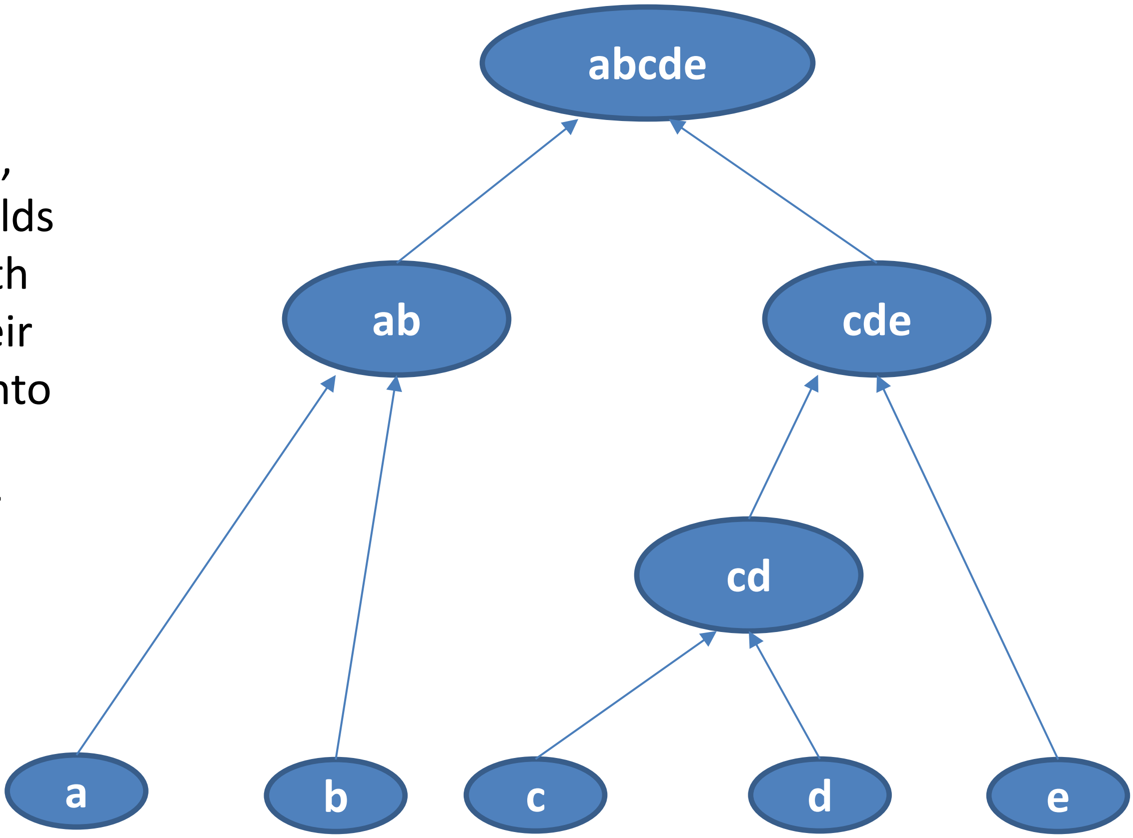
The number of clusters and the positions of cluster centroids are randomly chosen initially.


Start
Number of clusters K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
End

Types of Clustering Algorithms

Agglomerative Clustering:

Agglomerative is a bottom-up approach for hierarchical clustering. Hierarchical clustering, as the name suggests is an algorithm that builds hierarchy of clusters. This algorithm starts with all the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.



Distance Measures

There are a few ways to determine how close two clusters are:

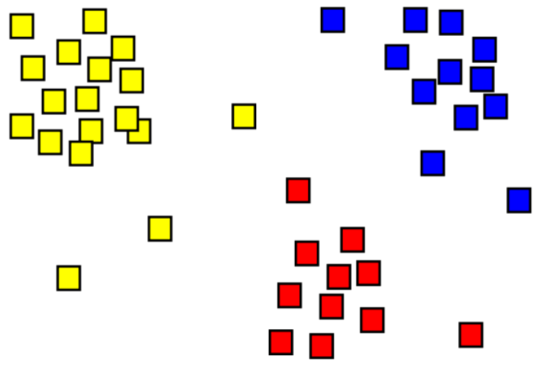
Complete linkage clustering: Find the maximum distance between points belonging to two different clusters.

Single linkage clustering: Find the minimum possible distance between the points belonging to two different clusters.

Mean linkage clustering: Find all possible pairwise distances for points belonging to two different clusters and then calculate the average.

Centroid linkage clustering: Find the centroid of each cluster and calculate the distance between centroids of two clusters.

K-means Clustering



- Input: K, set of data points $x_1 \ \ldots x_n$
- Place centroids $c_1 \ \ldots c_k$ at random locations.
- Repeat until convergence:

-- for each point x_i:

- find nearest centroid c_j, distance can be the Euclidian distance between instance x_i to cluster center c_j.
- assign the point x_i to cluster j.

