EMORY
UNIVERSITY

Emory Continuing Education

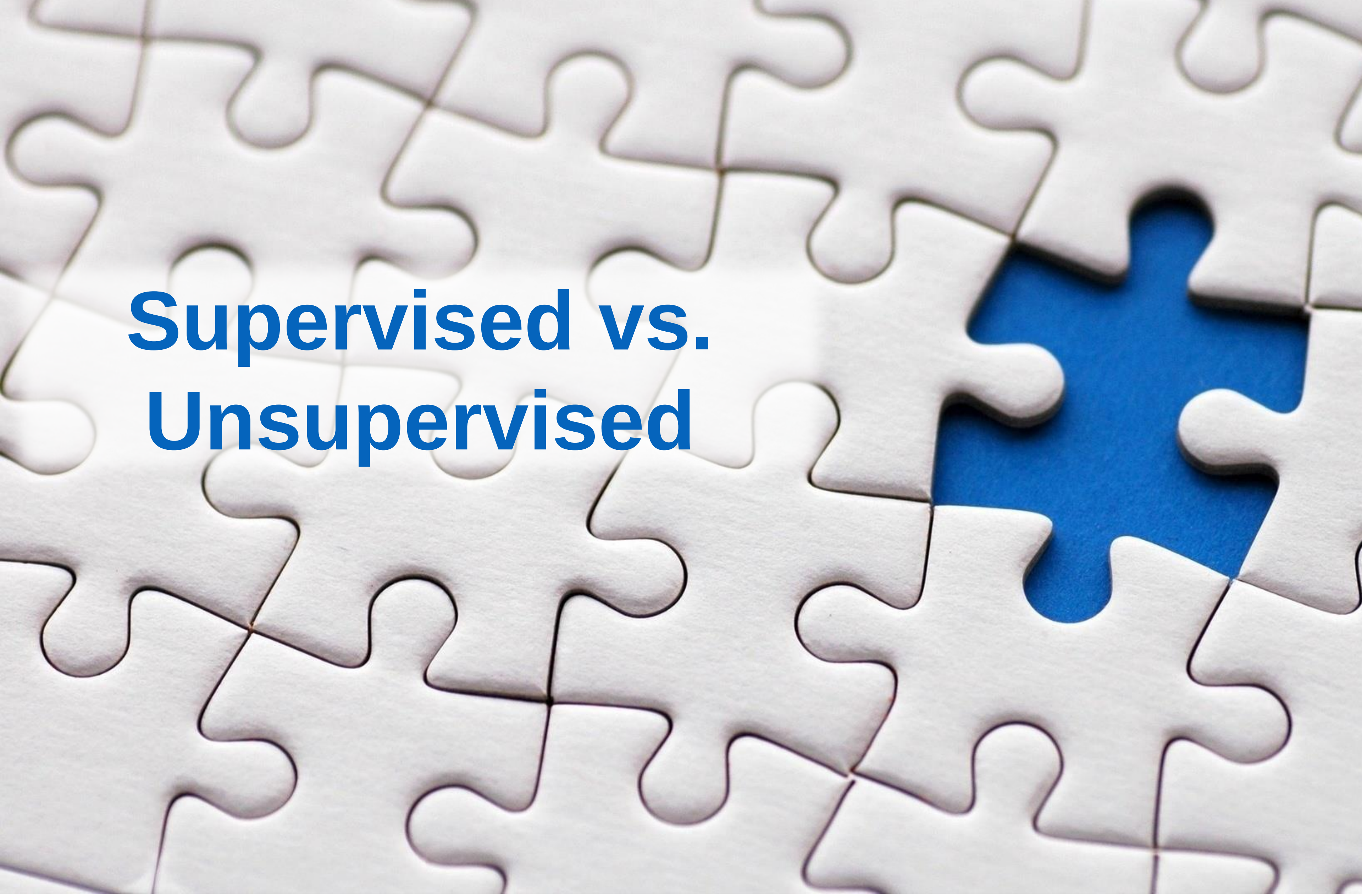
Unsupervised Learning

Sridhar Palle, Ph.D

Consort Institute

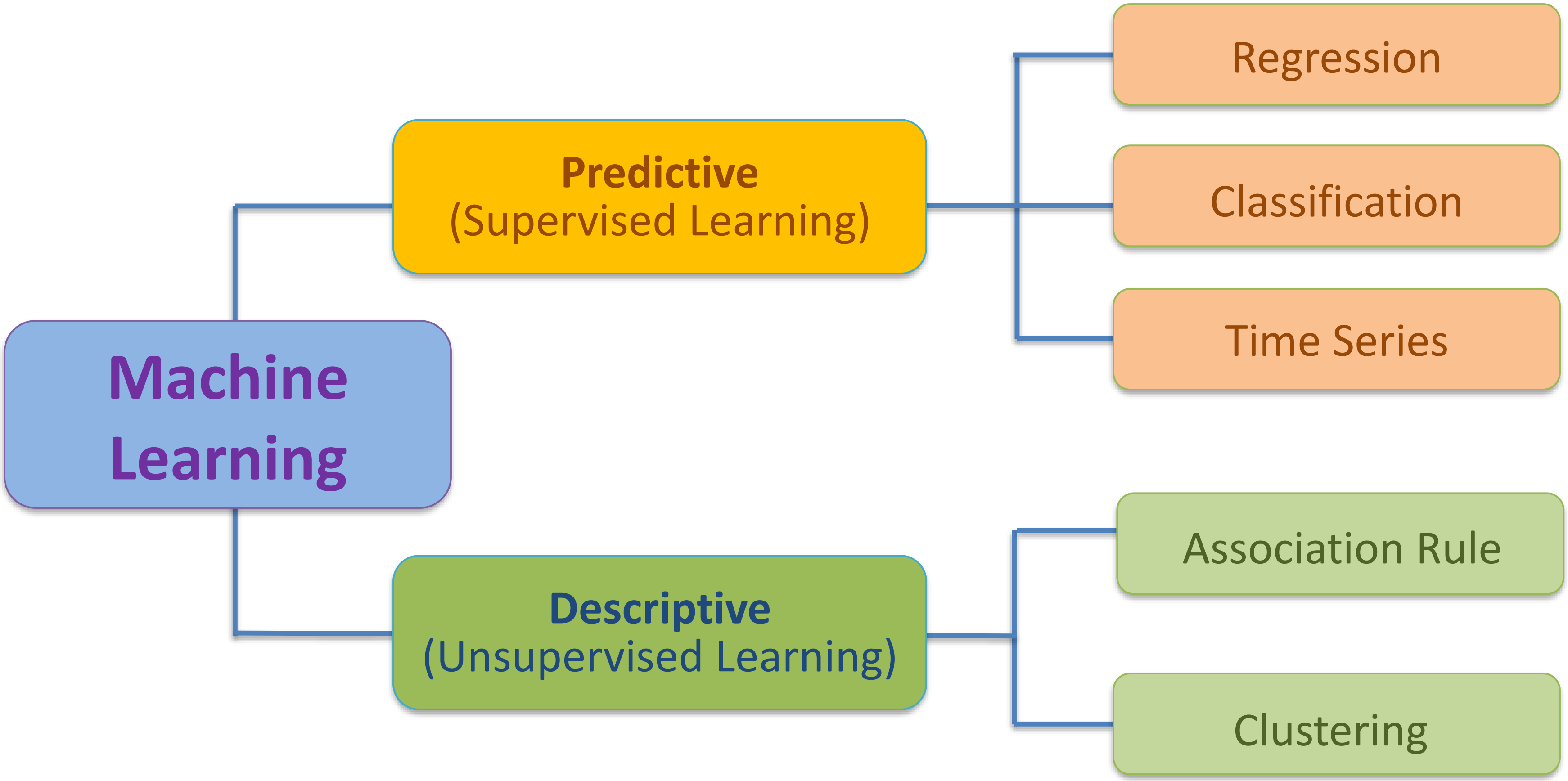
Module Objectives

- Review the definitions of supervised and unsupervised learning
- Introduce some popular unsupervised learning techniques
- Learn about association rule mining and when it can be used
- Learn how clustering techniques work and when they can be applied



Supervised vs. Unsupervised

Machine Learning Techniques



Machine Learning

Predictive
(Supervised Learning)

Regression

Classification

Time Series

Descriptive
(Unsupervised Learning)

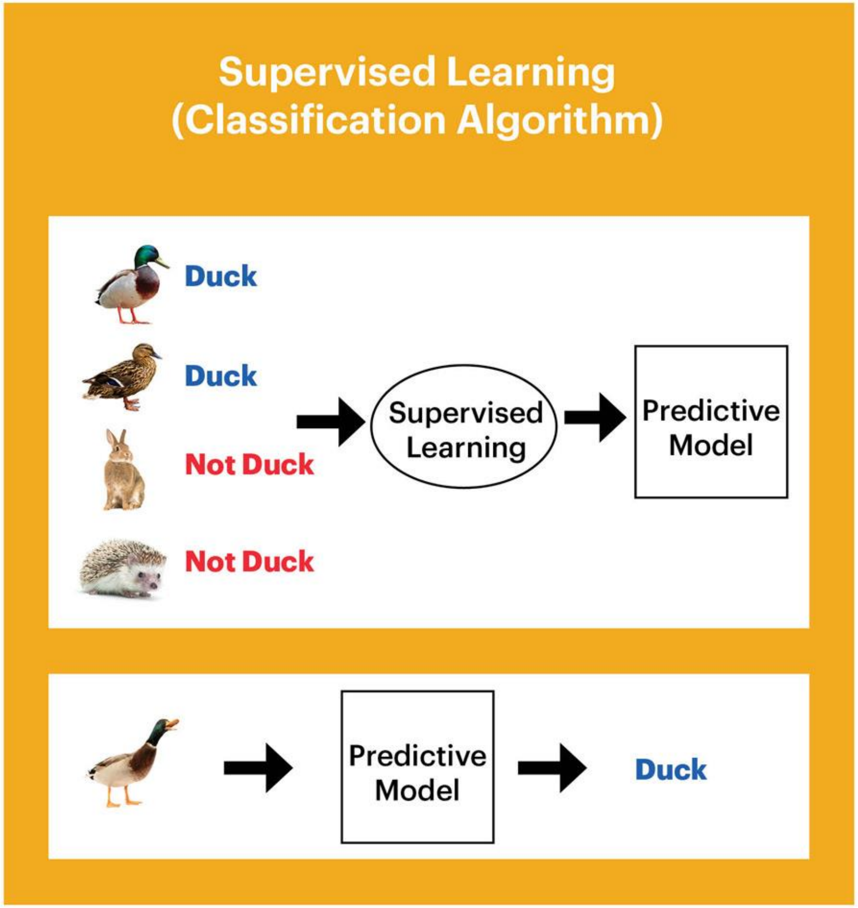
Association Rule

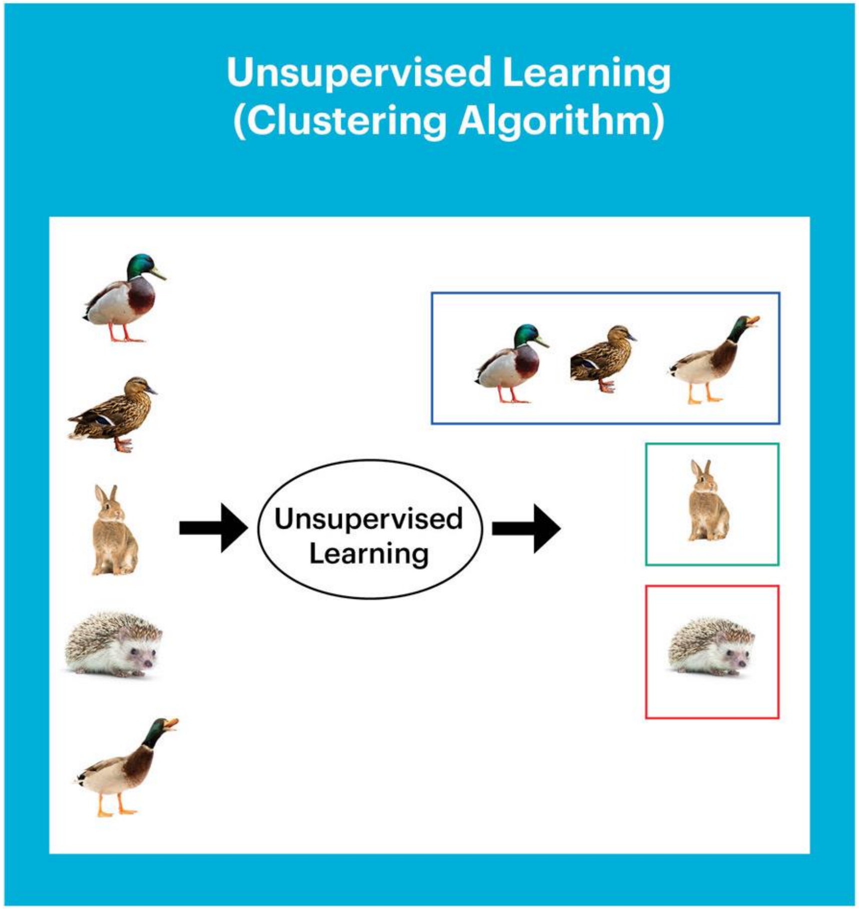
Clustering

Supervised vs Unsupervised

Labelled Data

No Labelled Data





Western Digital.

