CS5805 : Machine Learning I
Lecture # 15

Reza Jafari

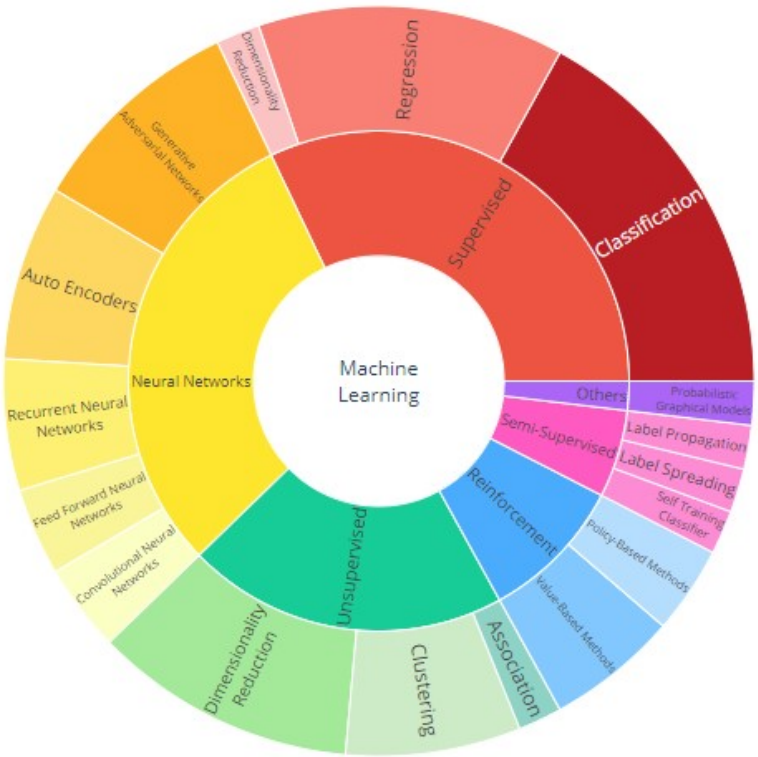
Associate Professor, Computer Science
Virginia Tech University

December 4, 2024

What is association rule mining?

- Association rule mining techniques determine if certain data points are more likely to occur together.
- A simple example would be the supermarket shopping basket analysis.
- If someone is buying ground beef, does it make them more likely to also buy spaghetti?
- Learning of Association rules is used to find relationships between attributes in large databases.
- Apriori algorithm is an unsupervised algorithm that can find the relationship between features.
- Apriori algorithm is given by R. Agrawal and R. Srikant in 1994 for finding frequent itemsets in a dataset for boolean association rule. Name of the algorithm is Apriori because it uses prior knowledge of frequent itemset properties. We apply an iterative approach or level-wise search where k-frequent itemsets are used to find k+1 itemsets.

Machine Learning algorithm classification



- Association rule mining (ARM) is a data mining technique that identifies relationships between items in large datasets.
- ARM is a type of unsupervised machine learning that uses association rules, which are if-then statements that describe the relationship between two or more items.

What is association rule mining?

- An association rule has 2 parts:

Antecedent (if)

- An antecedent is something that's found in data.

Consequent (then)

- An consequent is an item that is found in combination with the antecedent.

If a customer buys bread, he's 70% likely of buying milk

- In the above association rule, **bread** is **antecedent** and **milk** is the **consequent**.

Association Mining Applications

Market Basket Analysis(MBA)

- MBA is a technique for identifying consumer patterns by mining associations from store transactional databases.
- The outcomes of this "market basket analysis" can then be utilised to suggest product pairings.
- Choosing which goods to place next to one another on store shelves might assist raise sales significantly.

Medical Diagnosis

- Using relational association rule mining, we can identify the probability of the occurrence of illness concerning various factors and symptoms.

Association Mining Applications

Census Data

- Association rule mining and data mining has immense potential in supporting sound public policy and bringing forth an efficient functioning of a democratic society.

Protein Sequence

- Proteins are sequences made up of twenty types of amino acids.
- Each protein bears a unique 3D structure which depends on the sequence of these amino acids.
- A slight change in the sequence can cause a change in structure which might change the functioning of the protein.
- This dependency of the protein functioning on its amino acid sequence has been a subject of great research and knowledge and understanding of these association rules will come in extremely helpful during the synthesis of artificial proteins.

Association Rule Learning

- Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases
- Simple supermarket shopping basket analysis explains how the association rules are found.
- Most machine learning algorithms work with numeric datasets and hence tend to be mathematical.
- However, association rule mining is suitable for non-numeric, categorical data and requires just a little bit more than simple counting.

	Item 1	Item 2	Item 3
Shopper 1	Eggs	Bacon	Soup
Shopper 2	Eggs	Bacon	Apple
Shopper 3	Eggs	Bacon	Apple
Shopper 4	Soup	Bacon	Banana
Shopper 5	Banana	Butter	-
Shopper 6	Butter	-	-

Preliminaries

- Association rule mining is a procedure which aims to observe frequently occurring patterns, correlations, or associations from datasets found in various kinds of databases such as relational databases, transactional databases, and other forms of repositories.
- Binary Representation : Market basket data can be represented in a binary format.
- An item can be treated as a binary variable whose value is one if the item is present in a transaction and zero otherwise.
- This is a simplistic view of real market basket since important aspects of the data such as quantity or price are ignored.

TID	Eggs	Bacon	Apple	Soup	Banana	Butter
1	1	1	0	1	0	0
2	1	1	1	0	0	0
3	1	1	1	0	0	0
4	0	1	0	1	1	0
5	0	0	0	0	1	1
6	0	0	0	0	0	1

Terminology

- Itemset : Let

$$I = \{i_1, i_2, ..., i_d\}$$

be the set of d attributes and all items in a market basket data and

$$T = \{t_1, t_2, ..., t_N\}$$

be the set of all transactions. Each transaction t_i contains a subset of items chosen from I. A collection of zero or more items is called itemset.

