Class 7 Unsupervised Learning and K-Means Clustering

Dr. Wei Miao

UCL School of Management

October 22, 2025

Section 1

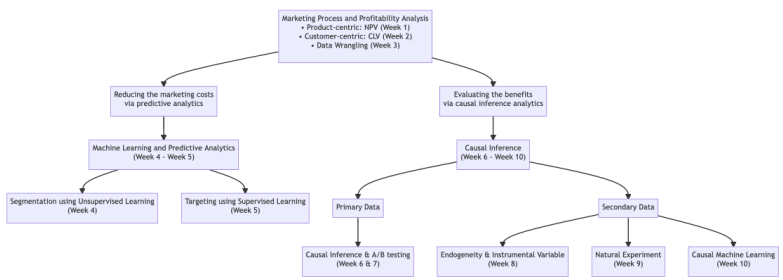
Overview of Predictive Analytics

Learning Objectives for Today

- Understand the concept of unsupervised learning
- Understand how to apply K-means clustering and determine the optimal number of clusters
- How to apply clustering analysis to customer segmentation for M&S

Roadmap for Predictive Analytics

- In Weeks 4 and 5, we will learn how to utilise **machine learning and predictive analytics** to improve ROI for M&S.
- We will learn two types of predictive analytics models:
  - Unsupervised learning (Week 4): K-means clustering for customer segmentation, and then target the most responsive customer segments
  - Supervised learning (Week 5): Decision trees and Random Forest for individualised customer targeting



Types of Predictive Analytics

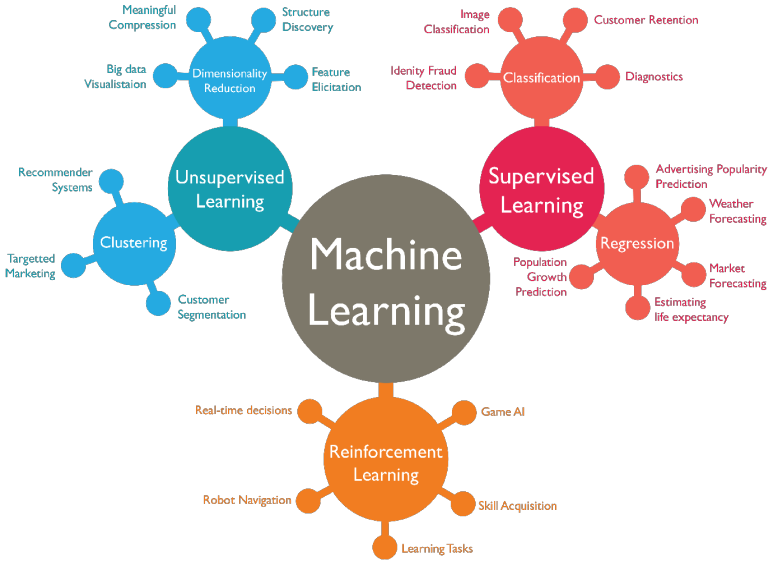
Definition

- **Features/Predictors (X)**: customer characteristics (age, income, spending, etc.)
- **Target/Response/Outcome (Y)**: dependent variable that we want to predict (e.g., whether a customer responds to a marketing campaign)

- Unsupervised Learning
  - Only observe X $=>$ want to uncover unknown subgroups
- Supervised Learning
  - Observe both X and Y $=>$ want to predict Y for new data

In Term 2, you will learn predictive analytics models systematically (MSIN0097 Predictive Analytics). By then, think about how those techniques can be applied to these case studies.

Types of Predictive Analytics



(Reinforcement learning is beyond the scope of this course, but at the high level, it involves learning optimal actions through trial and error to maximize cumulative rewards.)

Section 2

K-Means Clustering

K-Means Clustering

- K-means clustering is one of the most commonly used unsupervised machine learning algorithms for partitioning a given data set into a set of k groups (i.e. k clusters), where k represents the number of groups **pre-specified** by the analyst.
- For data scientists, it can classify customers into multiple segments (i.e., clusters), such that customers within the same cluster are as **similar** as possible, whereas customers from different clusters are as **dissimilar** as possible.
- Input: (1) customer characteristics; (2) the number of clusters
- Output: cluster membership of each customer

Similarity and Dissimilarity

- The clustering of observations into groups requires computing the (dis)similarity between each pair of observations. The result of this computation is known as a dissimilarity or distance matrix.
- The choice of similarity measures is a critical step in clustering.
- The most common distance measures are the **Euclidean distance** (the default for K-means).