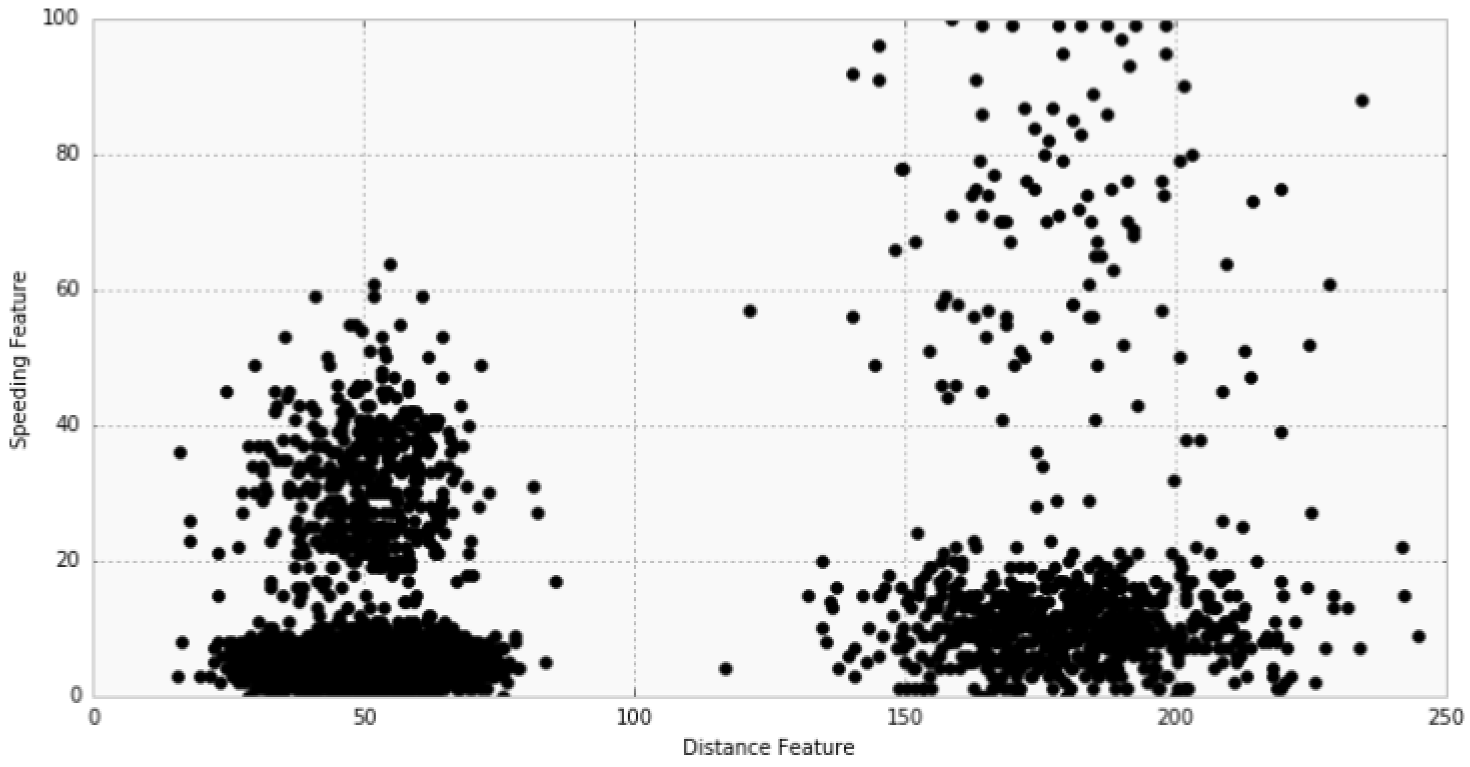
-- for each cluster j = 1 … K:

- new centroid c_j = mean of all points x_i assigned to cluster j
  in previous step.