- If an itemset contains k items, it is called k-itemset.
- For example {Eggs, Bacon, Apple} is an example of a 3-itemset.
- A transaction t_j is said to contain an itemset X if X is subset of t_j.
- For example the second transaction in the table contains the itemset {Eggs,Bacon} but not {Eggs,Soup}

Terminology

Support count

- Refers to number of transactions that contain a particular itemset.

$$\sigma(X) = |\{t_i | X \subseteq t_i, t_i \in T\}|$$

where $|.|$ denotes the cardinality of the set.

- For example the support count for {Eggs, Bacon, Apple} is 2 because there are only two transactions that contain all three items.

Support

- The frequency bought items: $s(X) = \frac{\sigma(X)}{N}$
- An itemset is called frequent if $s(X)$ is greater than some user-defined threshold $\boxed{minsup}$.

Example

- s(eggs) $= 3/6 = 0.5$
- s(Bacon) $= 4/6 = 0.667$
- s(Apple) $= 2/6 = 0.33$
- s(Soup) $= 2/6 = 0.33$
- s(Butter) $= 1/6 = 0.166$
- s(Banana) $= 2/6 = 0.333$
- For the sake of our example, let's set **minimum support to 0.5**, which leaves us to work with Eggs and Bacon for the rest of this example. (Most frequent)

	Item 1	Item 2	Item 3
Shopper 1	Eggs	Bacon	Soup
Shopper 2	Eggs	Bacon	Apple
Shopper 3	Eggs	Bacon	Apple
Shopper 4	Soup	Bacon	Banana
Shopper 5	Banana	Butter	-
Shopper 6	Butter	-	-

Terminology

Confidence

- This will tell us how confident (based on our data) we can be that an item will be purchased, given that another item has been purchased.

$$c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

- Support (σ) determines how often a rule is applicable to a given dataset.
- Confidence (c) determines how frequently items in Y appear in transactions that contain X.

Terminology

- Confidence(Eggs $\rightarrow$ Bacon) $= \frac{P(Eggs\&Bacon)}{s(Eggs)} = \frac{3/6}{3/6} = 1$
- Confidence(Bacon $\rightarrow$ Eggs) $= \frac{P(Eggs\&Bacon)}{s(Bacon)} = \frac{3/6}{4/6} = 0.75$
- The above tells us that whenever eggs are bought, bacon is also bought 100% of the time. Also, whenever bacon is bought, eggs are bought 75% of the time.

Lift

- Given that different items are bought at different frequencies, how do we know that eggs and bacon really do have a strong association, and how do we measure it?

$$Lift(X \rightarrow Y) = \frac{confidenace(X \rightarrow Y)}{support(Y)}$$

Example

- Lift(Eggs$\rightarrow$ Bacon) $= \frac{3/6}{3/6 \times 4/6} = 1.5$
- Lift(Bacon$\rightarrow$ Eggs) $= \frac{3/6}{3/6 \times 4/6} = 1.5$
- $Lift > 1$ means that the two items are more likely to be bought together.
- $Lift < 1$ means that the two items are more likely to be bought separately.
- $Lift = 1$ means that the two items are no association between two items.
- Lift is simply a measure that tells us whether the probability of buying eggs increases or decreases given the purchase of bacon.
- Since the probability of buying eggs in such a scenario goes up from 0.5 to 0.75, we see a positive lift of 1.5 times (0.75/0.5=1.5). This means you are 1.5 times (i.e., 50%) more likely to buy eggs if you have already put bacon into your basket.

Example

Conviction

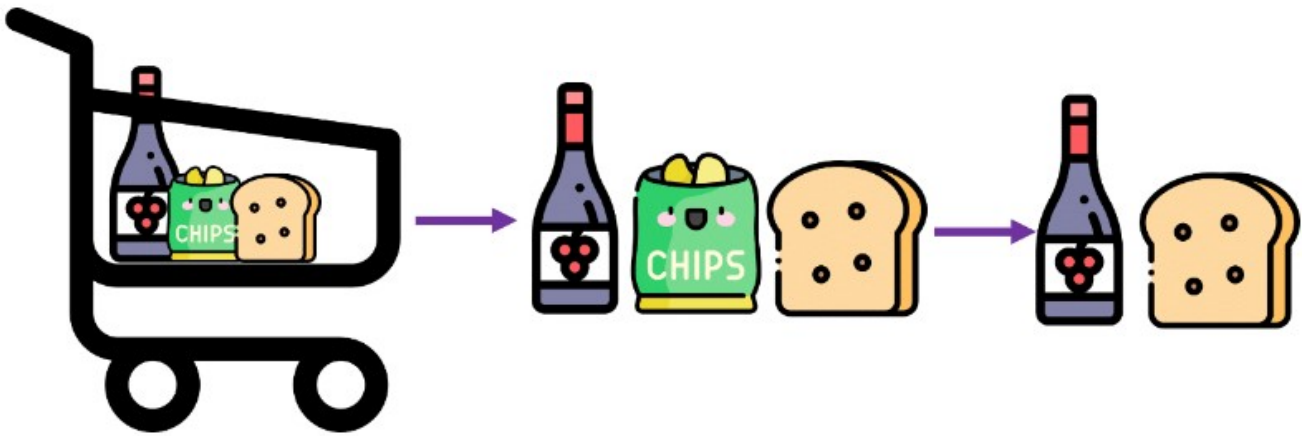
- Conviction is another way of measuring association.
- It compares the probability that X appears without Y if they were independent with the actual frequency of the appearance of X without Y.

$$Conviction(X \rightarrow Y) = \frac{1 - s(Y)}{1 - c(X \rightarrow Y)}$$

- conviction (Eggs$\rightarrow$ Bacon) $= \frac{1-s(Bacon)}{1-c(Eggs \rightarrow Bacon)} = \frac{1-2/3}{1-1} = \infty$
- conviction (Bacon$\rightarrow$ Eggs) $= \frac{1-s(Eggs)}{1-c(Bacon \rightarrow Eggs)} = \frac{1-1/2}{1-3/4} = 2$
- conviction(Eggs$\rightarrow$Bacon)$=\infty$ because no single instance of eggs being bought without bacon (confidence=100%)
- Conviction=1 means that items are not associated, while conviction$>$ 1 indicates the relationship between the items (the higher the value, the stronger the relationship).