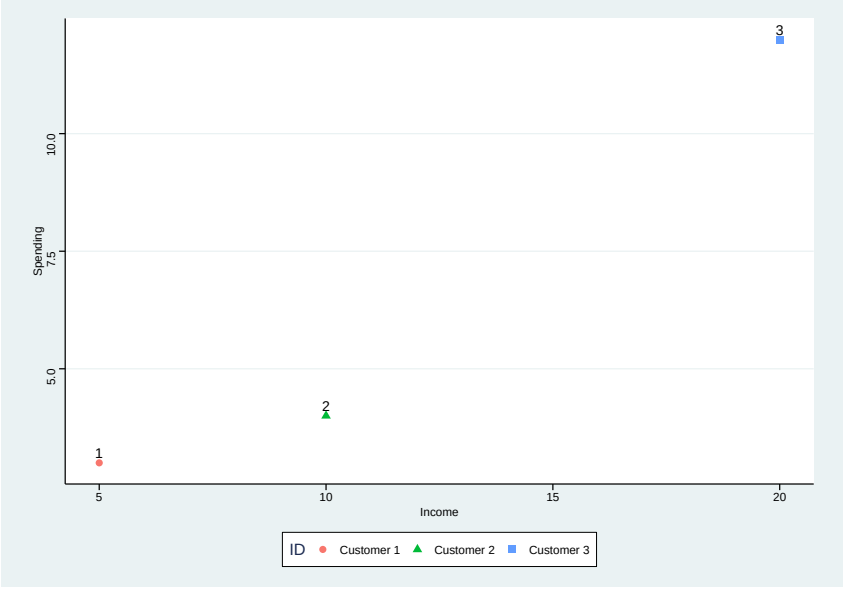
Euclidean Distance

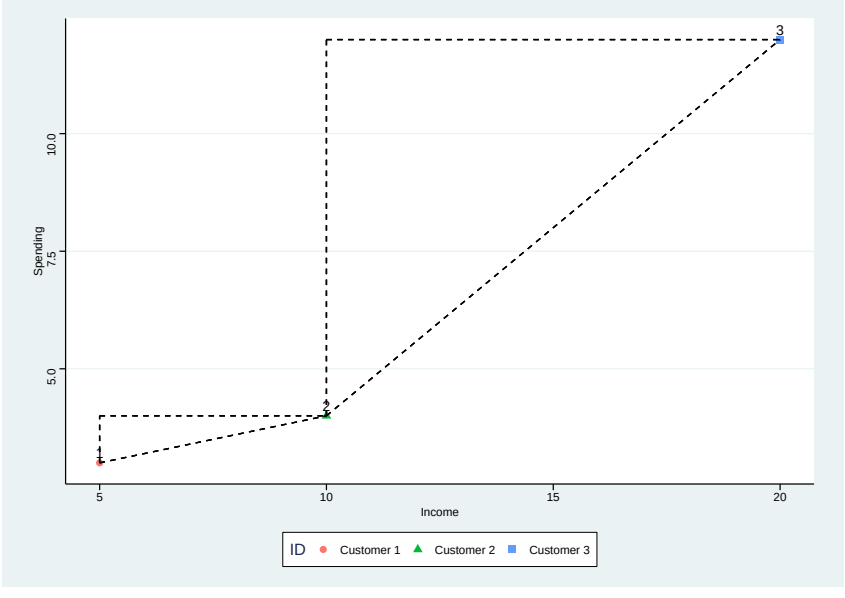
- The most common distance measure in machine learning is the Euclidean distance. In the following equation, we define the Euclidean distance between two customers (x) and (y) in an n-dimensional space (i.e., n different characteristics) as:

$$d_{\text{euc}}(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

- Example of Income and Spending for 3 customers
  - $Customer1 =>$ `c("income"=5, "spending"=3)`
  - $Customer2 =>$ `c("income"=10, "spending"=4)`
  - $Customer3 =>$ `c("income"=20, "spending"=12)`
- Euclidean distance
  - $d_{\text{euc}}(2,1) = \sqrt{(10-5)^2 + (4-3)^2} = \sqrt{25+1} = \sqrt{26}$
  - $d_{\text{euc}}(2,3) = \sqrt{(10-20)^2 + (4-12)^2} = \sqrt{100+64} = \sqrt{164}$