Unsupervised Learning

- Unsupervised learning is the task of describing hidden structure from unlabeled data (no target variable)
- Since the examples given to the “learner” are unlabeled, the error metric to be minimized must be defined in a *general* sense, across the group, instead of right/wrong answers.
- The absence of an error or reward signal is the key difference between supervised and unsupervised learning.

Unsupervised Learning Techniques

- Association Rules Mining
- Clustering
- Dimensionality Reduction

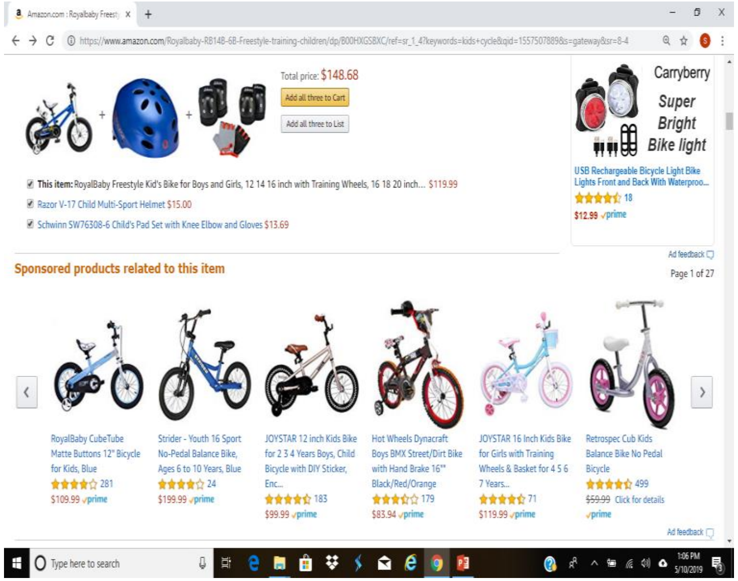
A.R.M.

Association Rules Mining (ARM)

- Association Rules Mining (ARM) is a rules based Machine Learning technique for discovering relationships between different items (or products) in a large dataset

- Objective is not to predict the occurrence of an item, but to find usable, hidden patterns among the available items (or item sets) in a transactional dataset

- The goal is to identify the co-occurrences of items/item sets with reasonable confidence


Amazon.com : Royalbaby Freesty
https://www.amazon.com/Royalbaby-RB14B-6B-Freestyle-training-children/dp/B00HXGSBXC/ref=sr_1_4?keywords=kids+cycle&qid=1557507889&s=gateway&sr=8-4
Type here to search
1:06 PM
5/10/2019

Total price: $148.68

Add all three to Cart

Add all three to List

☑ **This item:** RoyalBaby Freestyle Kid's Bike for Boys and Girls, 12 14 16 inch with Training Wheels, 16 18 20 inch... $119.99

☑ Razor V-17 Child Multi-Sport Helmet $15.00

☑ Schwinn SW76308-6 Child's Pad Set with Knee Elbow and Gloves $13.69

Carryberry Super Bright Bike light

USB Rechargeable Bicycle Light Bike Lights Front and Back With Waterproo... ★★★★☆ 18 $12.99 prime

Ad feedback

Sponsored products related to this item

Page 1 of 27

RoyalBaby CubeTube Matte Buttons 12" Bicycle for Kids, Blue ★★★★☆ 281 $109.99 prime

Strider - Youth 16 Sport No-Pedal Balance Bike, Ages 6 to 10 Years, Blue ★★★★☆ 24 $199.99 prime

JOYSTAR 12 inch Kids Bike for 2 3 4 Years Boys, Child Bicycle with DIY Sticker, Enc... ★★★★☆ 183 $99.99 prime

Hot Wheels Dynacraft Boys BMX Street/Dirt Bike with Hand Brake 16"" Black/Red/Orange ★★★☆☆ 179 $83.94 prime

JOYSTAR 16 Inch Kids Bike for Girls with Training Wheels & Basket for 4 5 6 7 Years... ★★★★☆ 71 $119.99 prime

Retrospec Cub Kids Balance Bike No Pedal Bicycle ★★★★☆ 499 ~~$59.99~~ Click for details prime

Ad feedback

ARM Advantages

- Obvious Ones:

- Not so obvious
  - Prized discoveries
  - Business can profit from

- Advantages
  - Bundle Pricing
  - Product placement
  - Shelf space optimization

- Downside
  - May also lead to spurious relationships when dealing with data with billions of transactions

ARM Overview

- Association Rule Mining is the task of finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in a data set

- Mathematically rooted in Bayes’ conditional probability theorem which calculates the likelihood that Event B will occur given that we’ve just witnessed Event A.

- Primary focus is in generating rules such as

{Item A} -> {Item B}   [support, confidence]

Ex: {Baby Formula} → {Diaper}   [25, 50]

Important Terms:

- Support = Do these things co-occur enough for me to care?
- Confidence = Does Item A give me actionable information about Item B?

ARM Procedure & Challenges

Step 1: Prepare data in a particular format

Step 2: Short-list frequently occurring items (or items sets), based on some support level

Step 3: Generate relevant association rules from item sets generated in step 2 (based on some confidence parameters)

{Item A} -> {Item B} [Support, Confidence, Lift]

Item A – Antecedent or premise of a rule

Item B – Consequent or conclusion of a rule.

Main challenge for association rule analysis are

- Computational time and resources
- Ex: For association analysis of ‘n’ items, there will be

  2^n -1 item sets, and $3^n – 2^n + 1$ association rules can be found
- Fortunately there are algorithmic approaches to efficiently find the frequent item sets and rules based on some parameters (support, confidence, lift)

Support, Confidence, Lift

- **Support (A) =** $\dfrac{Number\ of\ Occurences\ of\ Item\ A}{Total\ Number\ of\ transactions}$

- **Confidence (A → B) =** $\dfrac{Support(A,B)}{Support(A)}$

- Lift (A → B) = $\dfrac{Support(A,B)}{Support(A) * Support(B)}$

ARM Practical Example: Grocery Store

Step 1: Prepare data in a particular format

Step 2: Short-list frequently occurring items/item sets based on some support level

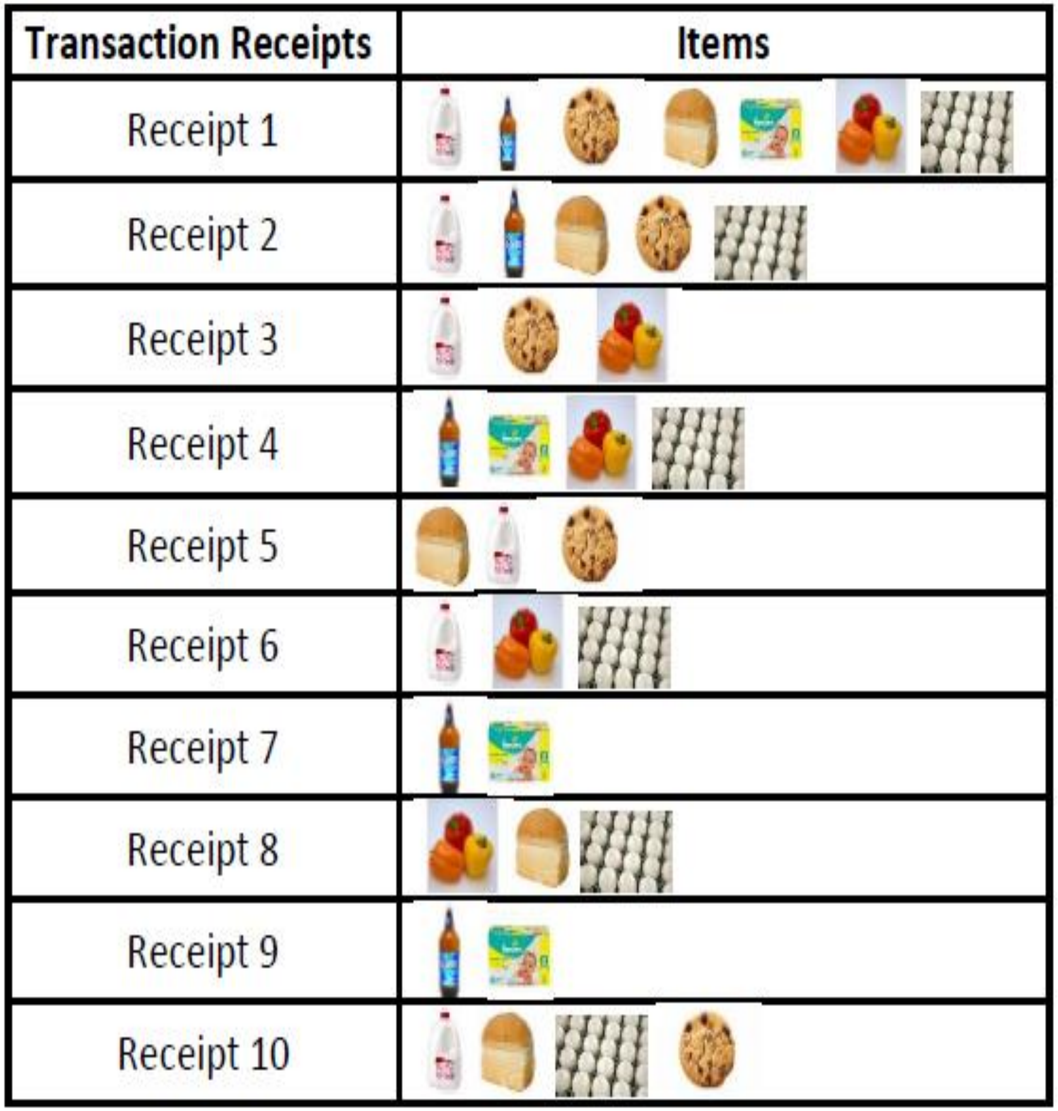
Support for Milk { 🥛 } = 60%,

Support of Cookies { 🍪 } = 50%,

Support of Milk and cookies { 🥛 & 🍪 } = 50%

Step 3: Generate Rules

{ 🥛 → 🍪 }



Transaction Receipts	Items
Receipt 1	🥛 🍾 🍪 🍞 🧷 🫑 🥚
Receipt 2	🥛 🍾 🍞 🍪 🥚
Receipt 3	🥛 🍪 🫑
Receipt 4	🍾 🧷 🫑 🥚
Receipt 5	🍞 🥛 🍪
Receipt 6	🥛 🫑 🥚
Receipt 7	🍾 🧷
Receipt 8	🫑 🍞 🥚
Receipt 9	🍾 🧷
Receipt 10	🥛 🍞 🥚 🍪

ARM: Grocery store

- Rule Significance

$$\text{Confidence}\ \{🥛 \rightarrow 🍪\} = \frac{\text{Support of Milk and cookies}\ \{🥛\ \&\ 🍪\}}{\text{Support for Milk}\ \{🥛\}} = 83\%$$

$$\text{Lift}\ \{🥛 \rightarrow 🍪\} = \frac{\text{Support of Milk and cookies}\ \{🥛\ \&\ 🍪\}}{\text{Support of Milk}\ \{🥛\} * \text{Support of Cookie}\ \{🍪\}} = 1.66$$

	Support	Confidence	Lift
Milk → Cookies	50%	83%	1.66
Milk → Eggs	40%	66%	1.11

Terminology

Why Use Support?

- Removes rules with low potential business relevance.
- Support can make association rule discovery more efficient.

Why Use Confidence?

- Indicates which items in a set have higher information value.
- Can be used to estimate the business impact of a rule.
- Measures the probability of Item B occurring because Item A has occurred.

Why Use Lift?

- Measures the importance of a rule
- Rules greater than 1 imply greater significance
- To check if Item A really has a positive effect in the occurrence of Item B

Apriori Algorithm

- Frequent itemset generation can become complex as the number of items in a database grows.

- The apriori algorithm helps manage this complexity.

Apriori Principle:

- If an itemset is frequent, then its subsets must also be frequent.
- If an itemset is infrequent, then all of its supersets must be infrequent as well.
- The support of an itemset $\leq$ the support of its subsets.

