Inspire…Educate…Transform.

Methods & Algorithms in Machine Learning

Unsupervised Learning : Clustering

Dr. Rohit Lotlikar
rohit.lotlikar@insofe.edu.in
Mentor, INSOFE

INTRODUCTIONS: CLASS & MENTOR

Sections

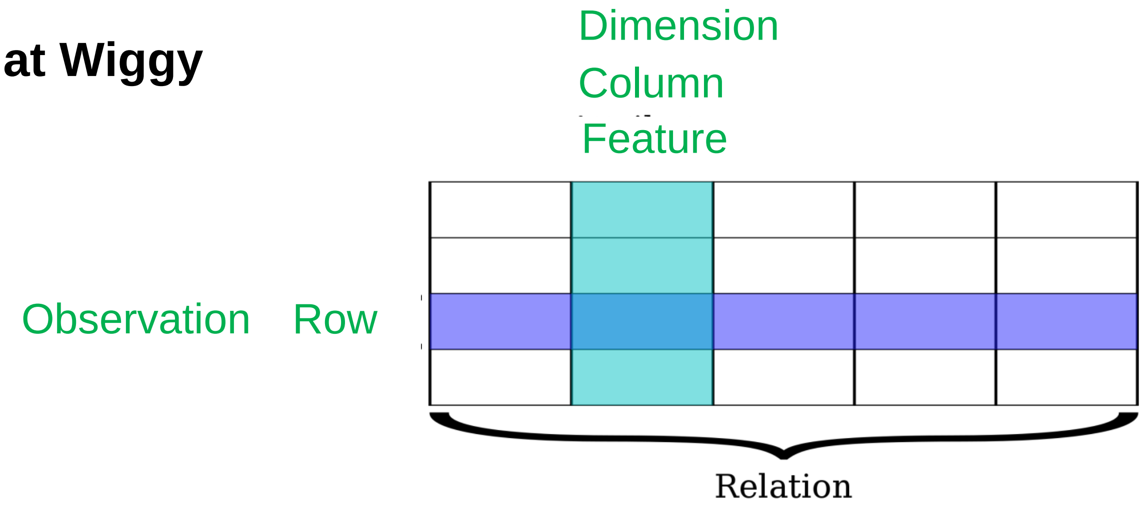
1. Similarity relationships in Data
2. The notion of distance
3. Clustering Framework
4. Clustering Algorithms (Partitioning Based)
5. Practical Considerations
6. Distance measures for non-numeric attributes
7. Hierarchical (Agglomerative) Clustering

SIMILARITY RELATIONSHIPS IN DATA

What does data look like?

	Num. orders in last 3 mo.	Avg. order amount	Lunch orders / Total orders
Customer 1	10	150	0.8
Customer 2	12	250	0.5
Customer 3	5	400	0.25
…	…	…	…
Customer 10201	7	130	0.75

Customer order data at Wiggy



Types of relationships/patterns in data we care about

Cause-Effect Relationships

What will happen next

Explain the effects / importance of ..

Helps us with :

1. What will be my sales next month?
2. Which product is the customer likely to want?

Helps us with:

1. What are the linkages between sales and various trade promo spends?
2. Is this image that of a cancerous cell or not?

Similarity Relationships

Identify natural groupings in the data (aka clustering or data driver segmentation). Helps us with:

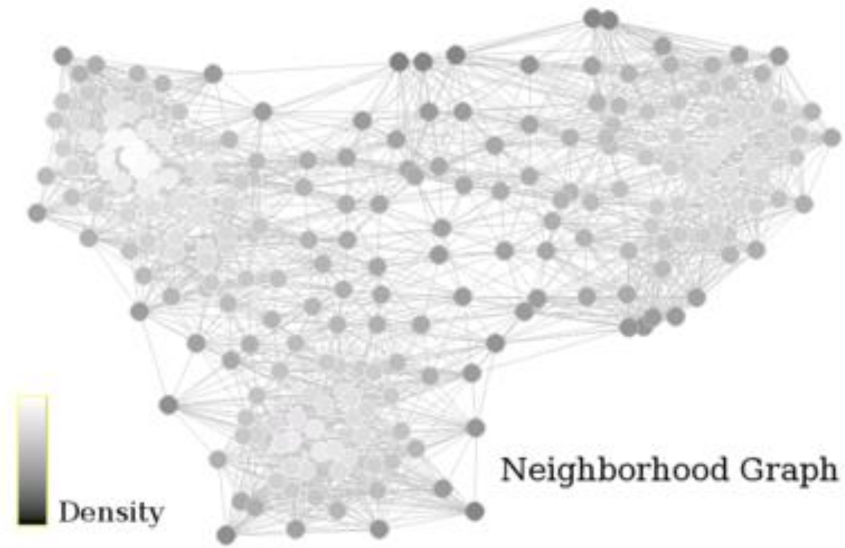
1. Retail Outlet Segmentation based on sku-mix
2. Anomaly Detection or Opportunity Identification

Similarity relationships – for Customer Segmentation

We want to split the set of customers into a small number of groups based on their purchase behavior

	Num. orders in last 3 mo.	Avg. order amount	Lunch orders / Total orders
Customer 1	10	150	0.8
Customer 2	12	250	0.5
Customer 3	5	400	0.25
…	…	…	…
Customer 10201	7	130	0.75