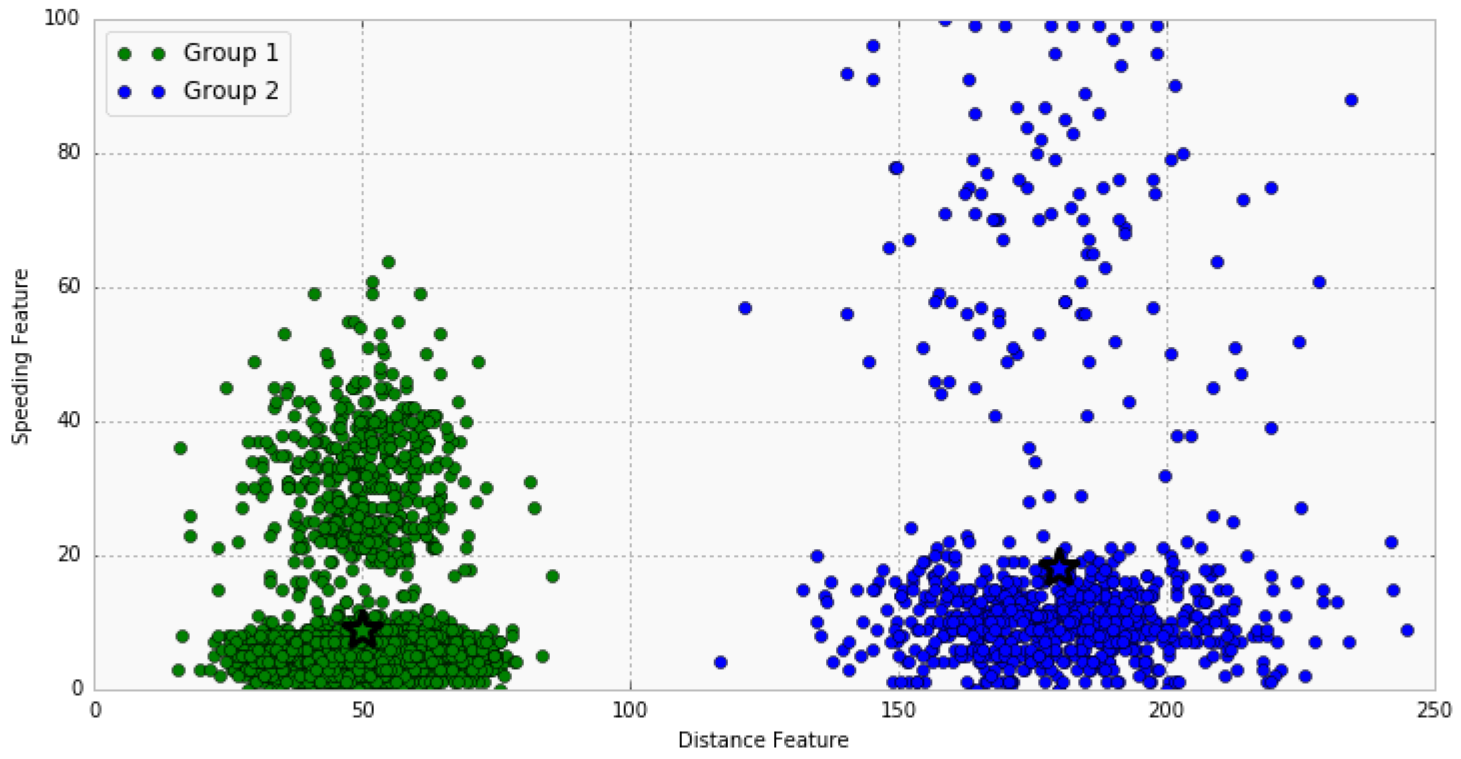
$$c_j(a) = \frac{1}{n_j} \sum_{x_i \to c_j} x_i(a) \qquad for\ a = 1\ \ldots\ d$$

- Stop when none of the cluster assignments change.