Apriori Steps

- Apriori uses a breadth-first search process.
- Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time.
- The apriori algorithm has 2 steps:
  - Item set generation (breadth-first)
  - Association rule generation

Apriori: Step 1

Customer ID	Items Purchased
5279	A, B, C
2980	A, C
6314	A, D
9065	B, E, F
1142	C, A, D, E

Min. Support = 50%
Min. Confidence = 60%

Item Set	Support
{A}	(4/5) = 80%
{B}	(2/5) = 40%
{C}	(3/5) = 60%
{D}	(2/5) = 40%
{E}	(2/5) = 40%
{F}	(1/5) = 20%
{A,B}	(1/5) = 20%
{A,C}	(3/5) = 60%
{A,D}	(2/5) = 40%
{A,E}	(1/5) = 20%
{A,F}	(0/5) = 0%

Apriori: Step 1

Customer ID	Items Purchased
5279	A, B, C
2980	A, C
6314	A, D
9065	B, E, F
1142	C, A, D, E

Min. Support = 50%
Min. Confidence = 60%

Item Set	Support
{A}	(4/5) = 80%
~~{B}~~	~~(2/5) = 40%~~
{C}	(3/5) = 60%
~~{D}~~	~~(2/5) = 40%~~
~~{E}~~	~~(2/5) = 40%~~
~~{F}~~	~~(1/5) = 20%~~
~~{A,B}~~	~~(1/5) = 20%~~
{A,C}	(3/5) = 60%
~~{A,D}~~	~~(2/5) = 40%~~
~~{A,E}~~	~~(1/5) = 20%~~
~~{A,F}~~	~~(0/5) = 0%~~

Apriori: Step 2

Customer ID	Items Purchased
5279	A, B, C
2980	A, C
6314	A, D
9065	B, E, F
1142	C, A, D, E

Min. Support = 50%
Min. Confidence = 60%

Item Set	Support
{A}	**(4/5) = 80%**
{A,C}	**(3/5) = 60%**

Rule	Confidence
A => B	(1/4) = 25%
A => C	(3/4) = 75%
A => D	(2/4) = 50%
A => E	(1/4) = 25%
A => F	(0/4) = 0%
C => A	(3/3) = 100%
C => B	(1/3) = 33%

Apriori: Step 2

Customer ID	Items Purchased
5279	A, B, C
2980	A, C
6314	A, D
9065	B, E, F
1142	C, A, D, E

Min. Support = 50%
Min. Confidence = 60%

Item Set	Support
{A}	(4/5) = 80%
{A,C}	(3/5) = 60%

Rule	Confidence
~~A => B~~	~~(1/4) = 25%~~
A => C	(3/4) = 75%
~~A => D~~	~~(2/4) = 50%~~
~~A => E~~	~~(1/4) = 25%~~
~~A => F~~	~~(0/4) = 0%~~
C => A	(3/3) = 100%
~~C => B~~	~~(1/3) = 33%~~

Apriori: Step 2

Customer ID	Items Purchased
5279	A, B, C
2980	A, C
6314	A, D
9065	B, E, F
1142	C, A, D, E

Min. Support = 50%
Min. Confidence = 60%

Item Set	Support
{A}	(4/5) = 80%
{A,C}	(3/5) = 60%

Rule	Confidence
A => C	(3/4) = 75%
C => A	(3/3) = 100%

Association Rule

“Body” => “Head” [support, confidence]

A => C [60%, 75%]

C => A [60%, 100%]

Apriori Principle

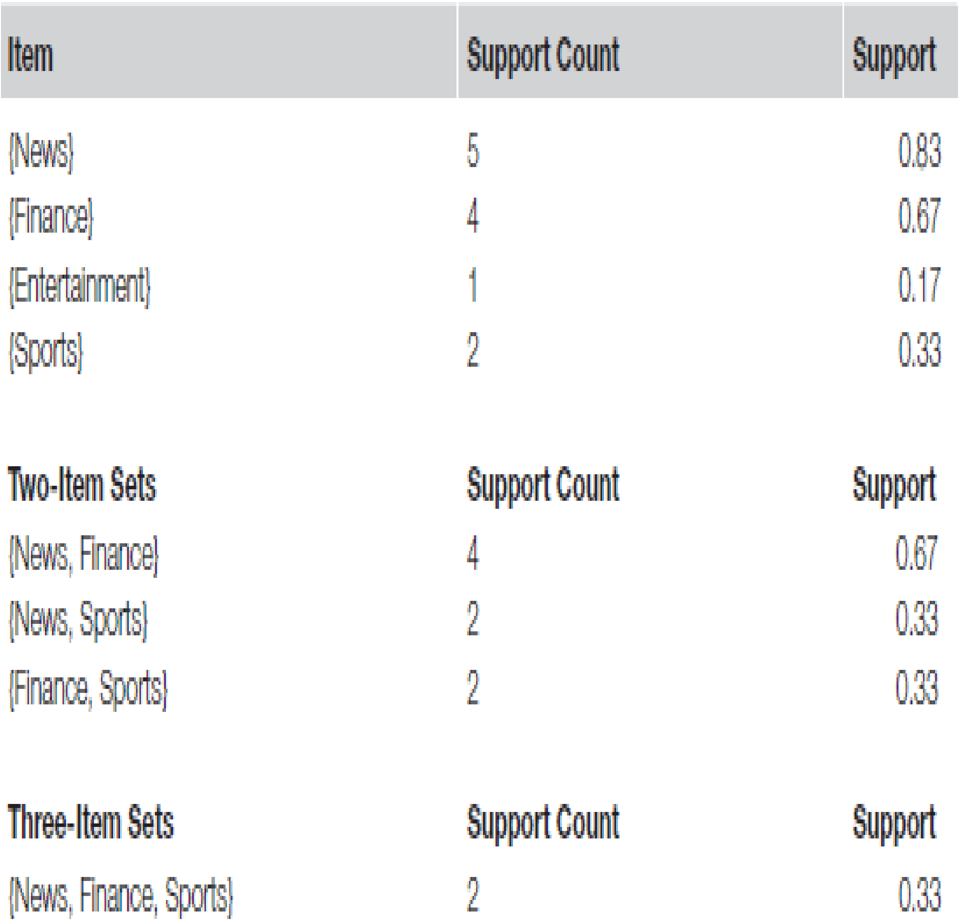
Apriori Principle:

- If an itemset is frequent, then its subsets must also be frequent.
- If an item set is infrequent, then all its supersets will also be infrequent
- The support of an itemset ≤ the support of its subsets.

Item Set	Support
{A}	**(4/5) = 80%**
~~{B}~~	~~(2/5) = 40%~~
{C}	**(3/5) = 60%**
~~{D}~~	~~(2/5) = 40%~~
~~{E}~~	~~(2/5) = 40%~~
~~{F}~~	~~(1/5) = 20%~~
~~{A,B}~~	~~(1/5) = 20%~~
{A,C}	**(3/5) = 60%**
~~{A,D}~~	~~(2/5) = 40%~~
~~{A,E}~~	~~(2/5) = 20%~~
~~{A,F}~~	~~(0/5) = 0%~~

Click-Stream Data Example

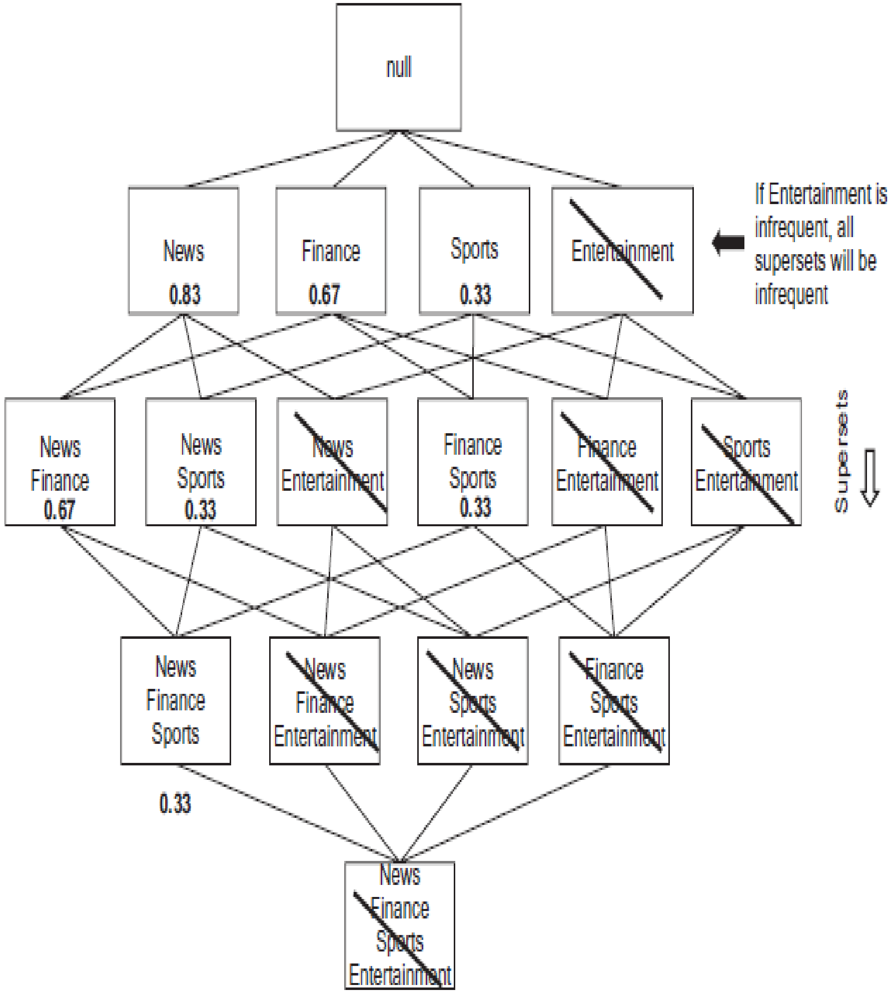
Support threshold = 0.25



Item	Support Count	Support
{News}	5	0.83
{Finance}	4	0.67
{Entertainment}	1	0.17
{Sports}	2	0.33

Two-Item Sets	Support Count	Support
{News, Finance}	4	0.67
{News, Sports}	2	0.33
{Finance, Sports}	2	0.33