Customer orders at Wiggy's



Each dot represents one customer

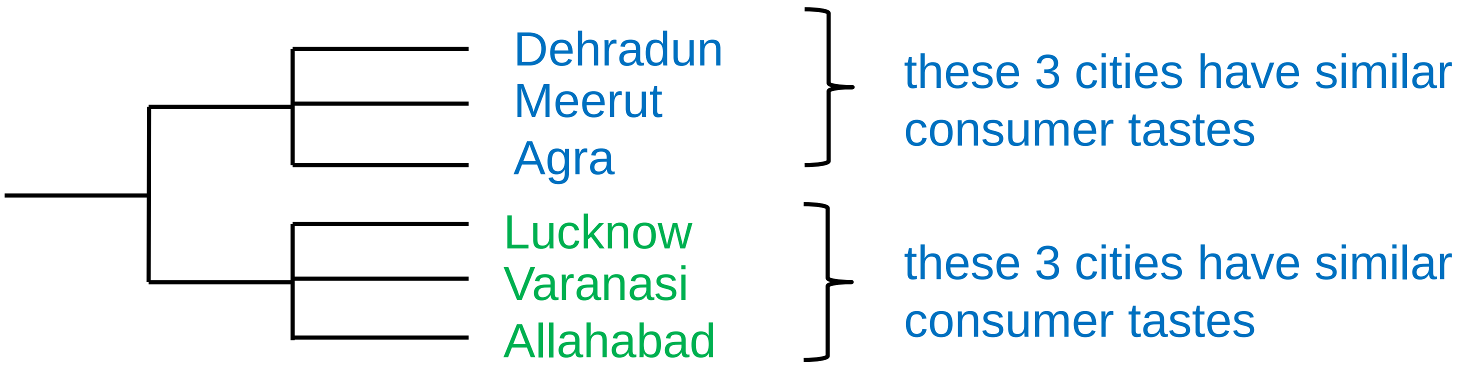
Typically the intent is to plan and analyze marketing promotions at a group level.

Similarity Relationships – for Geographic segmentation

Problem definition : A biscuit manufacturer wants to cluster cities into groups based on consumer preferences/tastes.

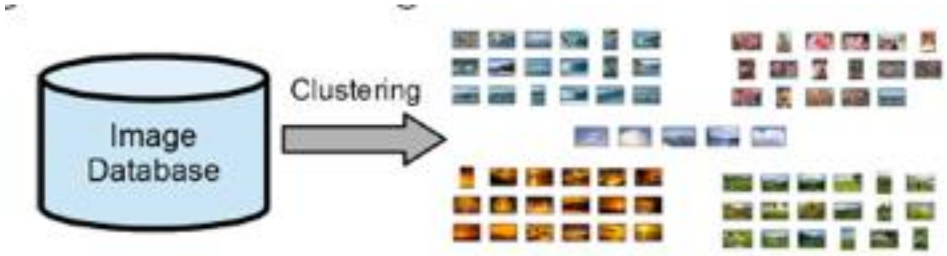
Table : **Data used for clustering – Sales in KG by town and biscuit**

id	Glucose	Simple Cream	Premium biscuits	Healthy biscuits
City 1	1330	311	240	42
City 2	870	233	231	36
…				

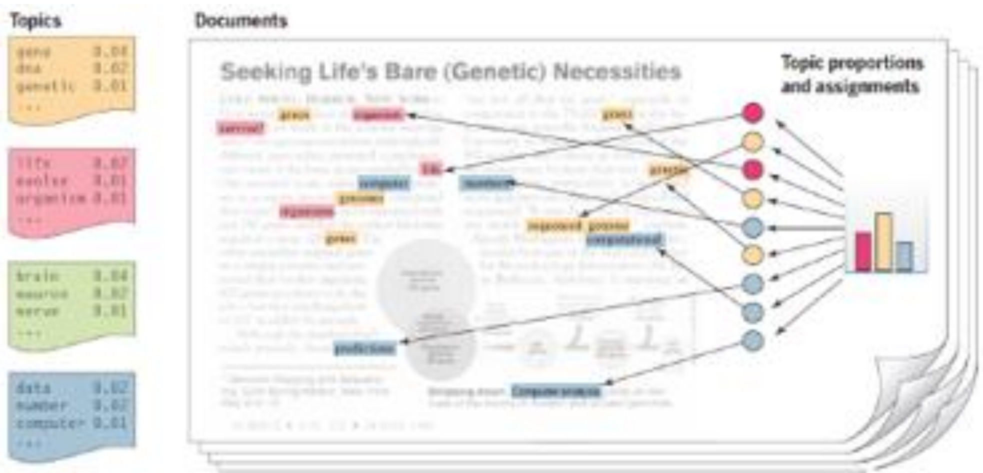


Similarity relationships of interest in unstructured data

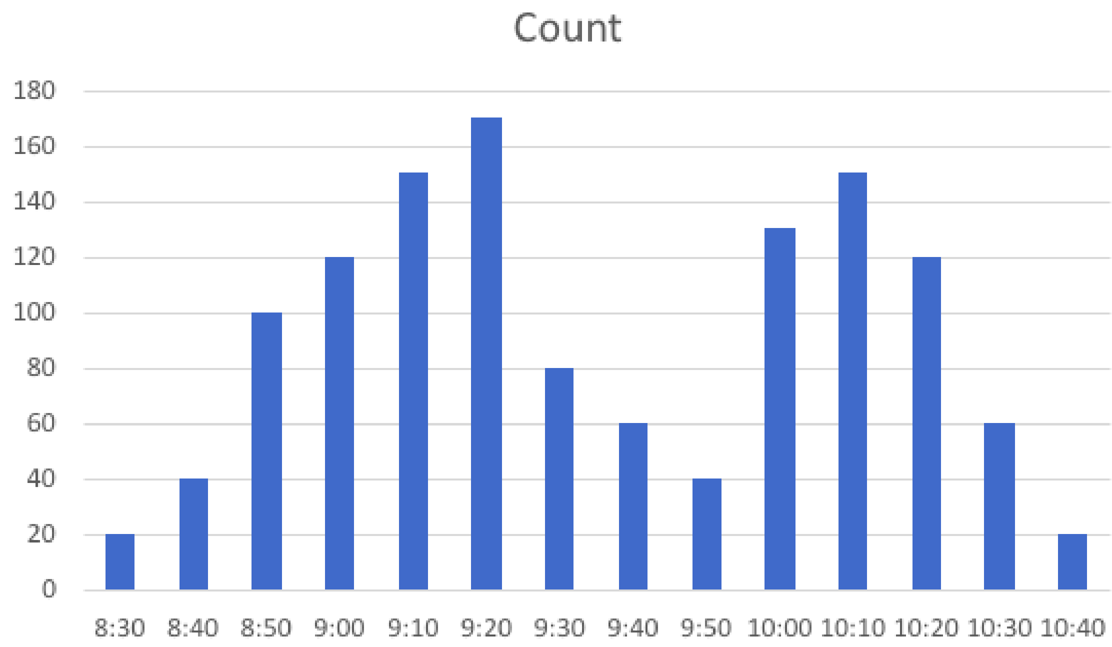
Clustering images based on image similarity