What is Apriori Algorithm?

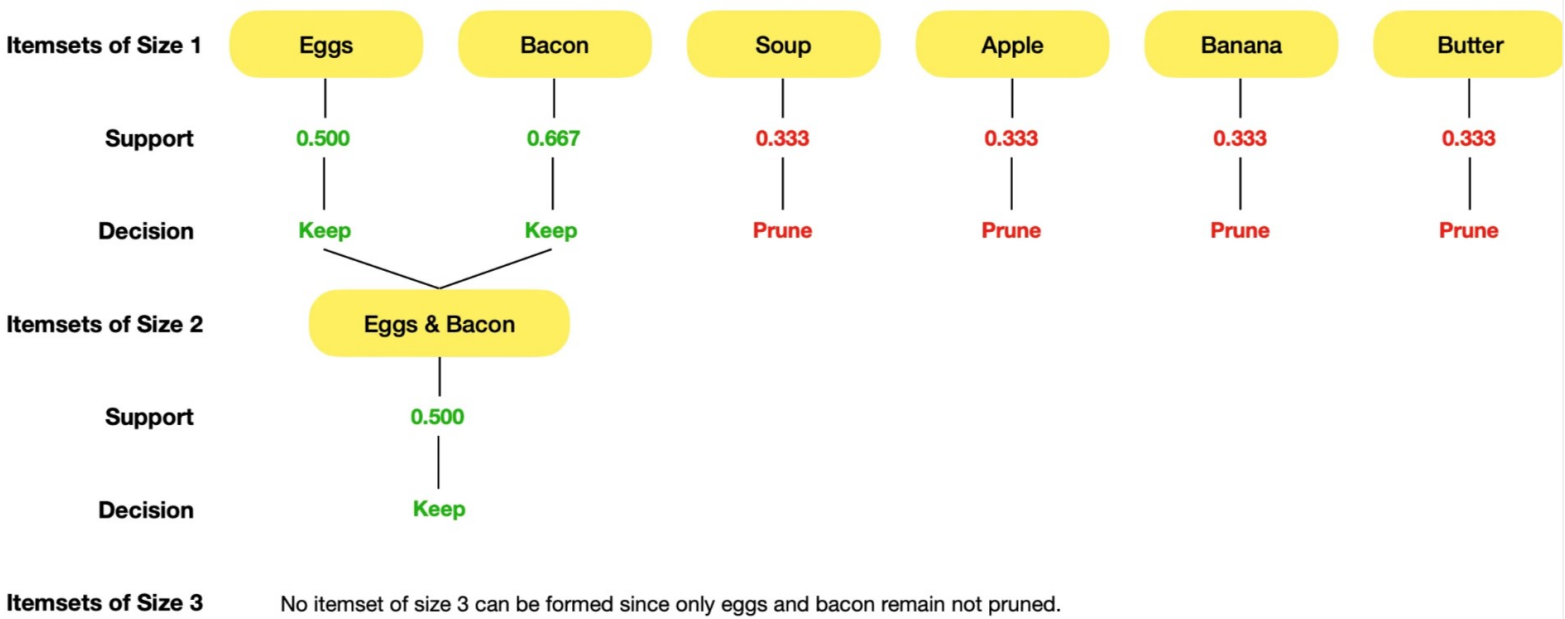
- Apriori algorithm is the algorithm that is used to find out the association rules between objects.
- The key concept in the **Apriori algorithm** is that it assumes all subsets of a frequent itemset to be frequent. Similarly, for any infrequent itemset, all its supersets must also be infrequent.
- In simple words, the apriori algorithm is an association rule learning that analyzes that People who bought item X also bought item Y.
- Apriori Algorithm is also known as **frequent pattern mining**.



Apriori Algorithm

Apriori Algorithm steps

- Calculate support for itemsets of size 1
- Apply the minimum support threshold and prune itemsets that do not meet the threshold.
- Move on to itemsets of size 2 and repeat steps one and two.
- Continue the same process until no additional itemsets satisfying the minimum threshold can be found.



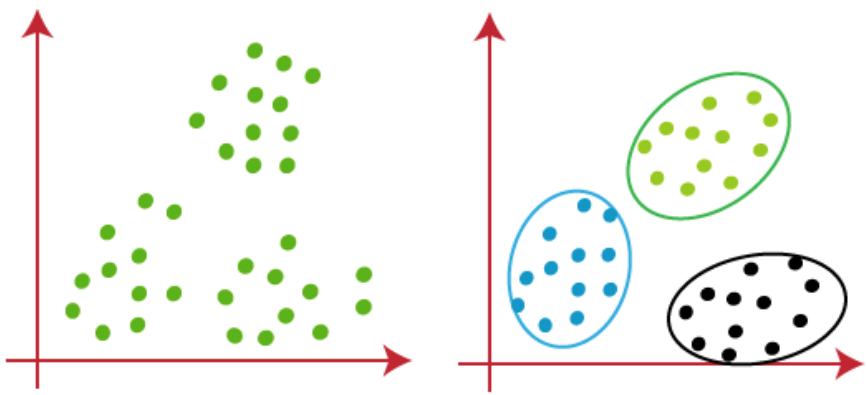
In class Assignment

- Suppose this is our dataset of any supermarket, where user id and items are as shown below. Using apriori algorithm find the strongest association rule between items.

User ID	Items
001	1, 3, 4
002	2, 3, 5
003	1, 2, 3, 5
004	2, 5

Clustering

- Clustering is a Machine Learning technique that involves the grouping of data points.
- Clustering is a method of unsupervised learning where the references need to be drawn from unlabelled datasets.
- Given a set of data points, we can use a clustering algorithm to classify each data point into a specific group.
- Data points in the same group should have similar properties and/or features, while data points in different groups should have highly dissimilar properties and/or features.



Applications of Cluster Analysis

- It is broadly used in many applications such as market research, pattern recognition, data analysis, and image processing.
- Clustering can also help marketers discover distinct groups in their customer base. And they can characterize their customer groups based on the purchasing patterns
- Clustering also helps in classifying documents on the web for information discovery.
- Clustering is also used in outlier detection applications such as detection of credit card fraud.
- It also helps in the identification of groups of houses in a city according to house type, value, and geographic location.