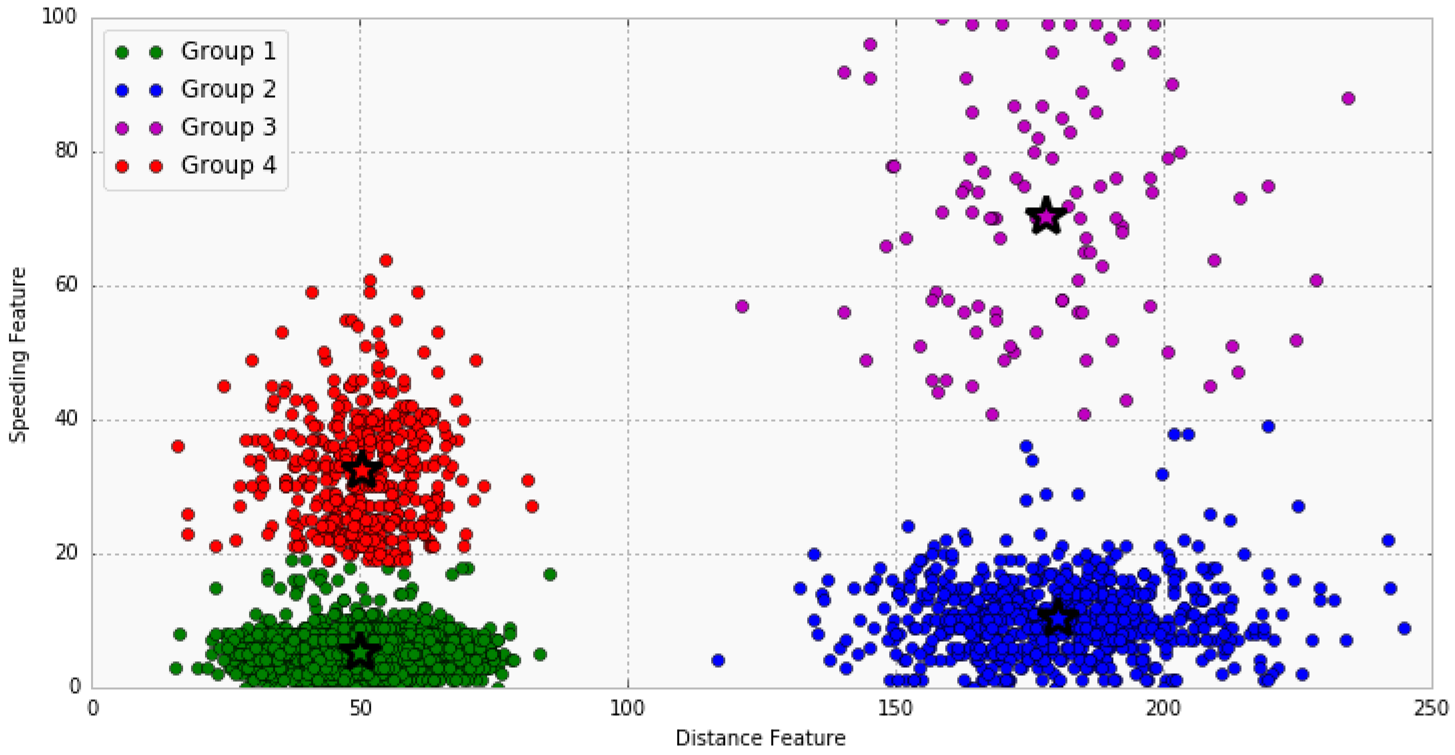
Clustering Example: Delivery fleet driver data



Two clusters created with initial centroids



Result after several iterations

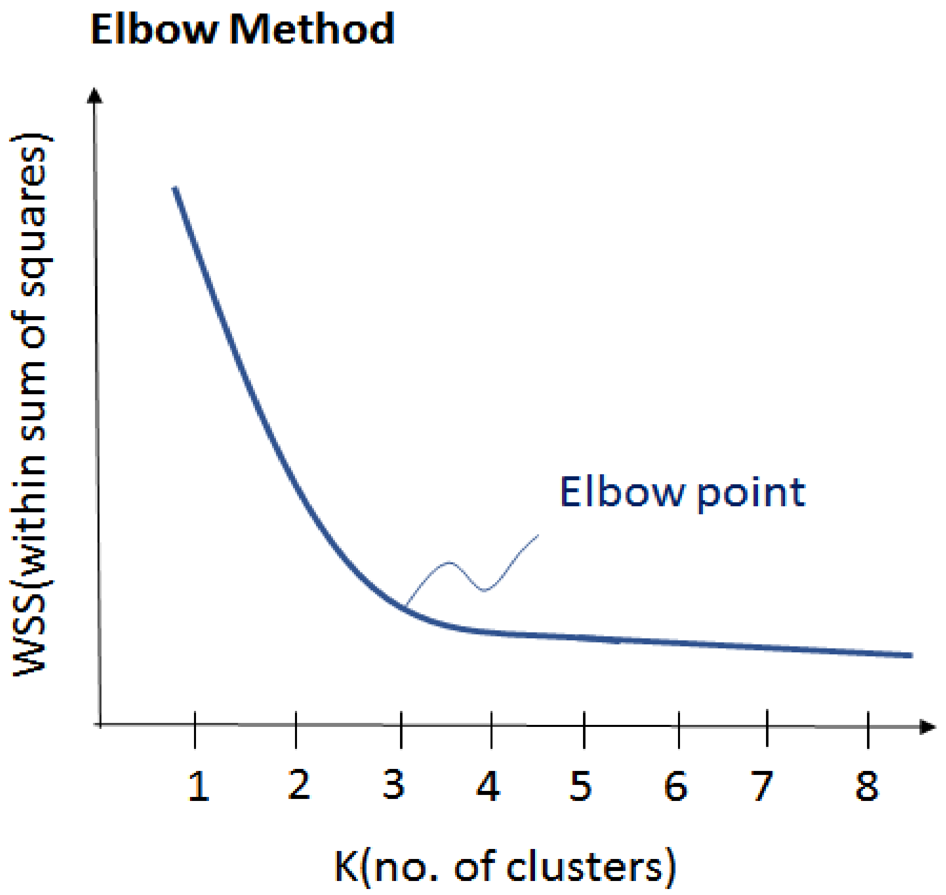


Elbow method: Right number of clusters

- The basic idea behind partitioning methods, such as k-means clustering, is to define clusters such that the total intra-cluster variation [or total within-cluster sum of square (WSS)] is minimized.
- The total WSS measures the compactness of the clustering and we want it to be as small as possible.
- Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.
- For each k, calculate the total within-cluster sum of square (wss).