Clustering documents based on document similarity



Clustering in 1-D



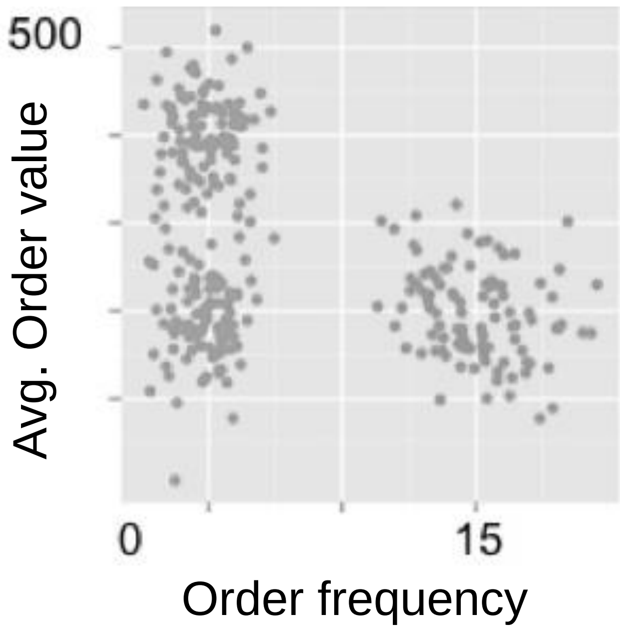
Time at which people are reaching office on an average

I want to provide a bus service with a choice of 2 arrival times, what should my choices be so that people arrive as close as possible to their current arrival time? (ignore capacity constraints)

Clustering in 2D

- <u>Input</u> : Set of observations

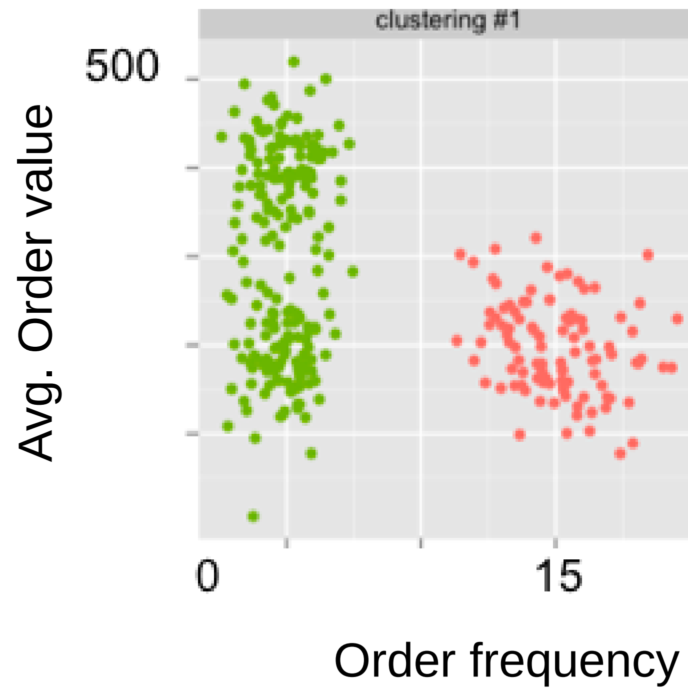
	Num. orders in last 3 mo.	Avg. order amount
Customer 1	10	150
Customer 2	12	250
Customer 3	5	400
…	…	…



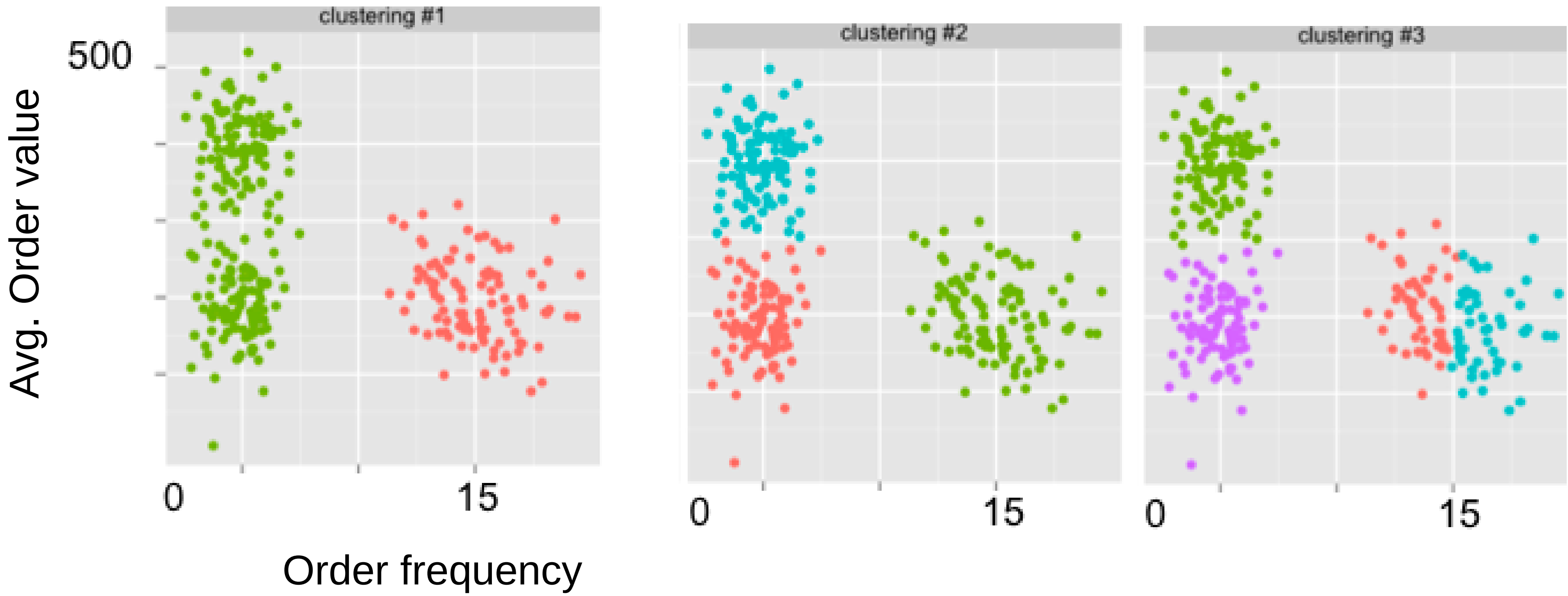
<u>Output of clustering</u> : The set of observations with a cluster label assigned to each observation.