Requirements of Clustering in Data Mining

- Scalability: To deal with large databases.
- Ability to deal with different kinds of attributes: Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) data, categorical, and binary data.
- Discovery of clusters with attribute shape: The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small sizes.
- High dimensionality:he clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.
- Ability to deal with noisy data:Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.
- Interpretability: The clustering results should be interpretable, comprehensible, and usable.

K-Means clustering Description

- Given a set of observations $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ where each observations is a d-dimensional real vector, k-means clustering aims to partition the n observations in to $k(\leq n)$ sets $\mathbf{S} = \{S_1, S_2, ..., S_k\}$ so to minimize the within-cluster sum of squares:

$$\arg\min \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \mu_i\|^2$$

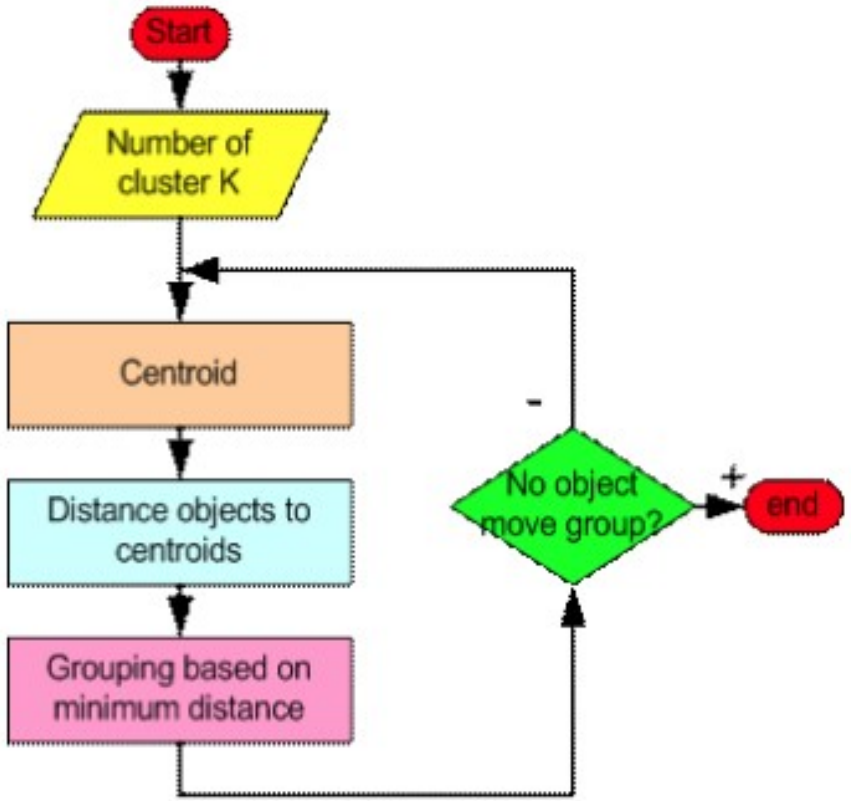
where μ_i is the mean centroid of points S_i:

$$\mu_i = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x}$$

$|S_i|$ is the size of S_i and $||.||$ is the usual L^2 norm.

K-Means clustering steps

- The K-means algorithm will do three steps below until convergence:
  1. Determine the centroid
  2. Determine the distance of each object to the centroid
  3. Group the object based on minimum distance



K-Means clustering

- K-means is a centroid-based clustering algorithm, where we calculate the distance between each data point and a centroid to assign it to a cluster.
- It is an iterative process of assigning each data point to the groups and slowly data points get clustered based on similar features.
- The objective is to minimize the sum of distances between the data points and the cluster centroid, to identify the correct group each data point should belong to.
- We divide a data space into K clusters and assign a mean value to each.

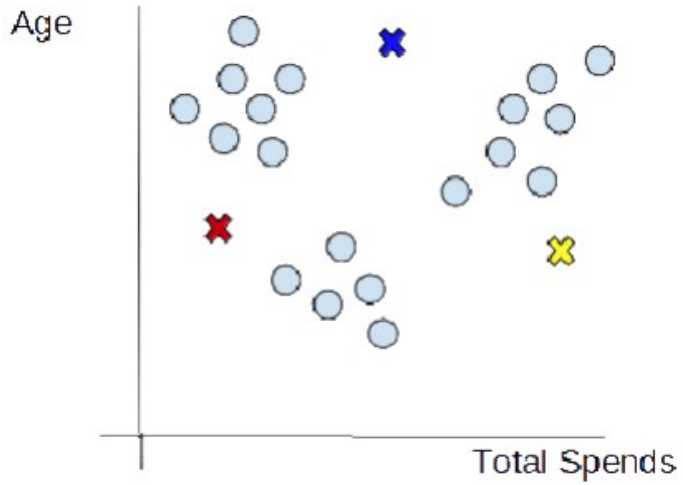
How does K-means work?

step 1 - Choosing the number of clusters

- Choosing the number of clusters. Let's select K=3.

step 2 - Initializing centroids

- Centroid is the center of a cluster but initially, the exact center of data points will be unknown so, we select random data points and define them as centroids for each cluster. We will initialize 3 centroids in the dataset

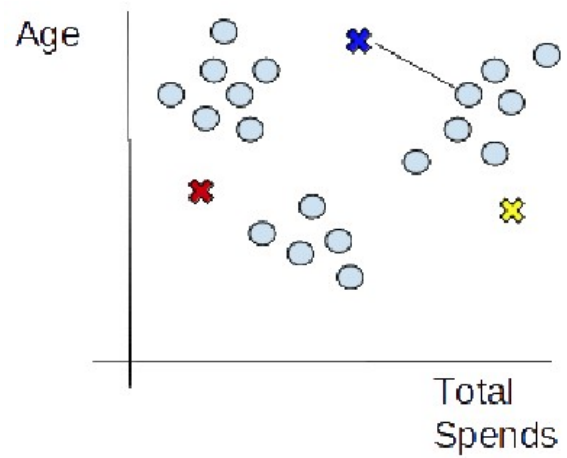


How does K-means work?