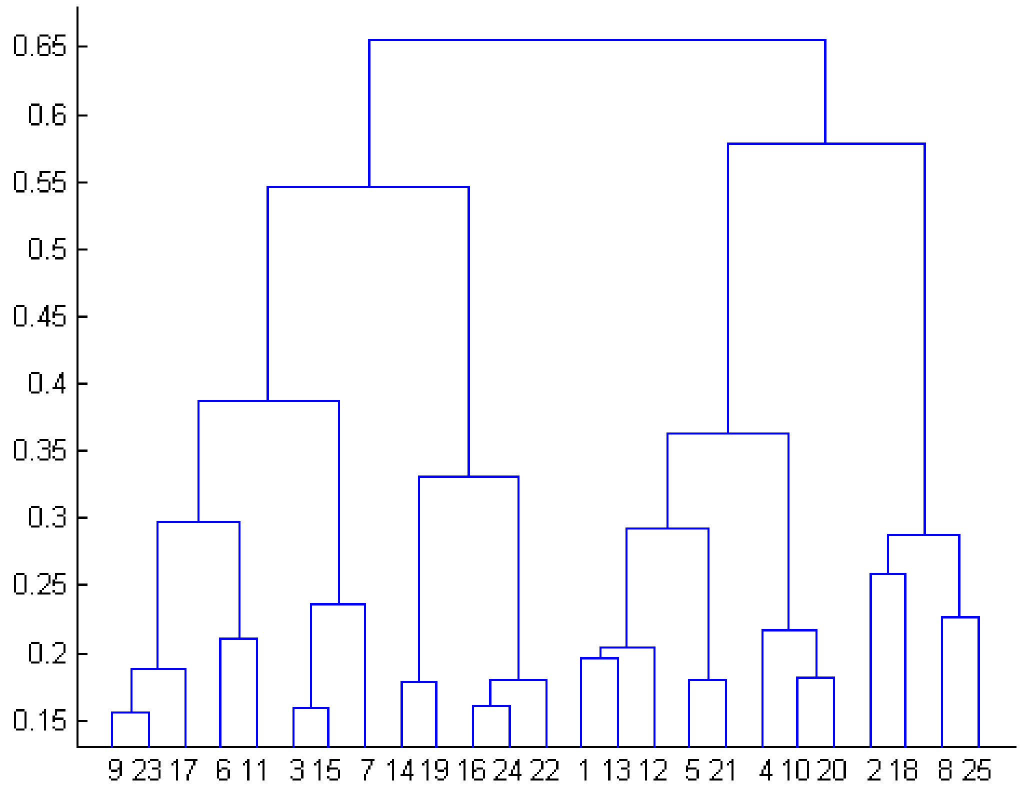
Elbow method: Right number of clusters

- Plot the curve of wss according to the number of clusters k.
- The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.



Hierarchical Agglomerative Clustering (HAC)

- Each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- The results of hierarchical clustering can be shown using dendrogram. The dendrogram can be interpreted as:
- At the bottom, we start with 25 data points, each assigned to separate clusters. Two closest clusters are then merged till we have just one cluster at the top. The height in the dendrogram at which two clusters are merged represents the distance between two clusters in the data space.