Cluster1

Cluster2



The nature of clustering



Clustering is Subjective !

Clustering – higher dimensional data

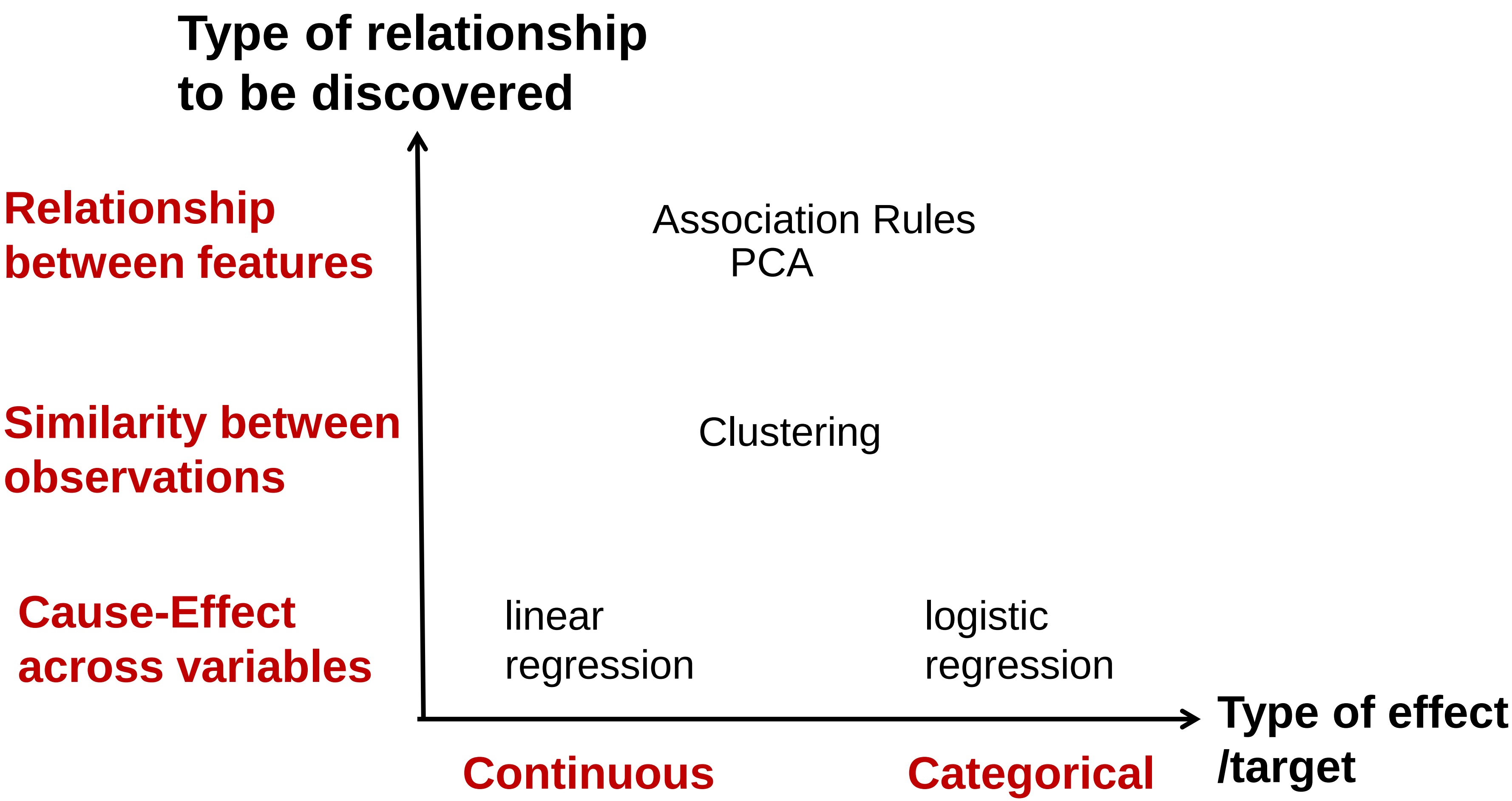
Cust id	Num. orders in last 3 mo.	Avg. order amount	Lunch/ Total	Days elapsed since last order	Weekday/ Weekend ratio	Loyalty member
1	10	150	0.3	2	2	Y
2	12	250	0.5	0	3	N
3	5	400	0.2	5	0.5	N
…	…	…				

?

- Large number of attributes
- Mix of numeric, ordinal and categorical attributes

We cannot visually cluster when we have a larger number of variables.
We need an algorithm

How clustering relates with other ML Techniques



How clustering relates with other ML Techniques

Machine Learning is about capturing the patterns in the data. Useful types of patterns:

- **Predicting one of the attributes** using the values of other attributes :
  - Attribute to be predicted is categorical -> Classification
  - Attribute to be predicted is numeric -> Regression
- Grouping of similar **rows/observations** -> Clustering
- Find patterns among **columns** :
  - Of the form x => y where columns are 0/1 -> Association Rules
  - Of the form of correlation between features -> PCA

NOTION OF DISTANCE BETWEEN OBSERVATIONS

Distance measure when all attributes are numeric

Euclidean distance:

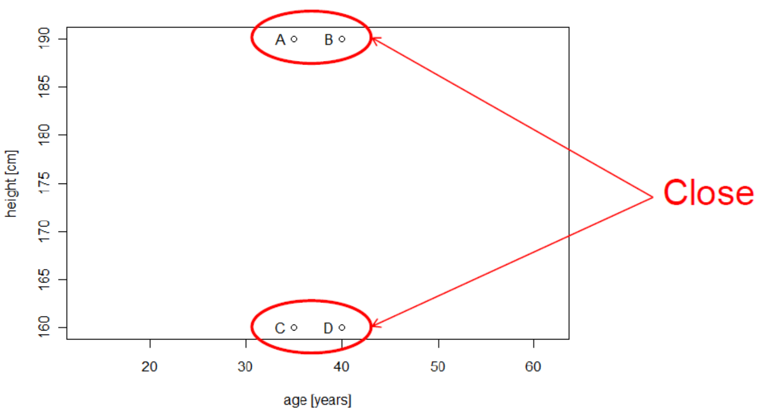
$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{ip} - x_{jp})^2}$$