step 3 - Assign data points to the nearest cluster

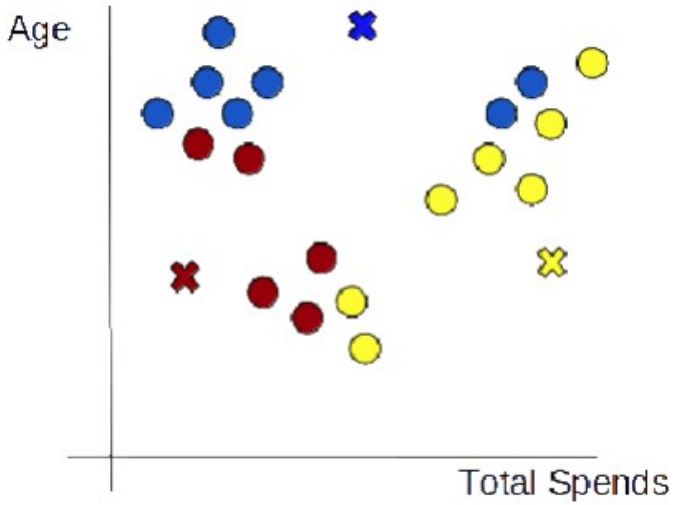
- Now that centroids are initialized, the next step is to assign data points X_n to their closest cluster centroid C_k
- we will first calculate the distance between data point X and centroid C using Euclidean Distance metric.

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



How does K-means work?

- And then choose the cluster for data points where the distance between the data point and the centroid is minimum.

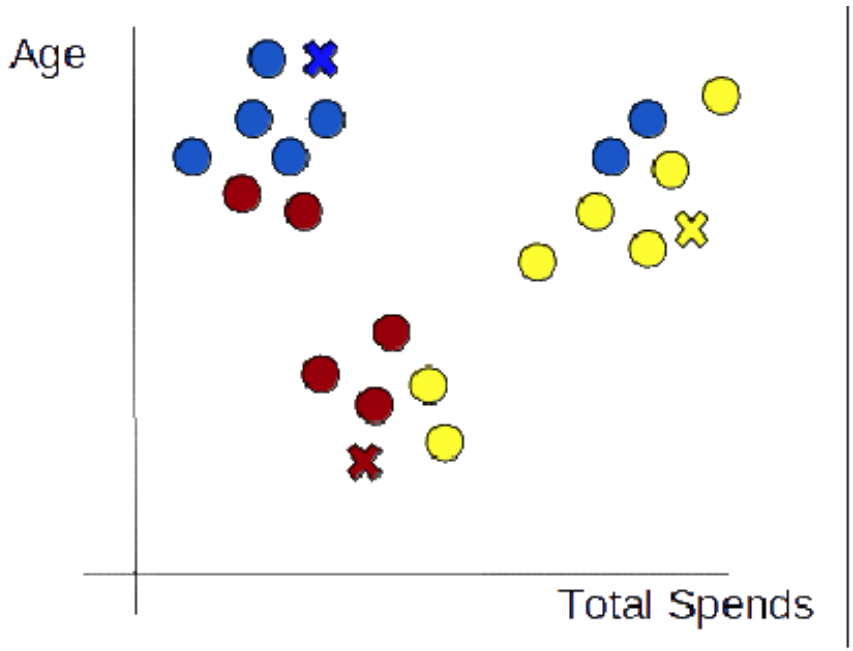


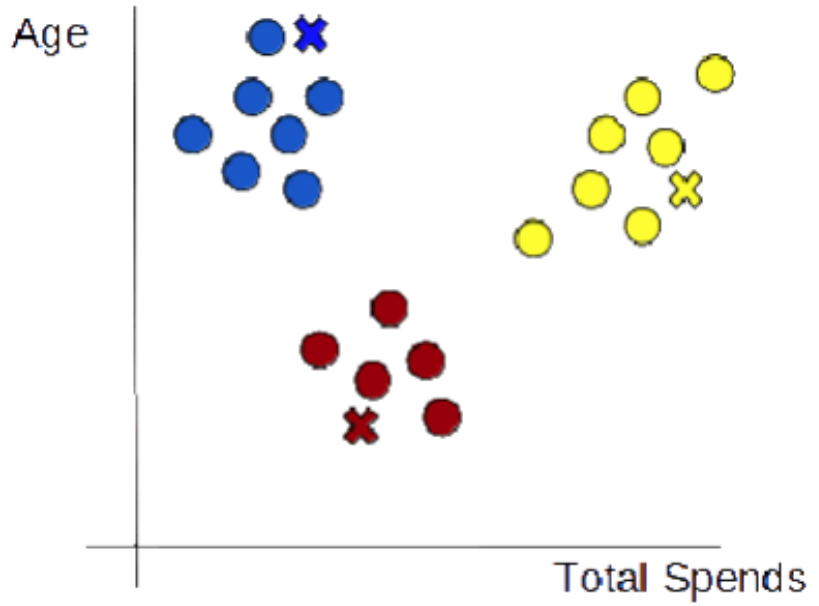
- Step 4 Re-initialize centroids: re-initialize the centroids by calculating the average of all data points of that cluster

$$C_i = \frac{1}{|N_i|} \sum x_i$$

- Repeating previous steps: We will keep repeating steps 3 and 4 until we have optimal centroids and the assignments of data points to correct clusters are not changing anymore.

Repetition of step 3 and 4

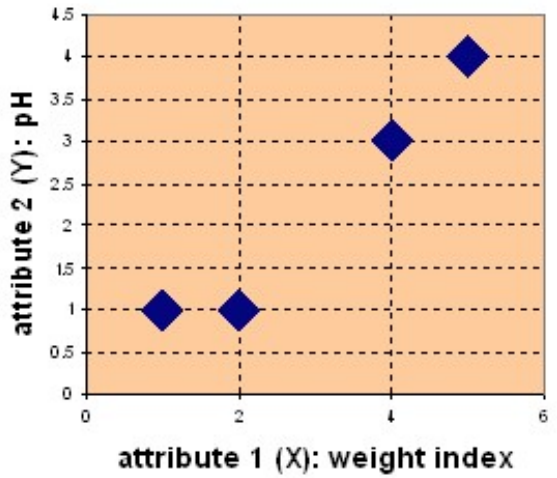




In class assignment

- Suppose we have several objects (4 types of medicines) and each object have two attributes or features as shown below. Our goal os to group these objects into $K = 2$ clusters of medicine based on two features (pH and weight index).