Three-Item Sets	Support Count	Support
{News, Finance, Sports}	2	0.33

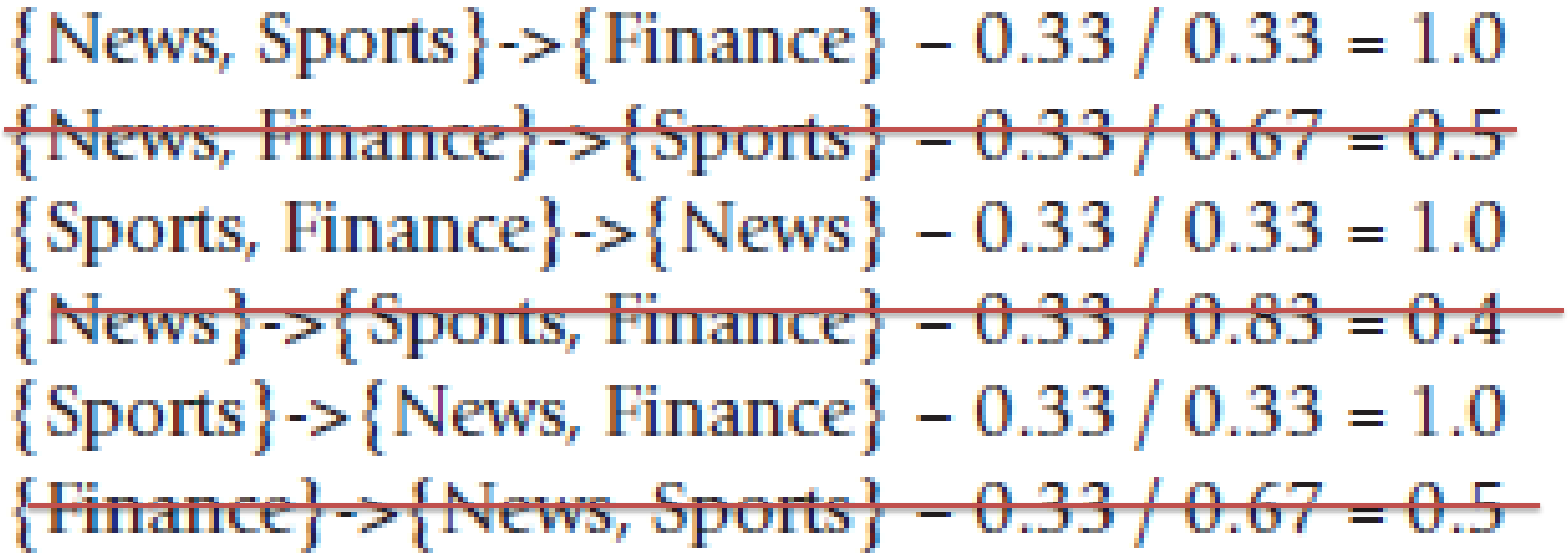


Source: Predictive Analytics & Data Mining - Kotu

Click-Stream Data Example

- Association Rules

Confidence threshold = 0.60



{News, Sports}->{Finance} – 0.33 / 0.33 = 1.0

~~{News, Finance}->{Sports} – 0.33 / 0.67 = 0.5~~

{Sports, Finance}->{News} – 0.33 / 0.33 = 1.0

~~{News}->{Sports, Finance} – 0.33 / 0.83 = 0.4~~

{Sports}->{News, Finance} – 0.33 / 0.33 = 1.0

~~{Finance}->{News, Sports} – 0.33 / 0.67 = 0.5~~

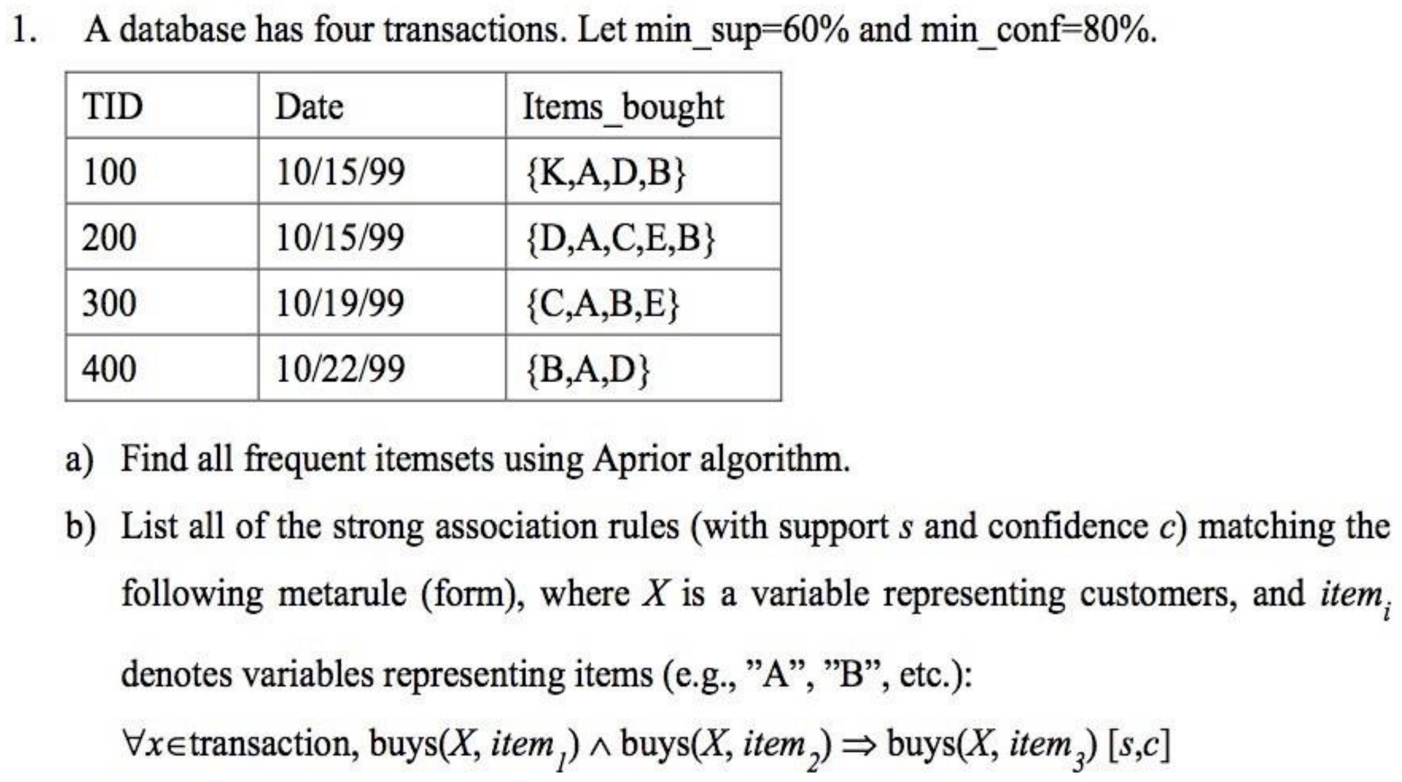
Confidence (A → B) = $\dfrac{Support(A,B)}{Support(A)}$

A.R.M. Practical Examples

Use Cases:

- Coupon printing at grocery store checkouts
- Remarketing ad content selection
- Drip marketing audience & campaign strategy
- “Add on” item recommendations
- Customer Lifetime Value estimation
- Product demand forecasting

A.R.M. Exercise 1



1. A database has four transactions. Let min_sup=60% and min_conf=80%.

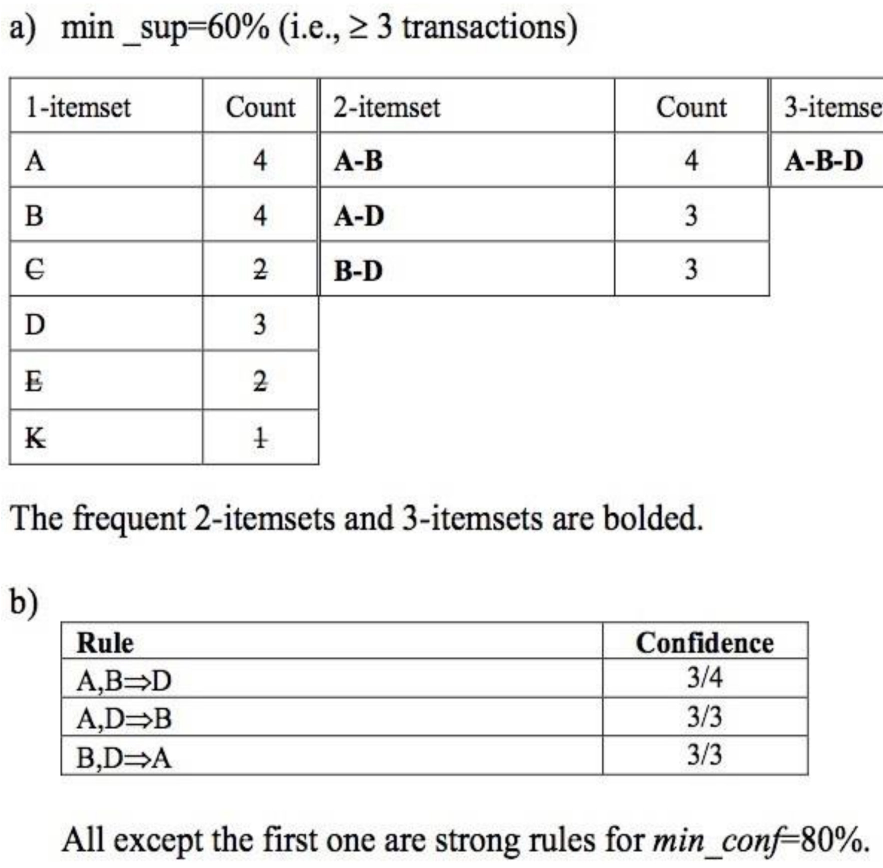
TID	Date	Items_bought
100	10/15/99	{K,A,D,B}
200	10/15/99	{D,A,C,E,B}
300	10/19/99	{C,A,B,E}
400	10/22/99	{B,A,D}

a) Find all frequent itemsets using Aprior algorithm.

b) List all of the strong association rules (with support *s* and confidence *c*) matching the following metarule (form), where *X* is a variable representing customers, and $item_i$ denotes variables representing items (e.g., ”A”, ”B”, etc.):

$\forall x \in$ transaction, buys(X, $item_1$) $\wedge$ buys(X, $item_2$) $\Rightarrow$ buys(X, $item_3$) [s,c]

A.R.M. Solution 1



a) min _sup=60% (i.e., ≥ 3 transactions)

1-itemset	Count
A	4
B	4
~~C~~	~~2~~
D	3
~~E~~	~~2~~
~~K~~	~~1~~

2-itemset	Count
A-B	4
A-D	3
B-D	3

3-itemset	Count
A-B-D	3

The frequent 2-itemsets and 3-itemsets are bolded.

b)

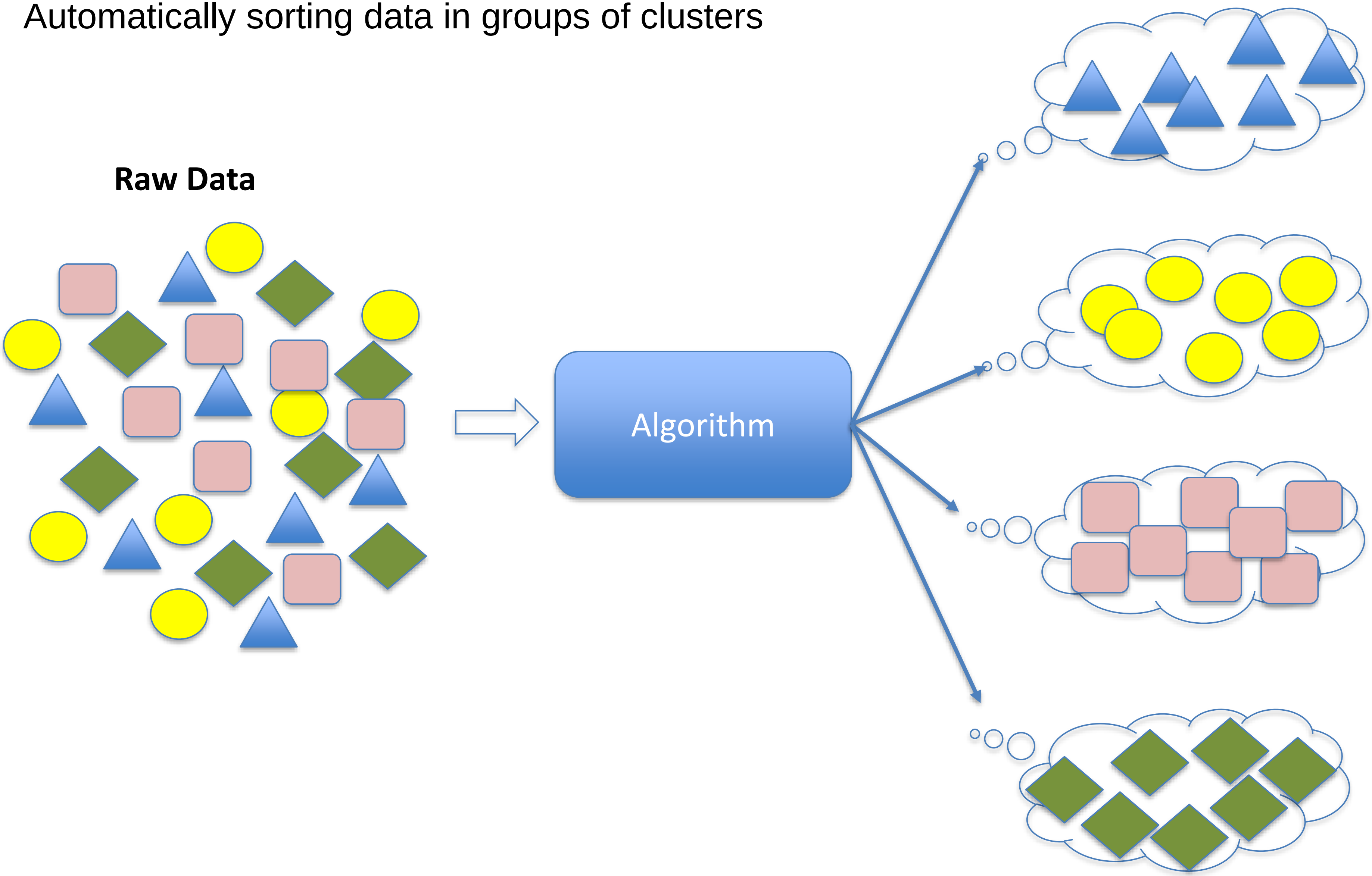
Rule	**Confidence**
A,B⇒D	3/4
A,D⇒B	3/3
B,D⇒A	3/3

All except the first one are strong rules for *min_conf*=80%.