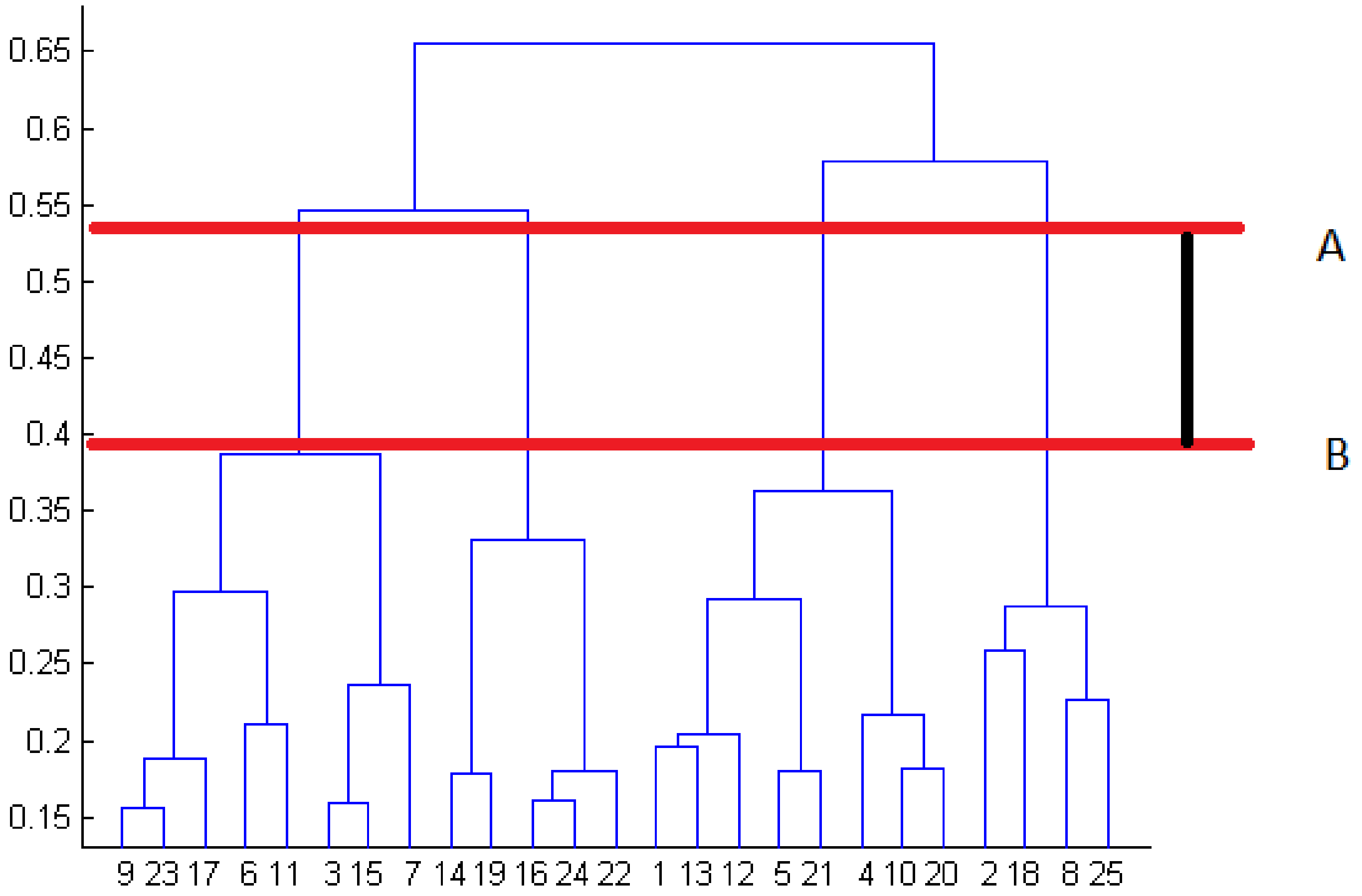


Source: Wikipedia

Right number of clusters

- The decision of the no. of clusters that can best depict different groups can be chosen by observing the dendrogram.
- The best choice of the no. of clusters is the no. of vertical lines in the dendrogram cut by a horizontal line that can transverse the maximum distance vertically without intersecting a cluster.
- In the above example, the best choice of no. of clusters will be 4 as the red horizontal line in the dendrogram in the below slide covers maximum vertical distance AB.

Dendrogram


0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
9 23 17 6 11 3 15 7 14 19 16 24 22 1 13 12 5 21 4 10 20 2 18 8 25
A
B