Object	weight Index	pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

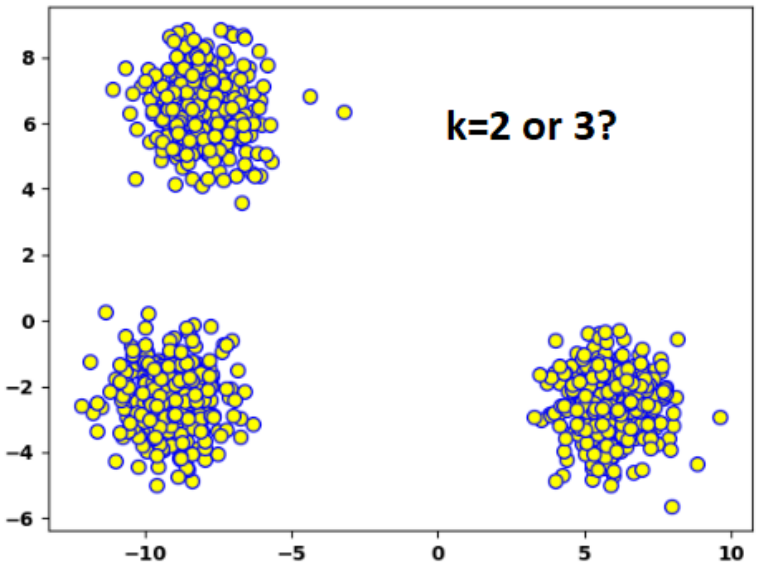


How to select K in k-mean?

- k-mean is simple and perhaps the most commonly used algorithm for clustering.
- But there is a catch. How do you decide the number of clusters?

The Elbow Method

The Silhouette Method



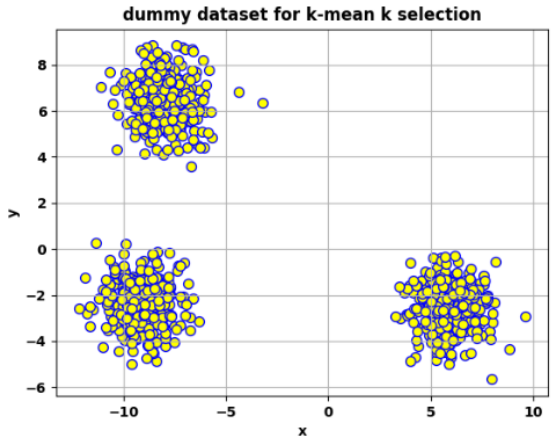
How to select K in k-mean?

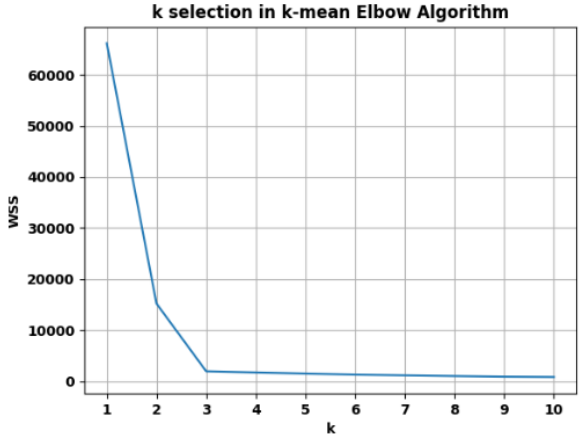
The Elbow Method

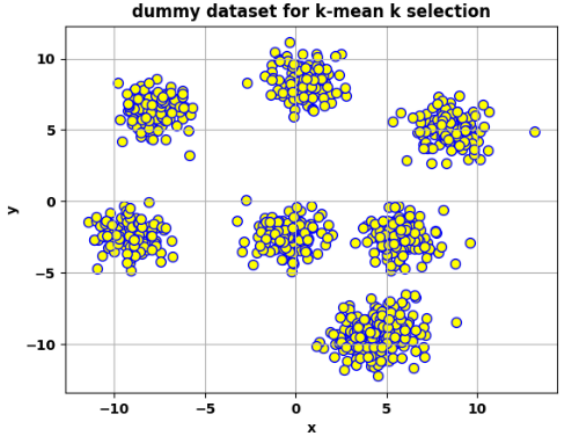
- Calculate the Within-Cluster-Sum of Squared Errors (WSS) for different values of k, and choose the k for which WSS becomes first starts to diminish. In the plot of WSS-versus-k, this is visible as an elbow.
- The Squared Error for each point is the square of the distance of the point from its representation i.e. its predicted cluster center.
- The WSS score is the sum of these Squared Errors for all the points.
- Any distance metric like the Euclidean Distance or the Manhattan Distance can be used.

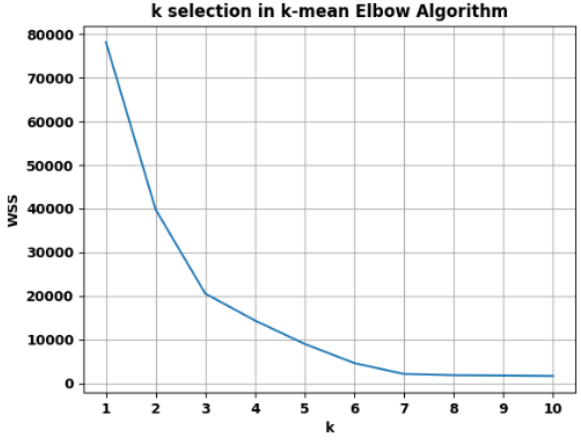
Elbow Method Result

- As the WSS plot looks like an arm with a clear elbow at $\boxed{k = 3}$.
- Unfortunately, we do not always have such clearly clustered data. This means that the elbow may not be clear and sharp.