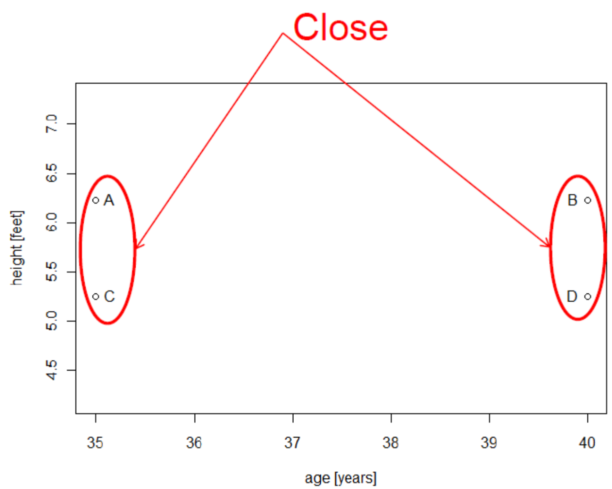
Scaling matters!

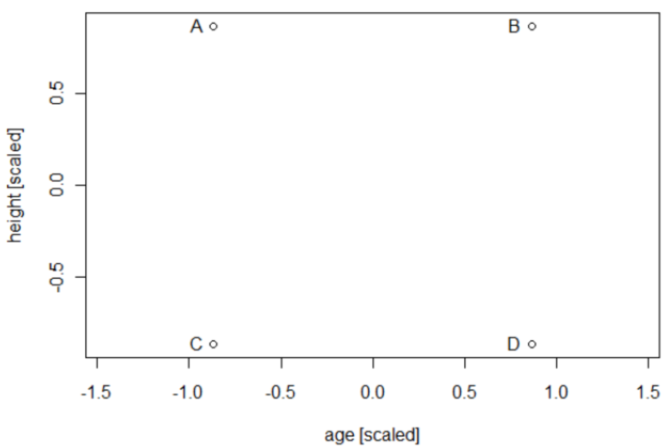
Person	Age [years]	Height [cm]
A	35	190
B	40	190
C	35	160
D	40	160

Person	Age [years]	Height [feet]
A	35	6.232
B	40	6.232
C	35	5.248
D	40	5.248

Person	Age [scaled]	Height [scaled]
A	-0.87	0.87
B	0.87	0.87
C	-0.87	-0.87
D	0.87	-0.87







To Scale or Not

- If variables are not scaled
  - Variable with largest range has most weight

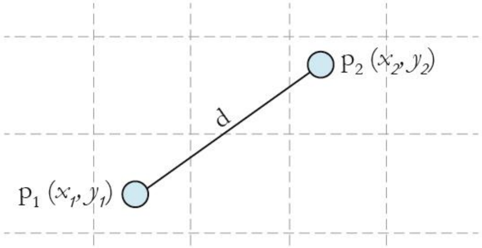
- If variables are scaled
  - Every variable gets equal weight
  - Similar alternative is re-weighing

$$d(i,j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \ldots + w_p(x_{ip} - x_{jp})^2}$$

- Perform scaling :
  - If variables measure different units (kg, meter, sec,…)
  - If you explicitly want to have equal weight for each variable
  - Default

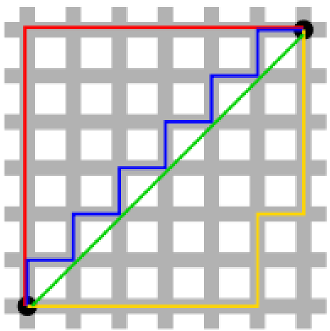
- Don't scale
  - if units are the same for all variables

Alternatives to Euclidean distance



Euclidean distance:

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



Manhattan distance:

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

It is important to choose the distance metric properly

TABLE : E-COMMERCE SITE CUSTOMER PURCHASES

Columns are sku's (items). Customer c1 has purchased items s4 and s10.

id	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14
c1	0	0	10	5	0	0	5	0	0	3	0	0	6	0
c2	0	1	2	0	0	0	1	0	0	0	0	2	2	0

What is an appropriate distance metric?

Cosine similarity - application

TABLE : CUSTOMER PURCHASES

Columns are sku's (items). Customer c1 has purchased items s4 and s10.

id	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14
c1	0	0	10	5	0	0	5	0	0	3	0	0	6	0
c2	0	1	2	0	0	0	1	0	0	0	0	2	2	0

If you care more about which products they buy without regard to quantities, use cosine similarity (cosine distance = 1 – cosine similarity)

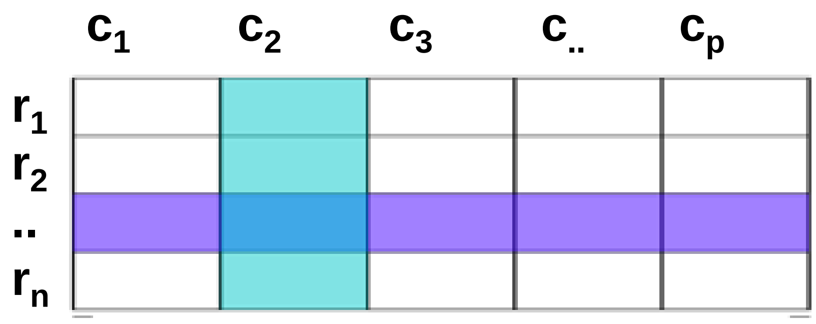
$$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

- The denominator normalizes the vectors to unit length.
- The numerator is the dot product i.e. overlap.
- This ratio is same as cosine of angle between two vectors

CLUSTERING FRAMEWORK

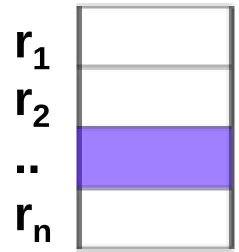
Clustering Framework

- Input
  - Data : n rows, p columns
  - A distance measure d()
  - k (Number of clusters)



Input Data

- Output
  - X partitioned into k-clusters;
  - Cluster-id for each observation (row)



cluster id (1.. k)

- Objective Goodness Measure
  - what would be a good measure?