K means clustering - Code

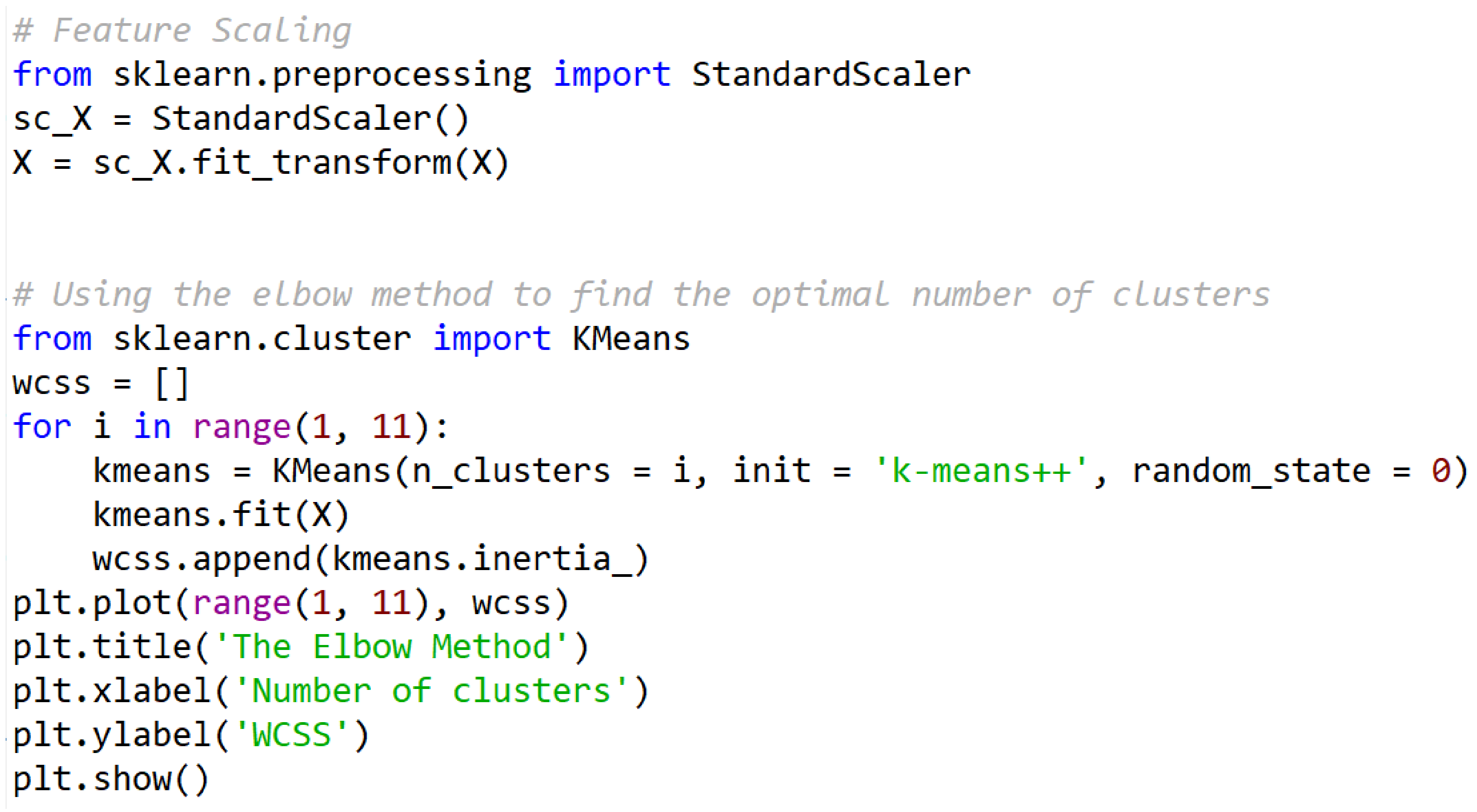
```python
# K-Means Clustering

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

import os
os.chdir('Give Path')


# Importing the dataset
dataset = pd.read_excel('Cluster_Problem_Dataset.xlsx')
dataset=dataset.dropna(axis=0)
X = dataset.iloc[:, 0:2].values
```

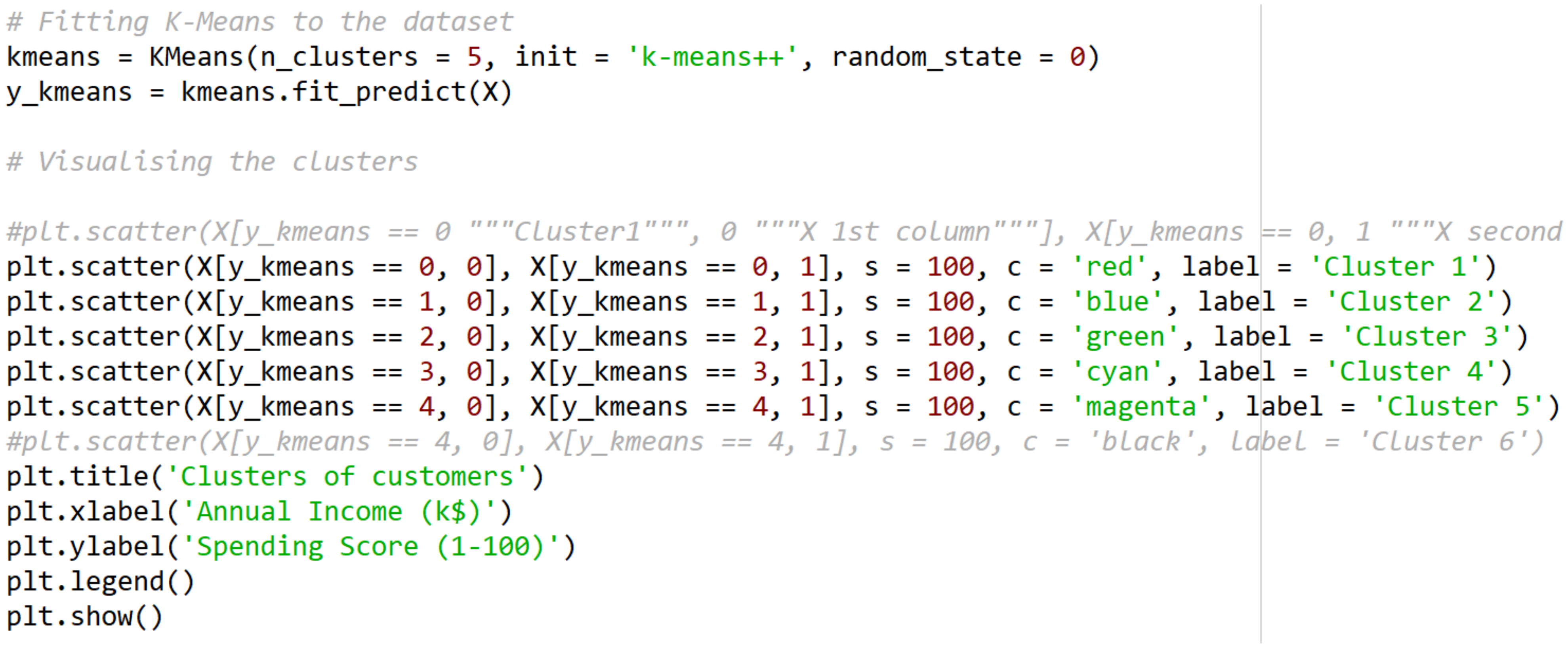
K means clustering - Code



```python
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X = sc_X.fit_transform(X)


# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

K means clustering - Code