Visualisation of Euclidean Distance



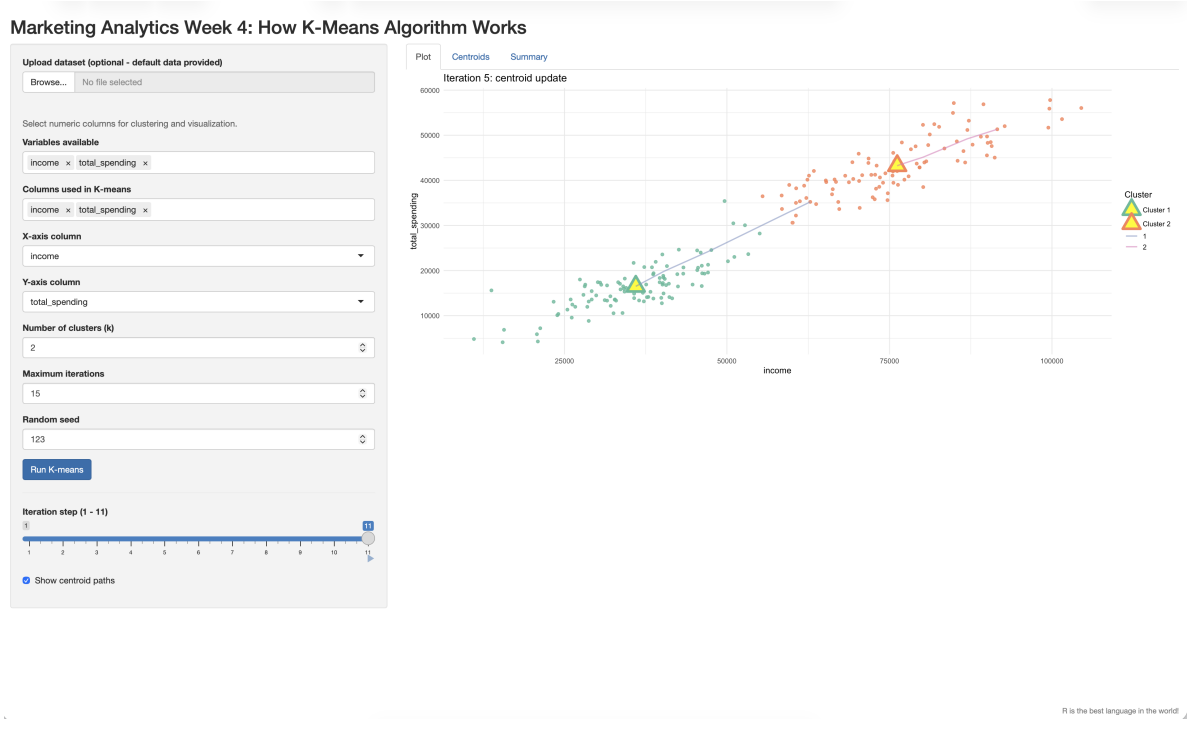


Section 3

Intuition of K-Means Algorithm

Interactive Illustration of K-Means Clustering

- Download the R Shiny app code from Moodle and let's run it locally.[1]



[1] If you are interested in developing your own Shiny apps, you can refer to the official R Shiny tutorial. It's really dope!