Cluster Analysis

Clustering

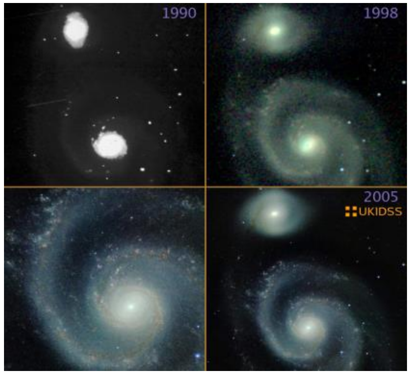
Automatically sorting data in groups of clusters

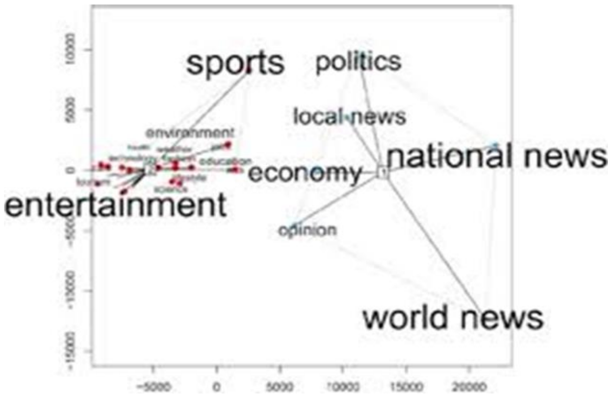


Clustering

Clustering is the process of automatically segregating dataset into different groups based on commonalities (similarities).

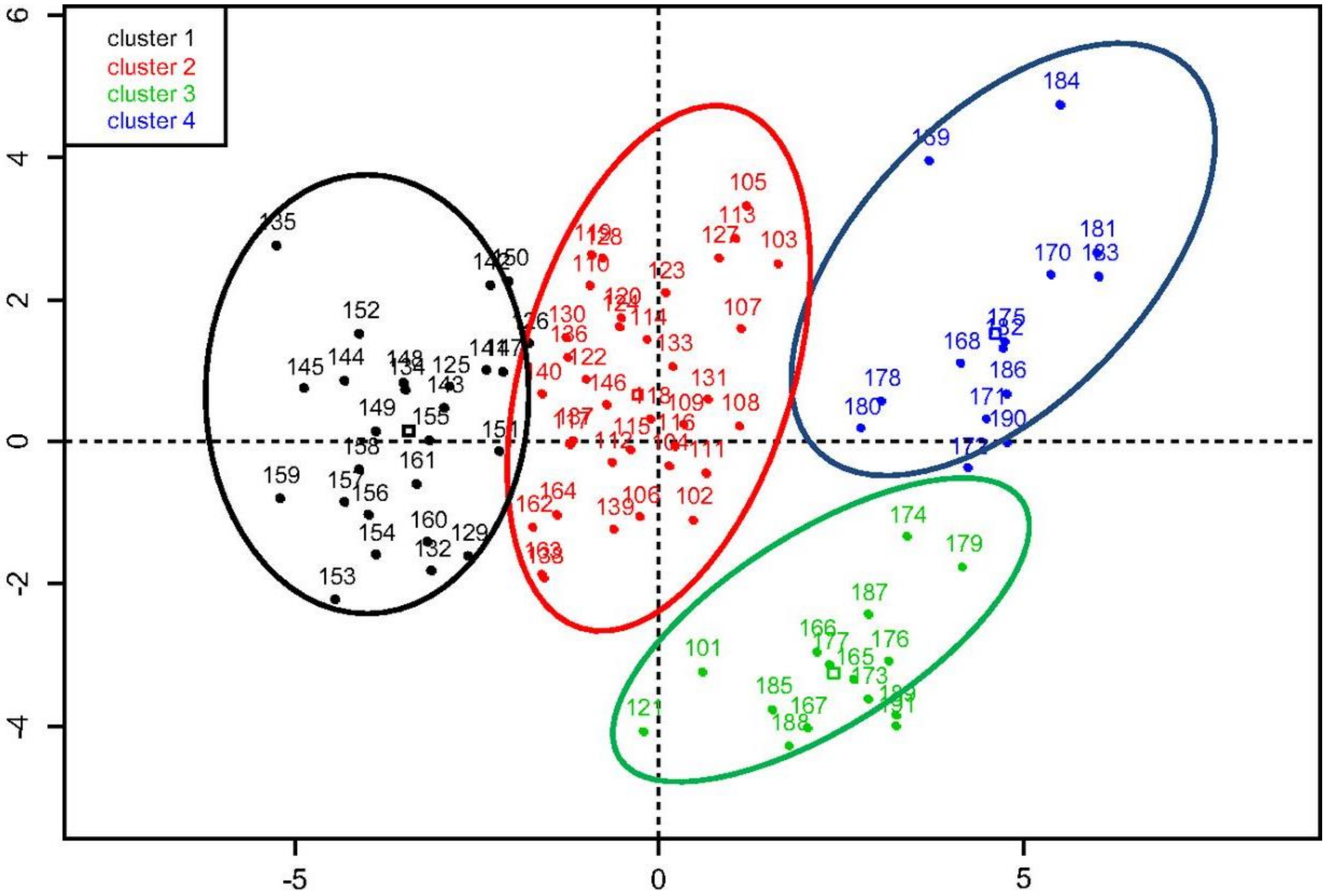
- Applications in diverse domains





Lets try to group them by similar shopping behaviours

Cluster Analysis



Overview

- Cluster analysis (a.k.a., clustering) is the task of grouping a set of objects in such a way that objects in the same group (a.k.a., cluster) are more similar to each other than to those in other groups (clusters).

- The “right” groupings are ***NOT*** known ahead of time. That is, they are not present in the data set.

Important Terms:

- Intra-class Similarity = Closeness of cluster members.
- Inter-class Similarity = Closeness of clusters.
- Centroid = Center of a cluster.

Terminology

Intra-class Similarity:

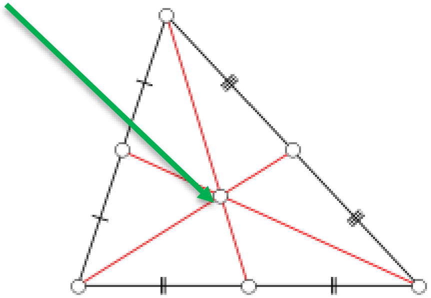
- How strongly units in the same group resemble each other.
- Greater similarity is better.

Inter-class Similarity:

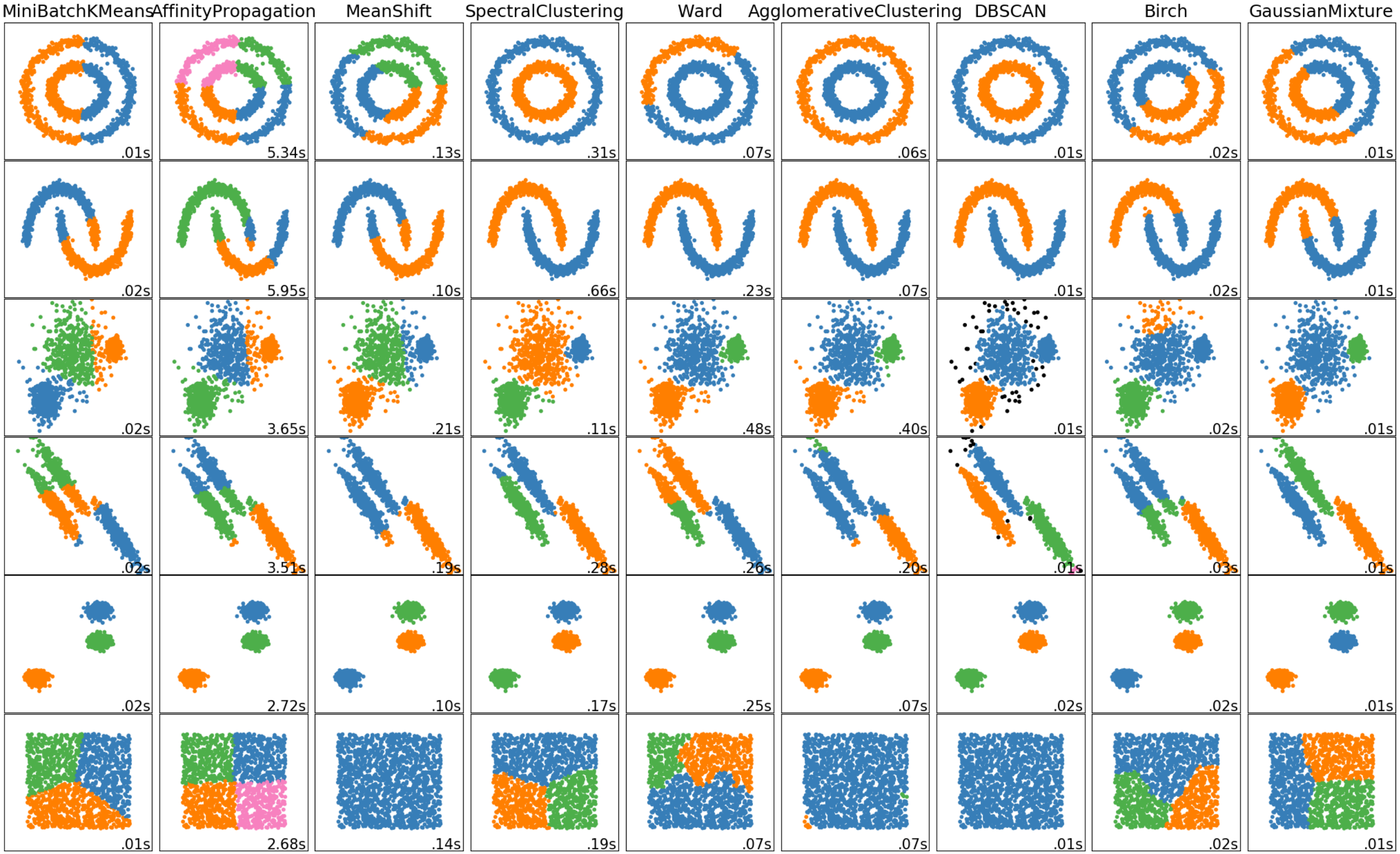
- How strongly units in different groups resemble each other.
- *Less* similarity is better.

Centroid:

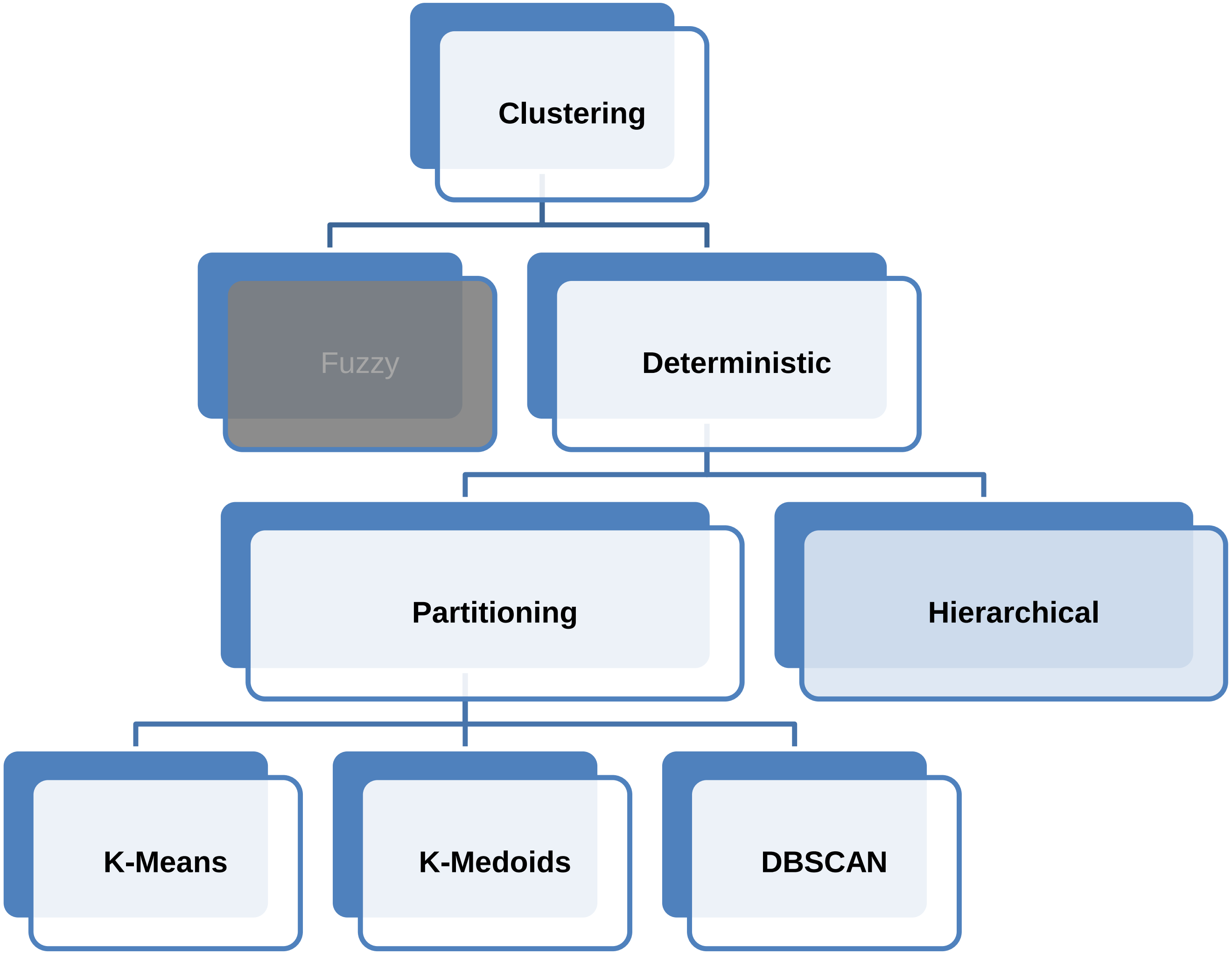
- The center of a cluster calculated as the average position of all the points in all of the coordinate directions.