```
# Fitting K-Means to the dataset
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 0)
y_kmeans = kmeans.fit_predict(X)

# Visualising the clusters

#plt.scatter(X[y_kmeans == 0 """Cluster1""", 0 """X 1st column"""], X[y_kmeans == 0, 1 """X second
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
#plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'black', label = 'Cluster 6')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

Hierarchical clustering - Code

```python
# Hierarchical clustering

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

import os
os.chdir('Give Path')

# Importing the dataset
dataset = pd.read_excel('Cluster_Problem_Dataset.xlsx')
dataset=dataset.dropna(axis=0)
X = dataset.iloc[:, 0:2].values
# y = dataset.iloc[:, 3].values
```

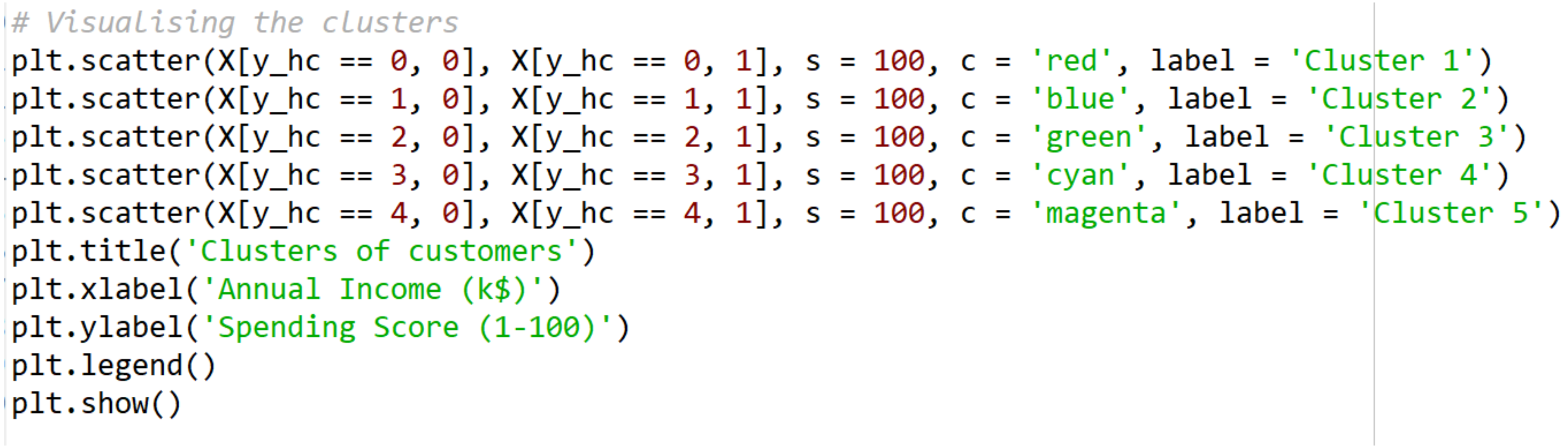
Hierarchical clustering - Code

```python
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X = sc_X.fit_transform(X)


# Using the dendrogram to find the optimal number of clusters
import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
plt.title('Dendrogram')
plt.xlabel('Customers')
plt.ylabel('Euclidean distances')
plt.show()

# Fitting Hierarchical Clustering to the dataset
from sklearn.cluster import AgglomerativeClustering
hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidean', linkage = 'ward')
y_hc = hc.fit_predict(X)
```

Hierarchical clustering - Code



```
# Visualising the clusters
plt.scatter(X[y_hc == 0, 0], X[y_hc == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_hc == 1, 0], X[y_hc == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_hc == 2, 0], X[y_hc == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_hc == 3, 0], X[y_hc == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_hc == 4, 0], X[y_hc == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

Cognixia
A Collabera LEARNING SOLUTIONS COMPANY

THANK YOU