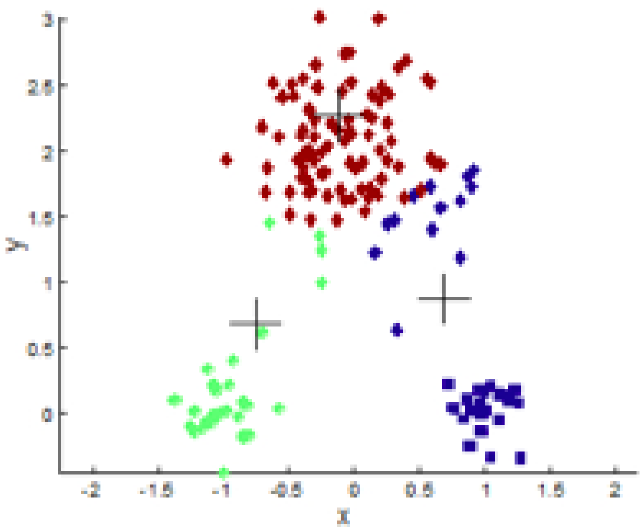
Objective goodness measure

- Minimize the sum of squares distances of each point to it’s cluster center

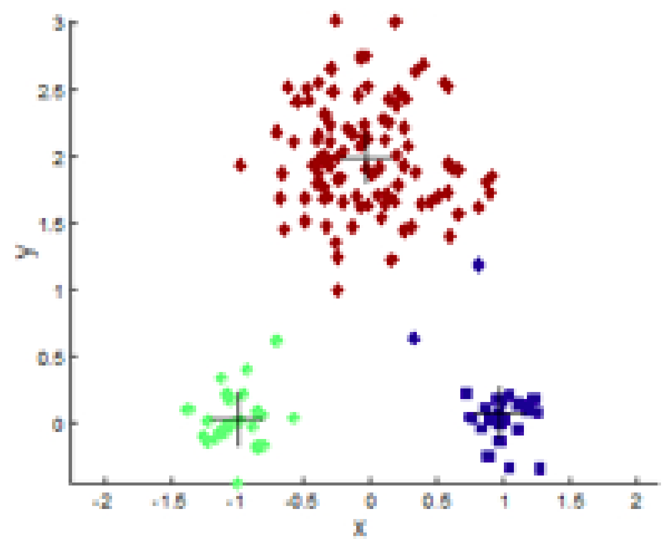
$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$

1. No (or minimum) re-assignments of data points to different clusters,

C_i is the jth cluster, $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j



Which is the better one?



Approaches for clustering

- **Bottom-up Agglomerative approach**:
  - Bottom-up hierarchical agglomeration.

- **Partitioning approach**:
  - Start with some partitions (splits) of the observations and iteratively refine the partition.
  - can be hierarchical

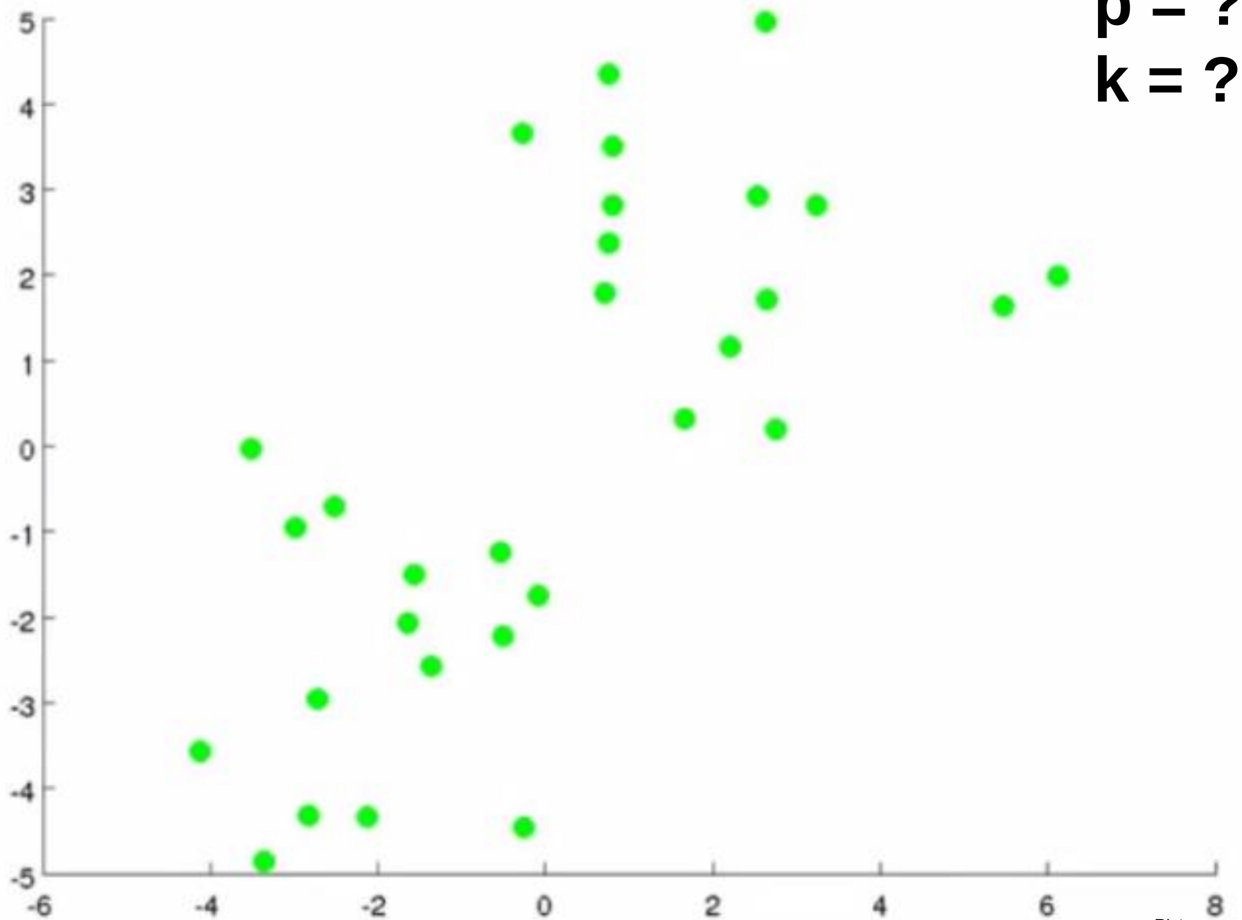
CLUSTERING ALGORITHMS (PARTITIONING BASED)