Clustering Methods



Clustering Methods



Partitioning Process

1) Choose # of clusters and randomly pick locations for centers.

2) Assign each observation to the cluster whose mean yields the least within-cluster sum of squares (i.e., the nearest center).

3) Recalculate centers as the arithmetic average of all the data points in the cluster.

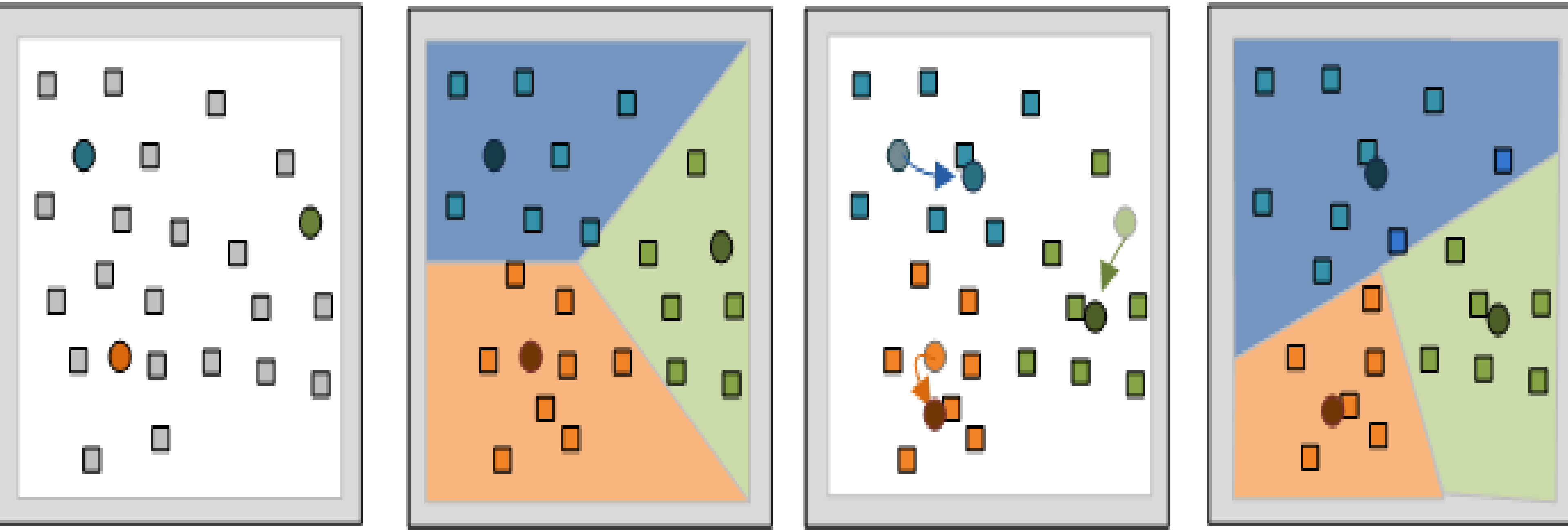
4) Repeat steps 2 & 3 until convergence or max iterations threshold is reached.

K-Means vs. K-Medoids

Define “Average”:

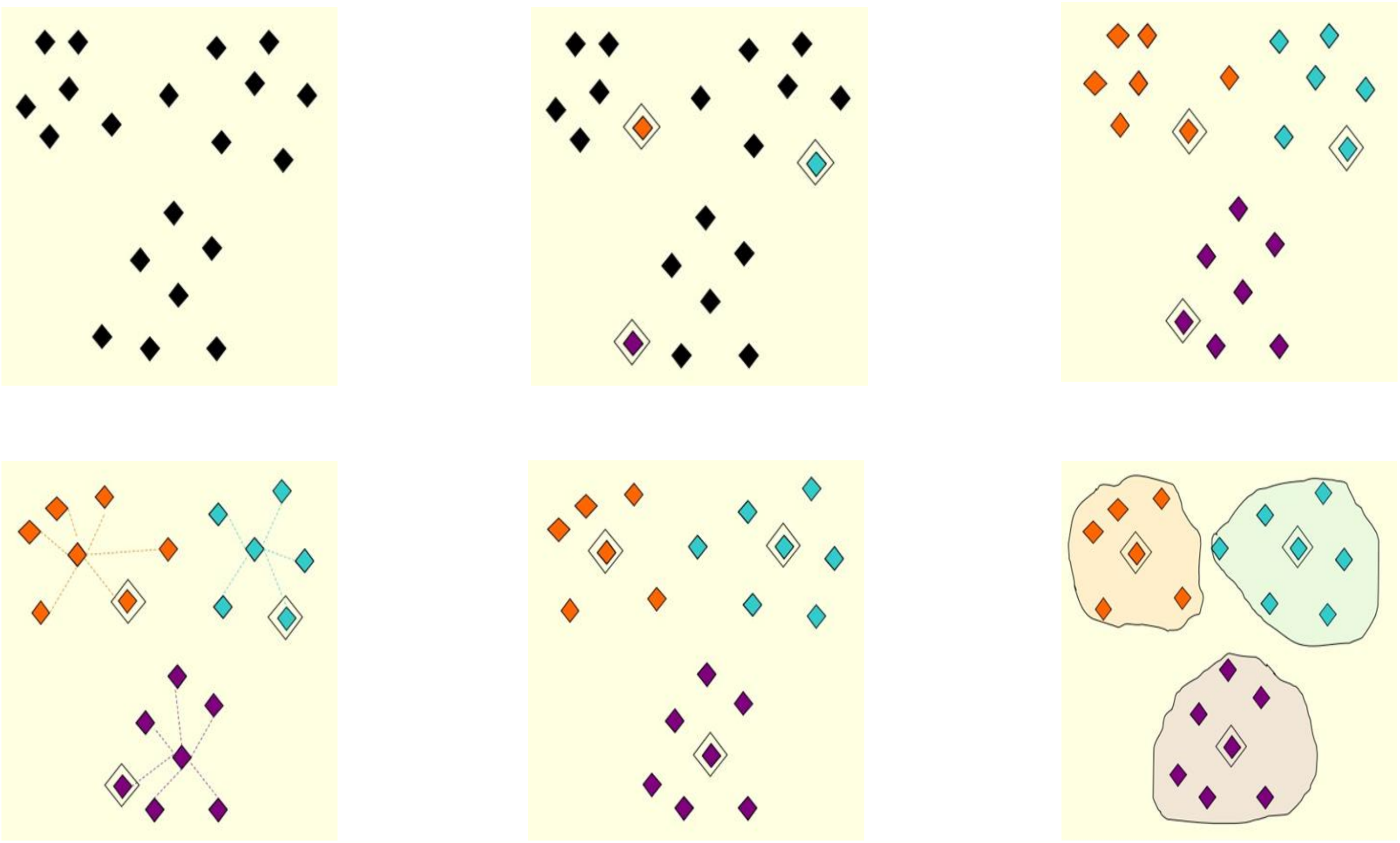
- The center of a cluster calculated as the average position of all the points in all of the coordinate directions.
- “Average” has different mathematical definitions.
  - Mean: sum of values / count of values
  - Median: middle value in an ordered list of values
- K-Means calculates *mean* positions from all cluster points as its average positions, called “centroids”.
- K-Medoids selects data points closest to the *median* of all cluster points as its average positions, called “medoids”.

K-Means in Action



http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

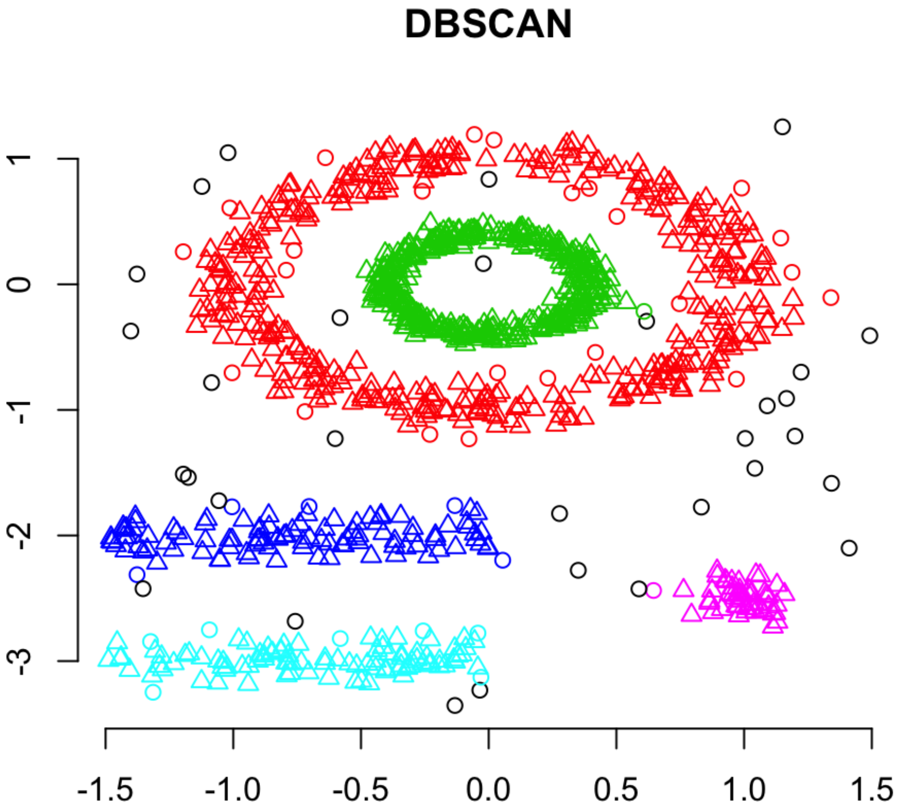
K-Medoids in Action



Density-Based Clustering

- Calculates the density of all data points in a dataset based on a chosen radius (Ɛ) and a MinPoints parameter.

- Then all high density spaces are categorized as separate clusters, surrounded by low density spaces which are just treated as noise.



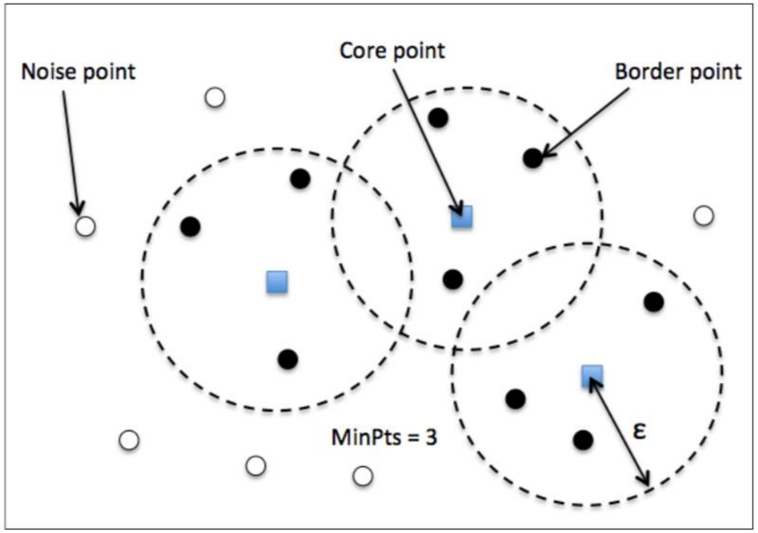
DBSCAN

- **Core point**: A point is considered as core point, if at least a specified number (MinPoints) of neighbhoring points fall within Ɛ