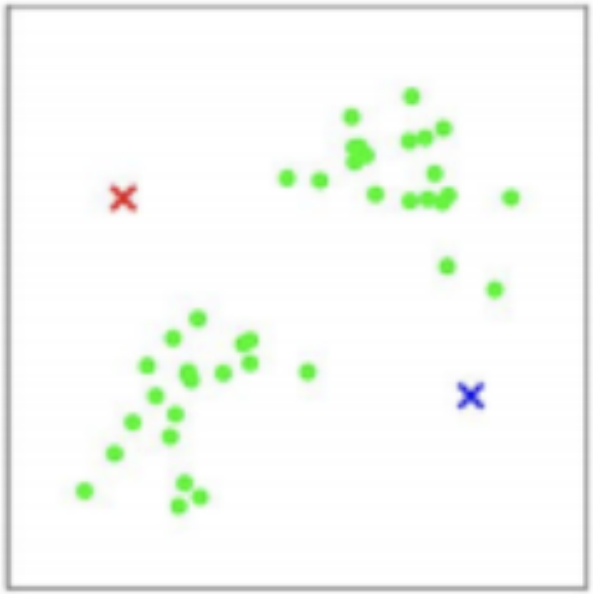
K-Means Clustering: Step 1



(a)

- Raw data points; each dot is a customer
- The X and Y axes are customer characteristics, for example, income and spending
- Obviously, there should be two segments
- Let's see how K-means uses a data-driven approach to classify customers into two segments

K-Means Clustering: Step 2



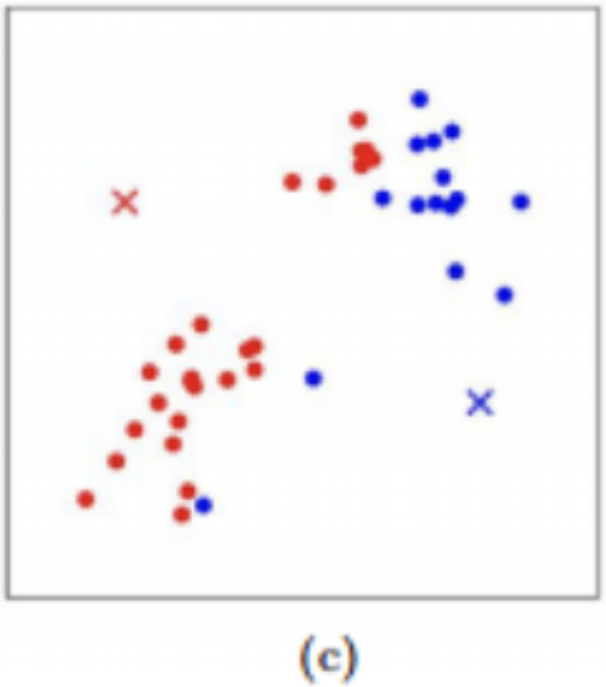
(b)

- We specify two segments
- K-means initialises the process by **randomly** selecting two centroids

Warning

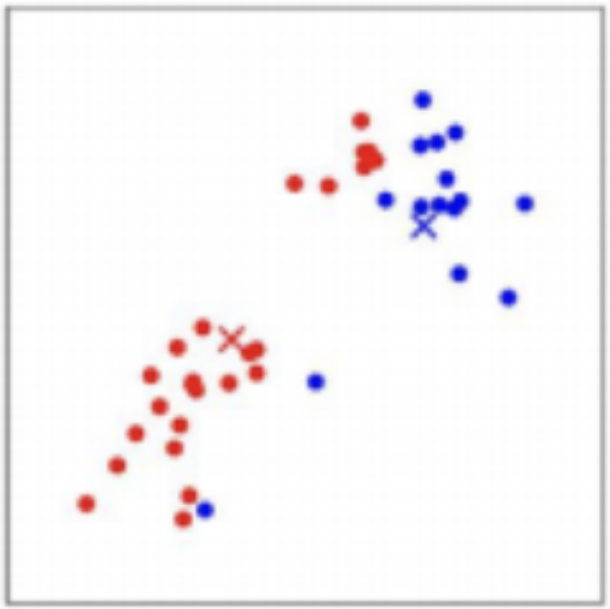
Due to this randomness, different starting points may yield varying results. We need to reinitialise the process repeatedly to ensure **robustness** of the results. That's why we set `nstart` to a value greater than 1 when applying K-means in R.