The Silhouette Method

Silhouette Analysis

- The silhouette value measures how similar a point is to its own cluster (cohesion) compared to other clusters (separation).
- The range of the Silhouette value is between $+1$ and -1. A high value is desirable and indicates that the point is placed in the correct cluster.
- A score of 1 denotes the best, meaning that point is very compact within the cluster to which it belongs and far away from the other clusters. The worst value is -1. Values near 0 denote overlapping clusters.

$$S_i = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

The Silhouette Method

- where S_i is the silhouette coefficient of the data point i.
- a_i is the measure of similarity of the point i to its own cluster. It is the average distance between i and all the other data points in the cluster to which i belongs.
- b_i is measure of dissimilarity of i from points in other clusters. It is the average distance from i to all clusters to which i does not belong.

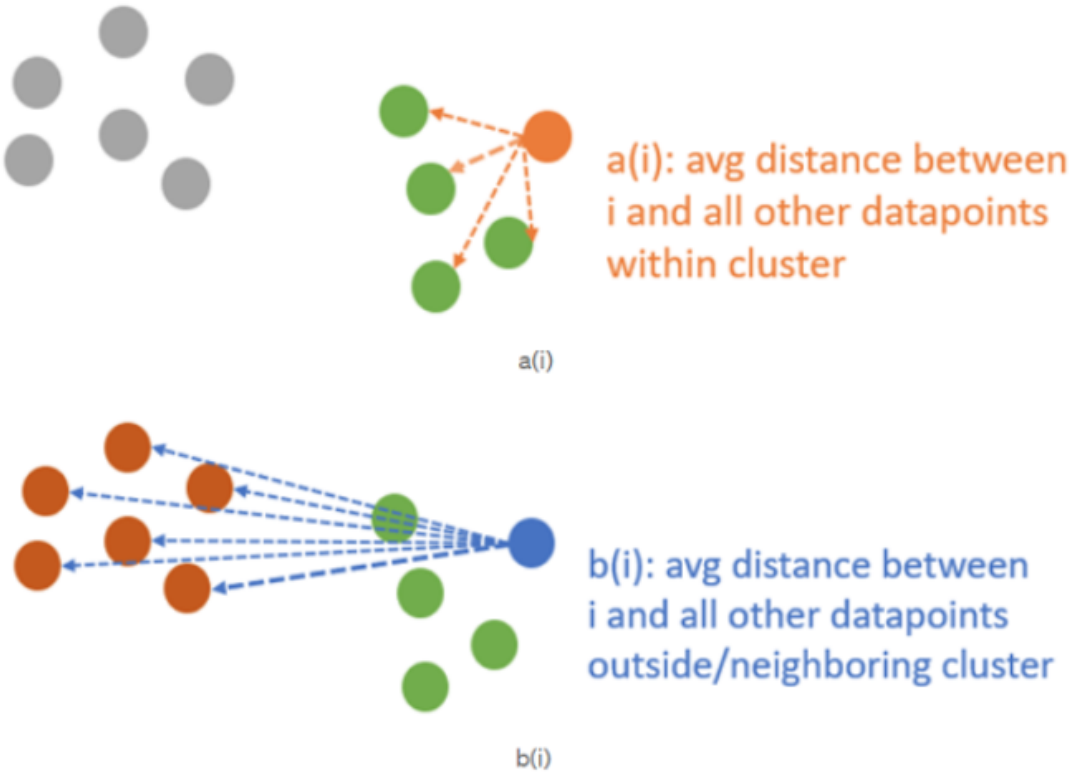
Points to remember

- The value of the silhouette coefficient is between [-1,1]
- A score of 1 denotes the best, meaning that the data point i is very compact within the cluster to which it belongs and far away from the other clusters.
- The worst value is -1. Values near 0 denote overlapping clusters.

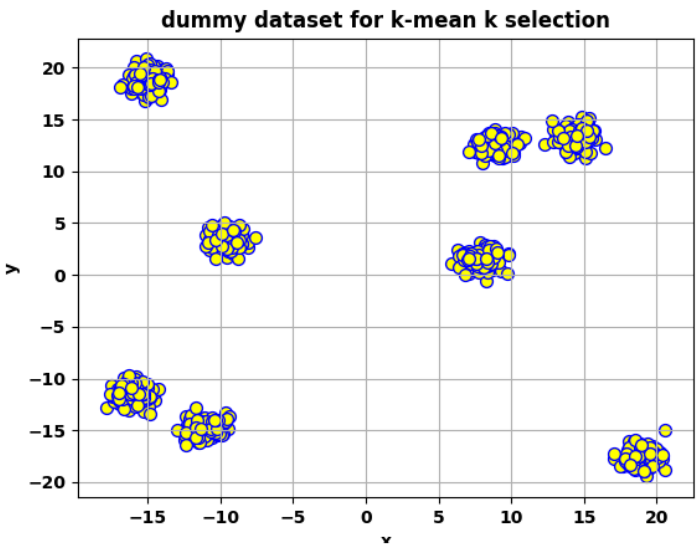
The Silhouette Method

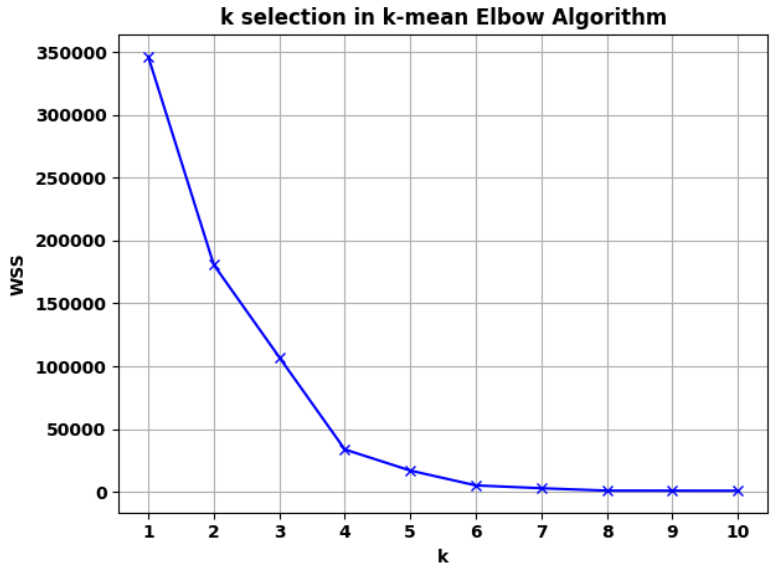
- We will then calculate the average_silhouette for every k.