K-Means and K-Medoids

K-Means Clustering

- K-means is a partitional clustering algorithm as it partitions the given data into k clusters.
  - Each cluster has a cluster **center**, called **centroid**.
  - k is specified by the user

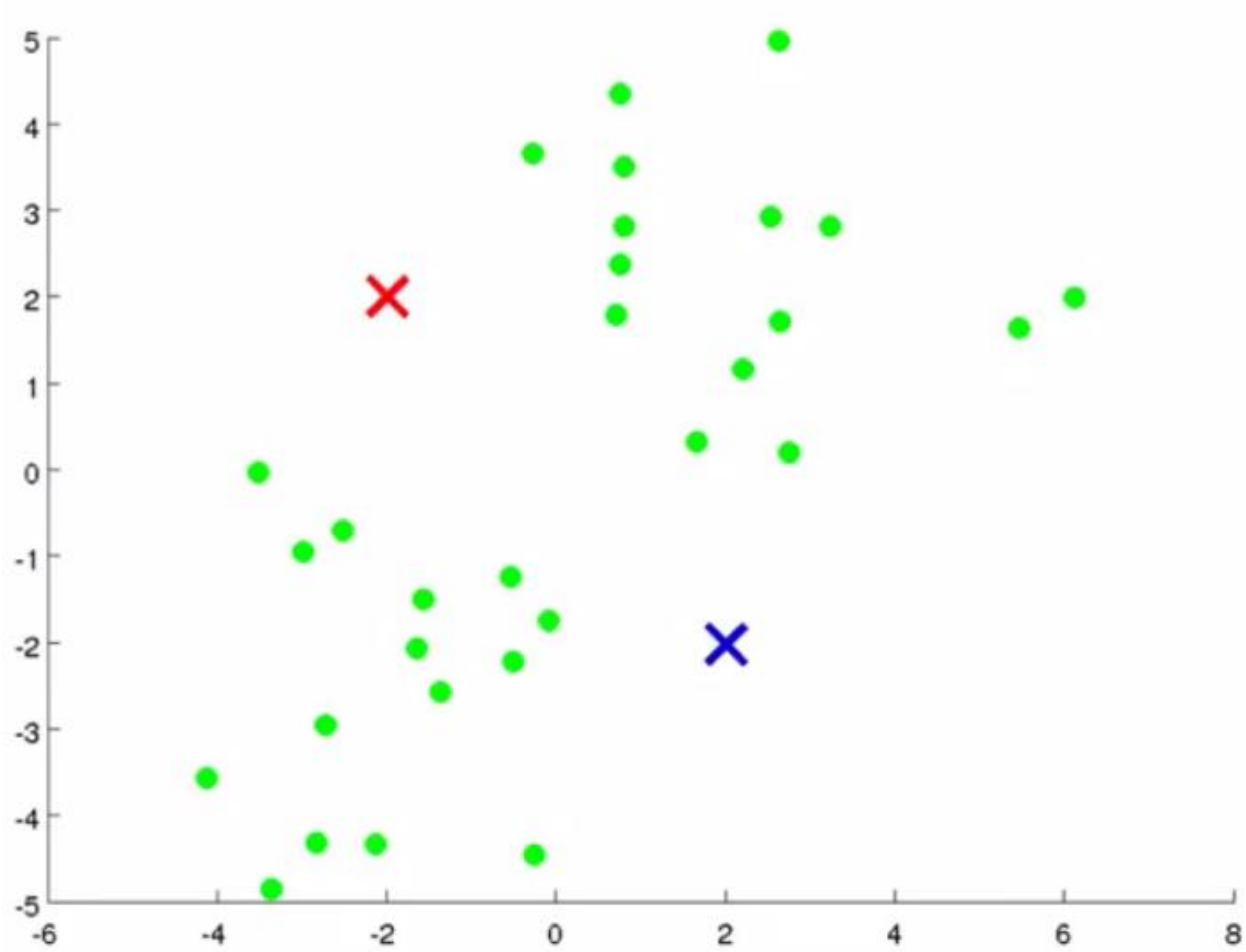
Input Data : Observations (rows) to be grouped into 2 clusters