K-Means Clustering: Step 3



- K-means computes the distance of each customer to the red and blue centroids
- K-means assigns each customer to the red or blue segment based on which centroid is closer

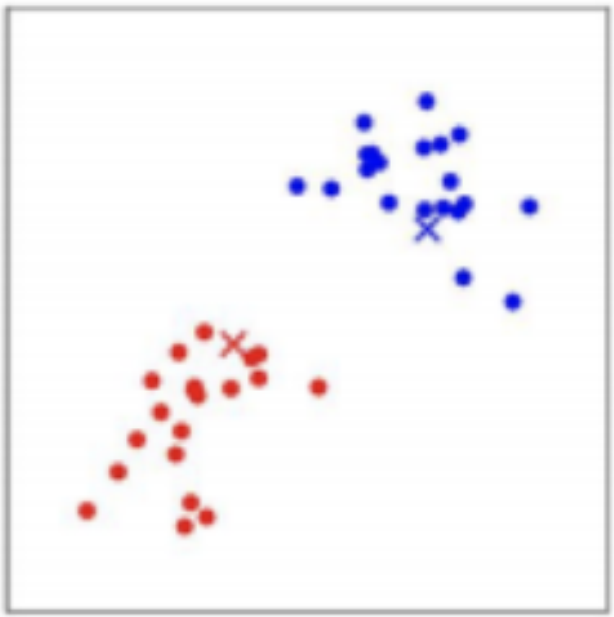
K-Means Clustering: Step 4



(d)

- K-means updates the centroids of each segment
- The red cross and blue cross in the picture are the new centroids
- We still see some outliers, so the algorithm continues

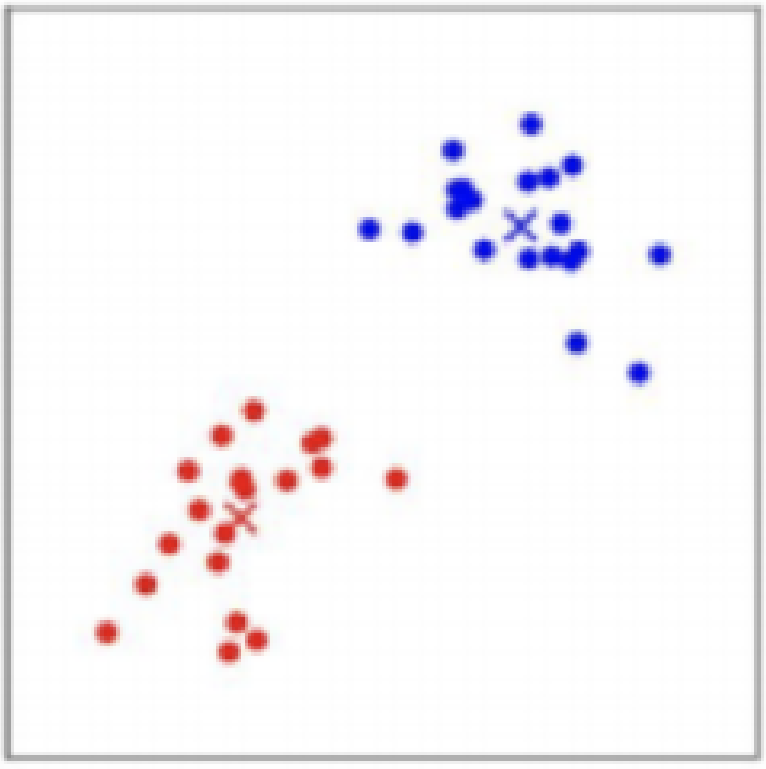
K-Means Clustering: Step 5



(e)

- K-means computes the distance of each customer to the red and blue centroids
- K-means assigns each customer to the red or blue segment based on which centroid is closer
- Now the outliers are correctly assigned to each segment

K-Means Clustering: Step 6



(f)

- K-means updates the centroids from the previous clustering
- K-means computes the distance of each customer to the new centroids
- K-means finds that all customers are correctly assigned to their nearest centroids, so the algorithm does not need to continue
- We say the algorithm has **converged**, and it stops

After-Class Readings

- More technical details: K-means Cluster Analysis