- **Border Point**: A point that has fewer neighbors than MinPoints within Ɛ

- **Noise Point**: All other points

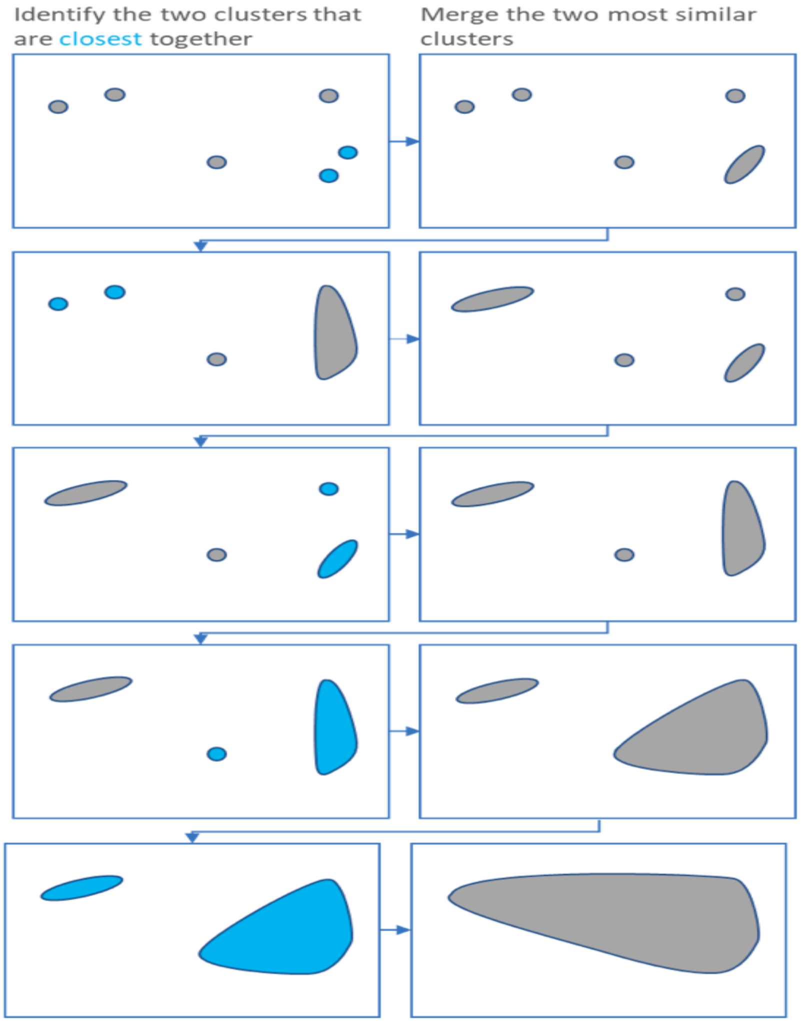
- **DBSCAN Algorithm**
  - Form a separate cluster for each core point or a connected group of core points.
  - Assign each border point to the cluster of its corresponding core point.



Hierarchical Clustering

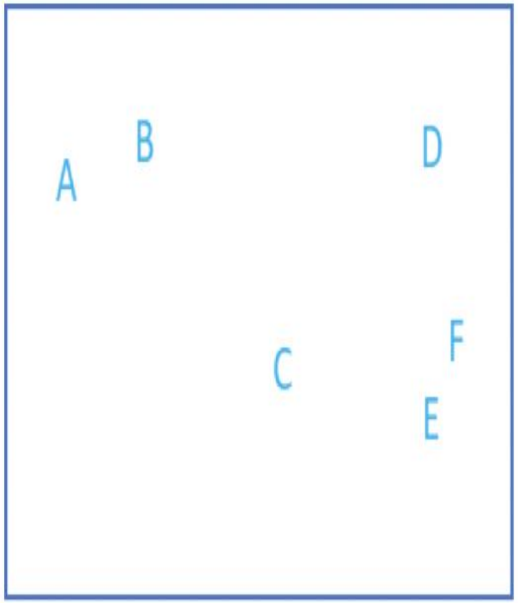
- It treats each observation as a cluster
- Identifies the two clusters that are closest, and merges them into one cluster
- The process is repeated until all the clusters are merged. This is called Agglomerative clustering (Bottom-up approach)

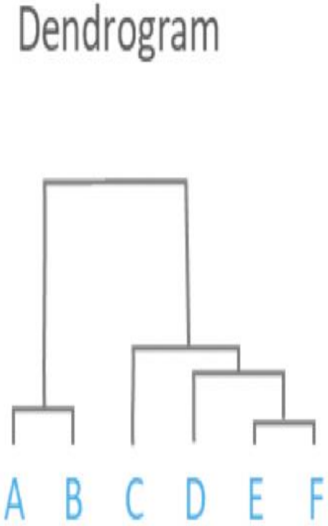
Hierarchical Clustering



Identify the two clusters that are closest together

Merge the two most similar clusters





Source: Displayr.com

Dimensionality Reduction

Dimensionality Reduction

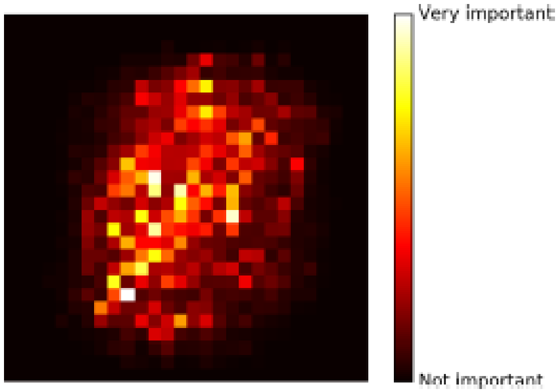
Curse of Dimensionality

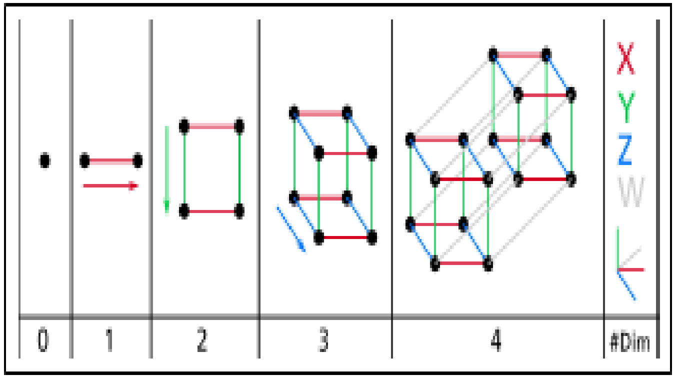
- Data Science and ML problems generally involve thousands or even million features for each training instance

- Having so many features can make training extremely slow, and may also lead to overfitting.

- Some of the many features may not be important like in pixel data

- Another problem with too many dimensions is in terms of data visualization. How many dimensions can we intuitively imagine or visualize???





Source: Aurelien Geron, Hands on ML

Dimensionality Reduction Techniques

Two main Approaches for dimensionality reduction

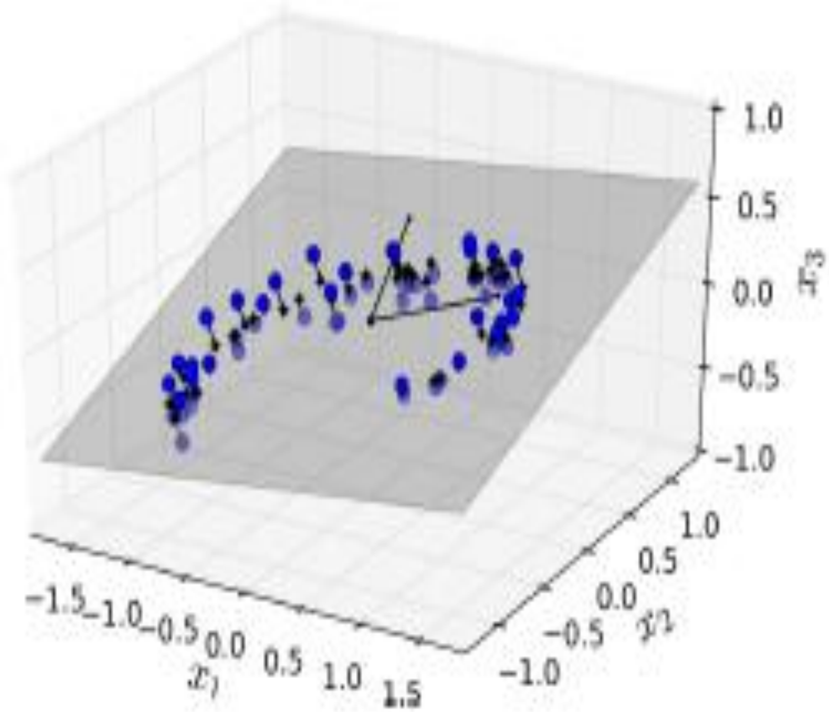
- Projection
  - Principal Component Analysis (PCA)
  - Incremental PCA, Randomized PCA
  - Kernel PCA

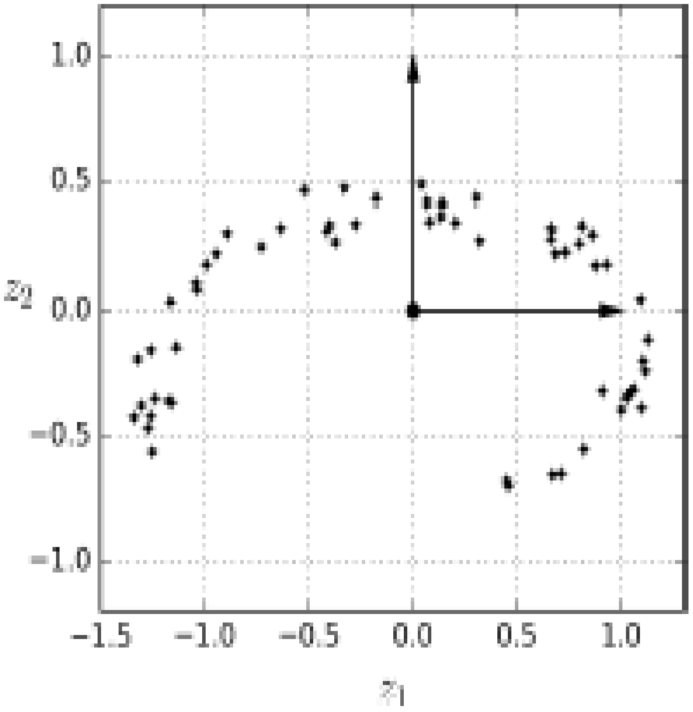
- Manifold Learning
  - Locally Linear Embedding (LLE)

Projection

- For several real world datasets, instances are not uniformly spread over all the dimensions. They typically scatter close to a lower dimensional subspace of the original higher dimensional space.

- Every training instance is then projected onto this lower dimensional subspace.



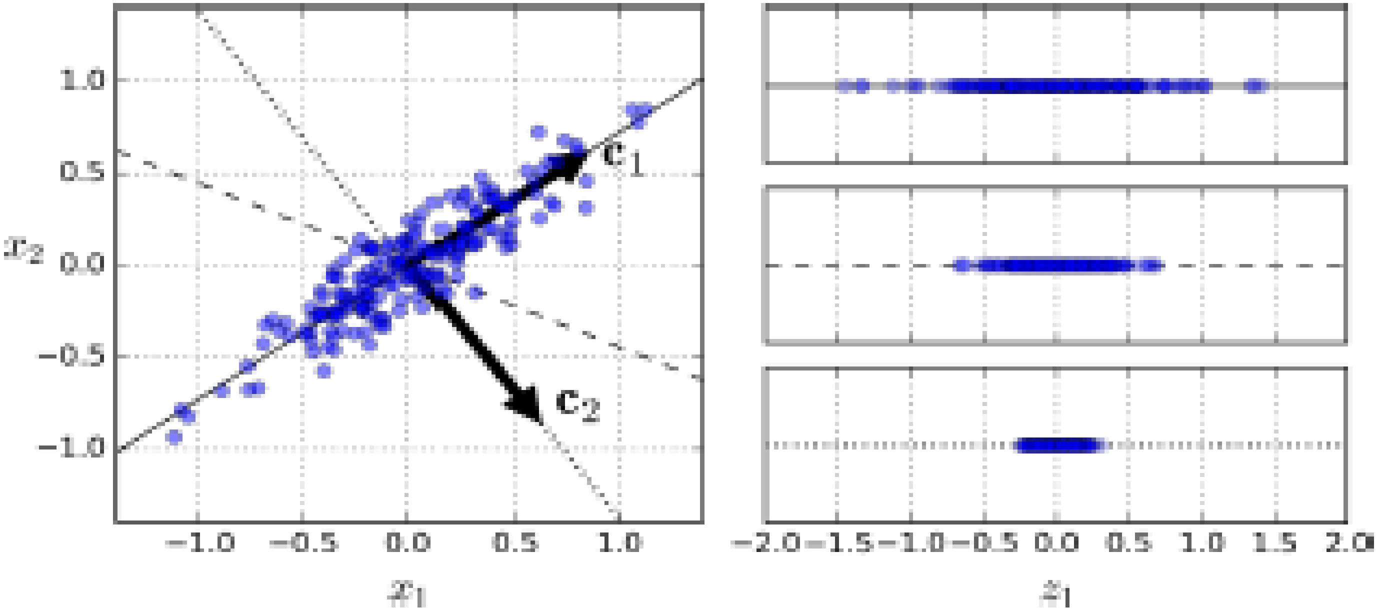


Principal Component Analysis

1. First an appropriate hyperplane (lower dimensional space) that lies closest to the data is identified. This will be the primary axis which preserves the maximum amount of variance.