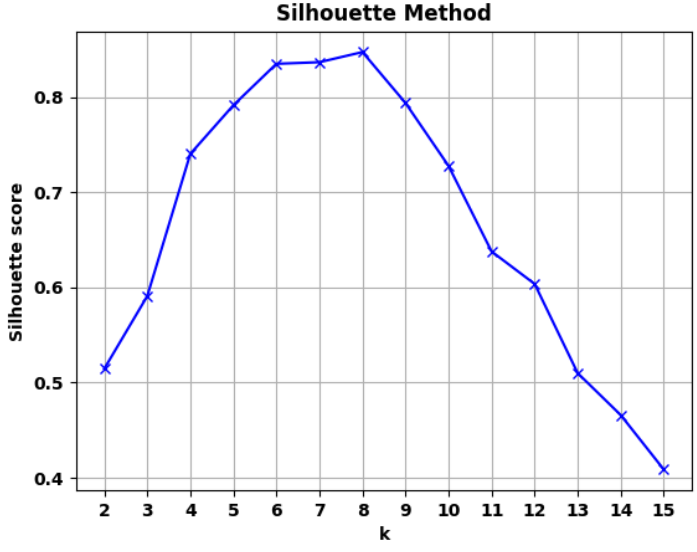
$$AverageSilhouette = mean\{S_i\}$$



Result with dummy dataset







Clustering Metrics

- Silhouette Score
- Davies-Bouldin Index
- Calinski-Harabasz Index (Variance Ratio Criterion)
- Adjusted Rand Index (ARI)
- Mutual Information (MI)

Davies-Bouldin Index

Davies-Bouldin Index

- It evaluates a dataset's clusters' compactness and separation.
- Lower Davies-Bouldin Index the better,since it is comparing each cluster's average similarity-to-dissimilarity ratio to that of its most similar neighbor.

$$DB = \left(\frac{1}{n}\right) \sum max(R_{ij})$$

where n is the number of clusters
R_{ij} is a measure of dissimilarity between cluster i and the cluster most similar to i

Variance Ratio Criterion

Calinski-Harabasz Index

- Higher values indicate compact and well-separated clusters.
- It computes the ratio of the within-cluster variance to the between-cluster variance.

$$CH = \left(\left(\frac{B}{W} \right) * \left(\frac{N-K}{K-1} \right) \right)$$

B sum of squares between clusters. W sum of squares within clusters. N number of data points. K number of clusters.

Variance Ratio Criterion

$$B = \sum_{k=1}^{K} n_k \times ||C_k - C||^2$$

$$W = \sum_{i=1}^{n_k} ||X_{ik} - C_k||^2$$

n_k number of observation in cluster k. X_{ik} the i-th observation of cluster k. C_k centroid of the cluster k.

Adjusted Rand Index (ARI)

ARI

- It evaluates whether data point pairs are clustered together or apart in both the true and anticipated clusterings.
- Higher values of the index imply better agreement.
- It ranges from -1 to 1, where 1 indicates perfect clustering, 0 indicates random clustering, and negative values suggest poor clustering.

$$ARI = \frac{(RI - E[RI])}{(max(RI) - E[RI])}$$

RI Rand Index.
$E[RI]$ expected value of Rand index.

Mutual Information (MI)

MI

- It evaluates the degree of agreement between the actual and expected cluster designations in the context of clustering evaluation.
- High MI values indicate better alignment between clusters and true labels, signifying good clustering results.

$$MI(y,z) = \sum\sum p(y_i, z_j) * \log\left(\frac{p(y_i, z_j)}{p(y_i) * p'(z_j)}\right)$$

y_i is a true label.
z_i is a predicted label.
$p(y_i, z_i)$ is the joint probability of yi and zj.
$p(y_i)$ and $p'(z_i)$ are the marginal probabilities.

Steps to Evaluate Clustering Using Sklearn

```python
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, davies_bouldin_score, calinski_harabasz_score
from sklearn.metrics import mutual_info_score, adjusted_rand_score
# Example using a built-in dataset (e.g., Iris dataset)
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data
kmeans = KMeans(n_clusters=3)
kmeans.fit(X)
silhouette = silhouette_score(X, kmeans.labels_)
db_index = davies_bouldin_score(X, kmeans.labels_)
ch_index = calinski_harabasz_score(X, kmeans.labels_)
ari = adjusted_rand_score(iris.target, kmeans.labels_)
mi = mutual_info_score(iris.target, kmeans.labels_)

# Print the metric scores
print(f"Silhouette Score: {silhouette:.2f}")
print(f"Davies-Bouldin Index: {db_index:.2f}")
print(f"Calinski-Harabasz Index: {ch_index:.2f}")
print(f"Adjusted Rand Index: {ari:.2f}")
print(f"Mutual Information (MI): {mi:.2f}")
```

Results

- Silhouette Score (0.55): Closer to 1 values suggest better-defined clusters.
- Davies-Bouldin Index (0.66): A lower score is preferable, and 0.66 suggests a pretty strong separation across clusters.
- The score Index (561.63) calculates the ratio of between-cluster variation to within-cluster variance. Higher values suggest more distinct groups. Your clusters are distinct and independent with a score of 561.63.
- The Adjusted Rand Index (0.73) A rating of 0.73 shows that the clustering findings and the actual class labels correspond rather well.
- Mutual Information (MI) (0.75): A score of 0.75 indicates a substantial amount of shared information between the true labels and the clusters assigned by the algorithm. It signifies that the clustering solution captures a significant portion of the underlying structure in the data, aligning well with the actual class labels