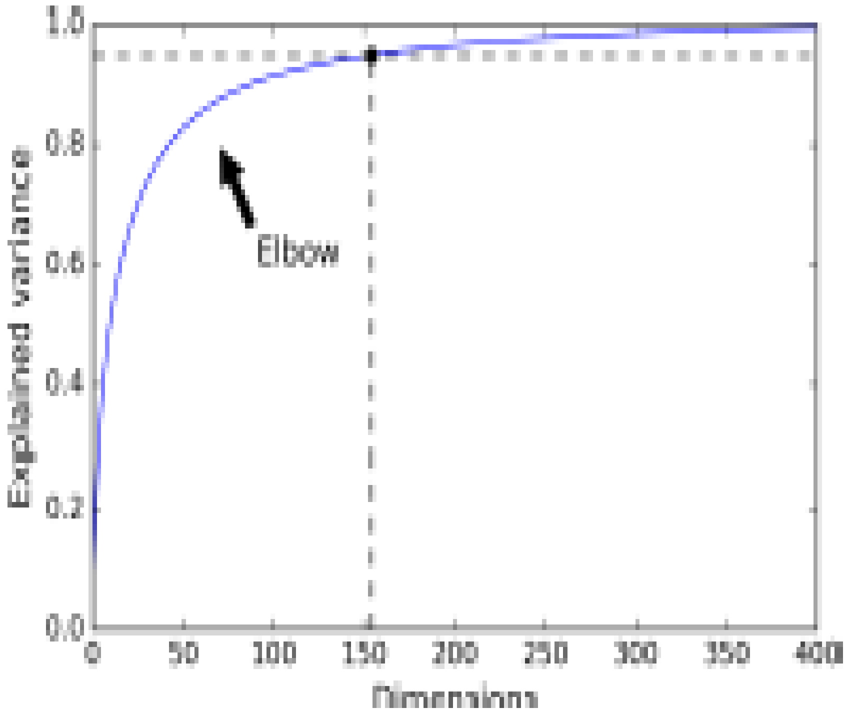
2. PCA will also find a second axis orthogonal to the first one, that accounts for the largest amount of remaining variance. This process can be repeated to find remaining Principal components (as many axes as dimensions)

3. Every training instance is then projected onto the hyperplane defined by the principal components.



Identifying the appropriate dimensions (or principal components)

- Explained variance ratio: Proportion of the variance in the data that lies along the axis of each principal component.



Other Projection Algorithms

- Incremental PCA
  - Implementation of general PCA algorithm requires the whole dataset to reside in memory
  - For large datasets or to apply PCA on the stream, Incremental PCA can be used, where training data is split into batches and each batch is passed to the algorithm incrementally.

- Randomized PCA
  - This uses a stochastic algorithm that rapidly finds an approximation to the first ‘p’ principal components. Faster than general PCA.

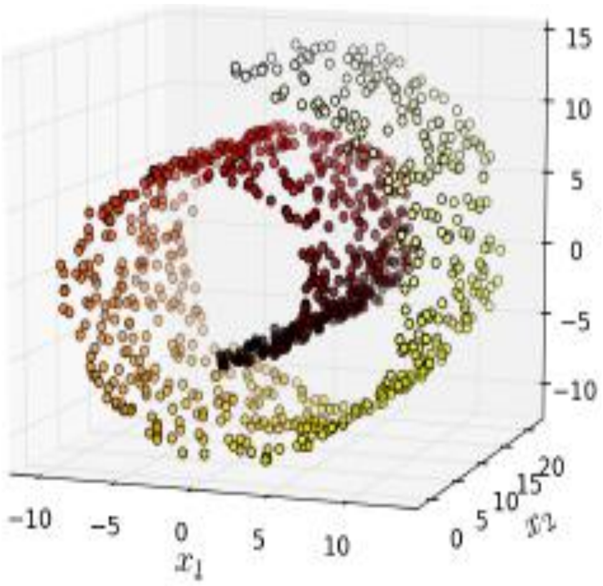
- Kernel PCA
  - Performs non-linear projections for dimensionality reduction
  - Good at preserving clusters of instances even after projection

Manifold Learning

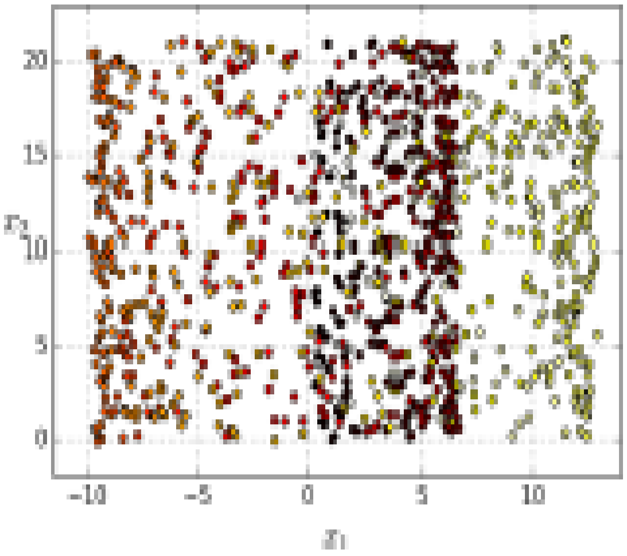
- Manifold: Shape that can be easily bent and twisted into a higher dimensional space.

- Manifold hypothesis: Most real-world high dimensional datasets lie close to lower dimensional manifold

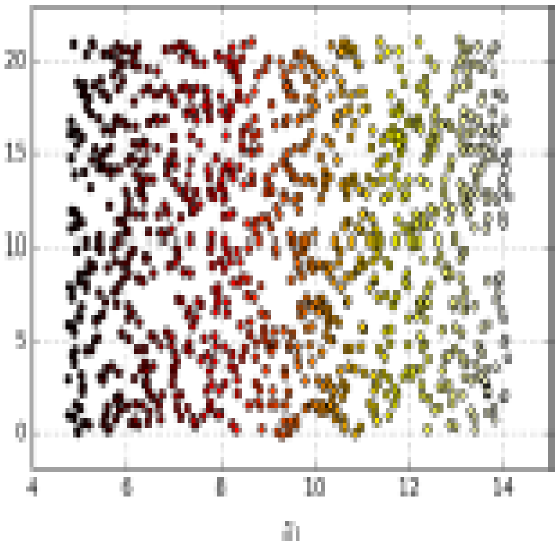
Original Swiss Roll Dataset



Projection



Manifold

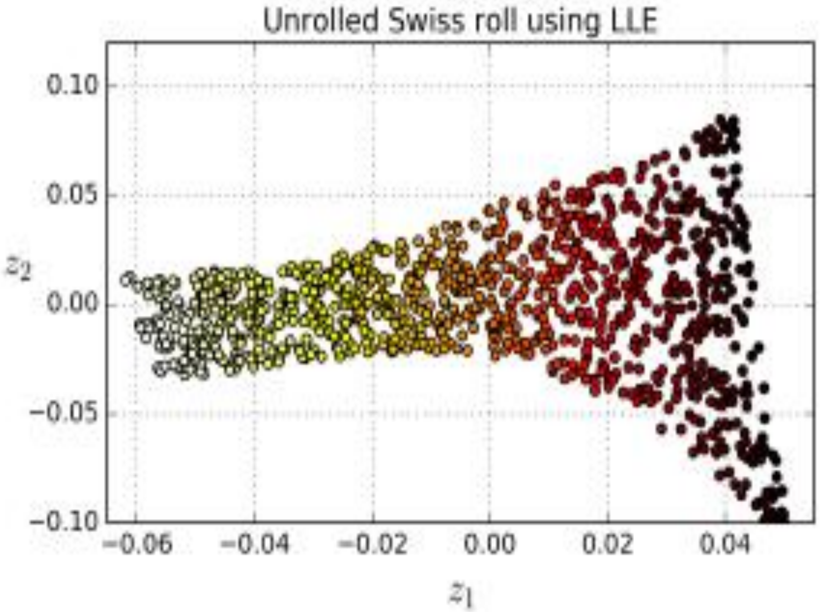


Locally Linear Embedding (LLE)

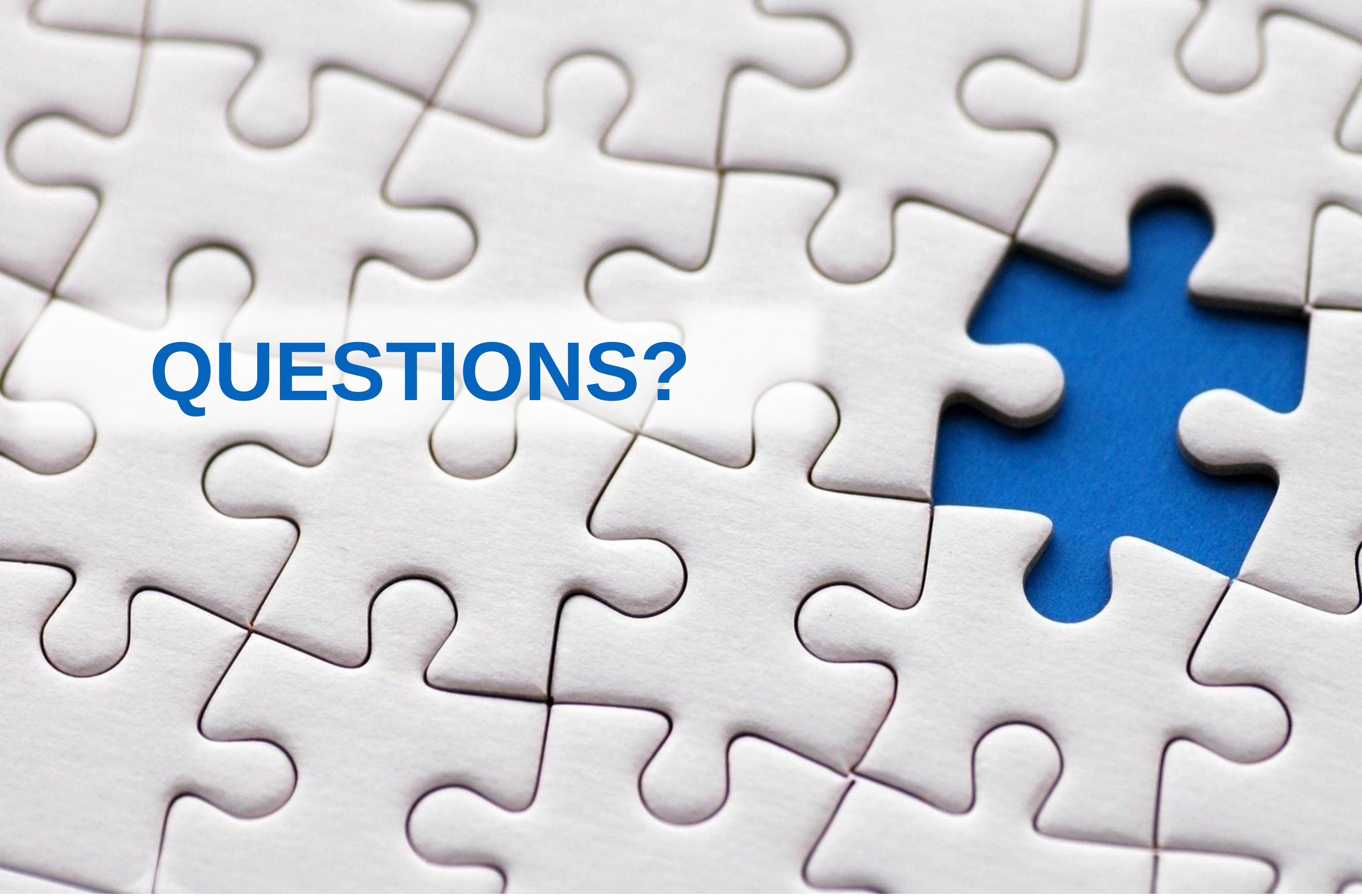
- Powerful non-linear dimensionality reduction technique based on Manifold Learning

- First identifies each instance’s closest neighbors and reconstructs the instance as a linear function of its closest neighbors.

- Algorithm then looks for a low-dimensional representation of the data set, where the relationships between the instance and its neighbors are preserved.



Source: Aurelien Geron, Hands on ML



QUESTIONS?