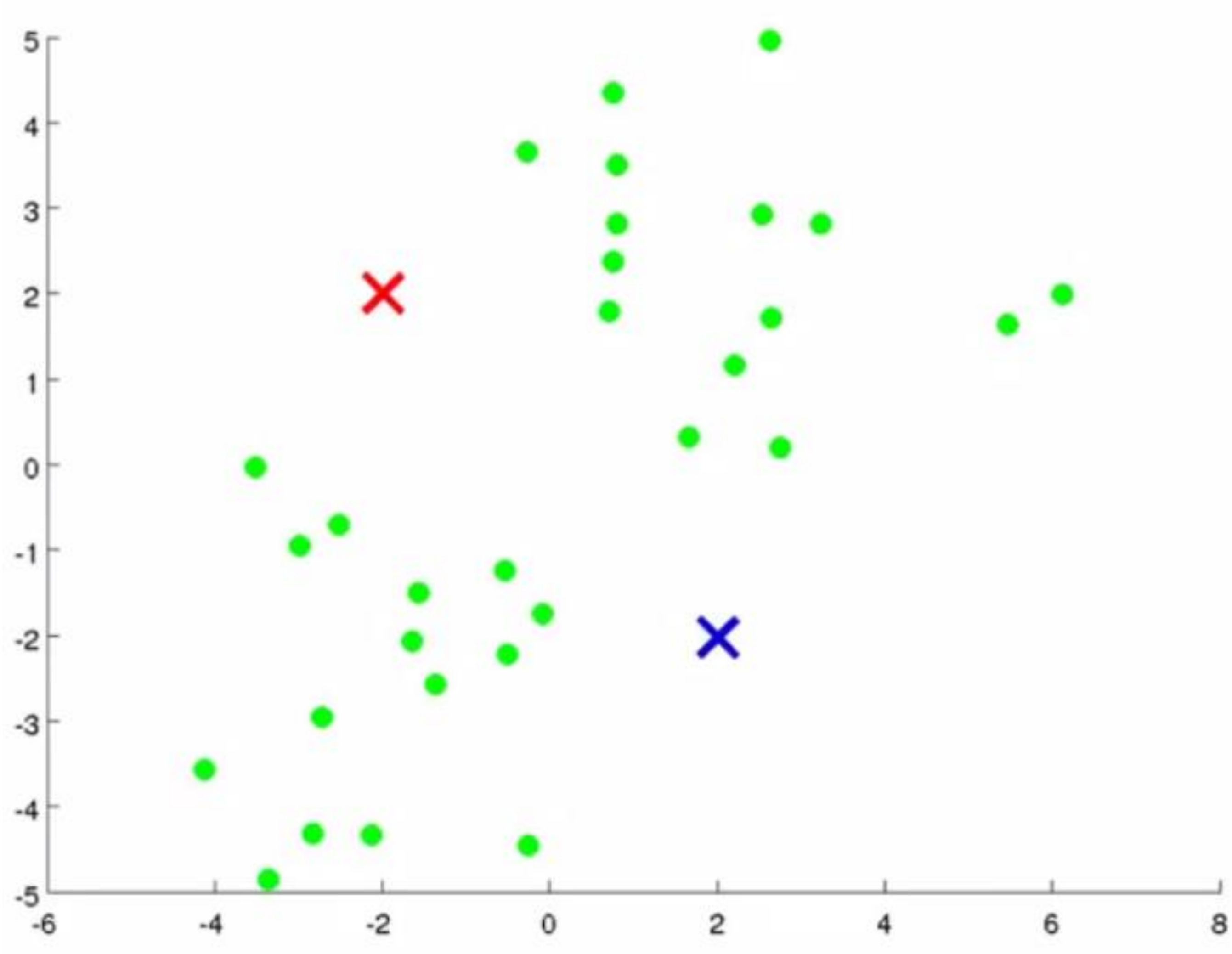
p = ?
k = ?

Pictures courtesy of Andrew Ng's course on Coursera.

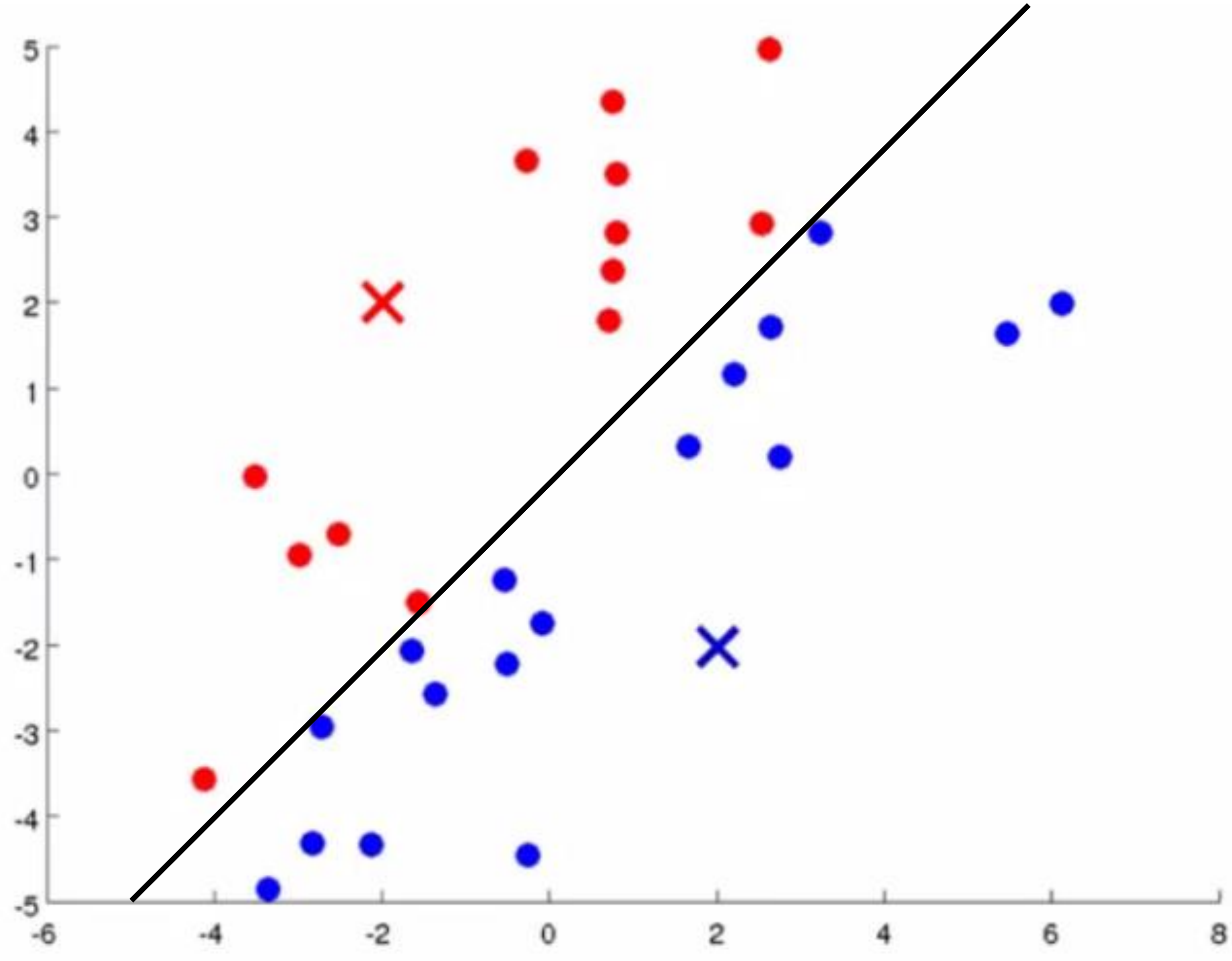
Step 1 : Randomly generate two cluster centers (aka centroids)



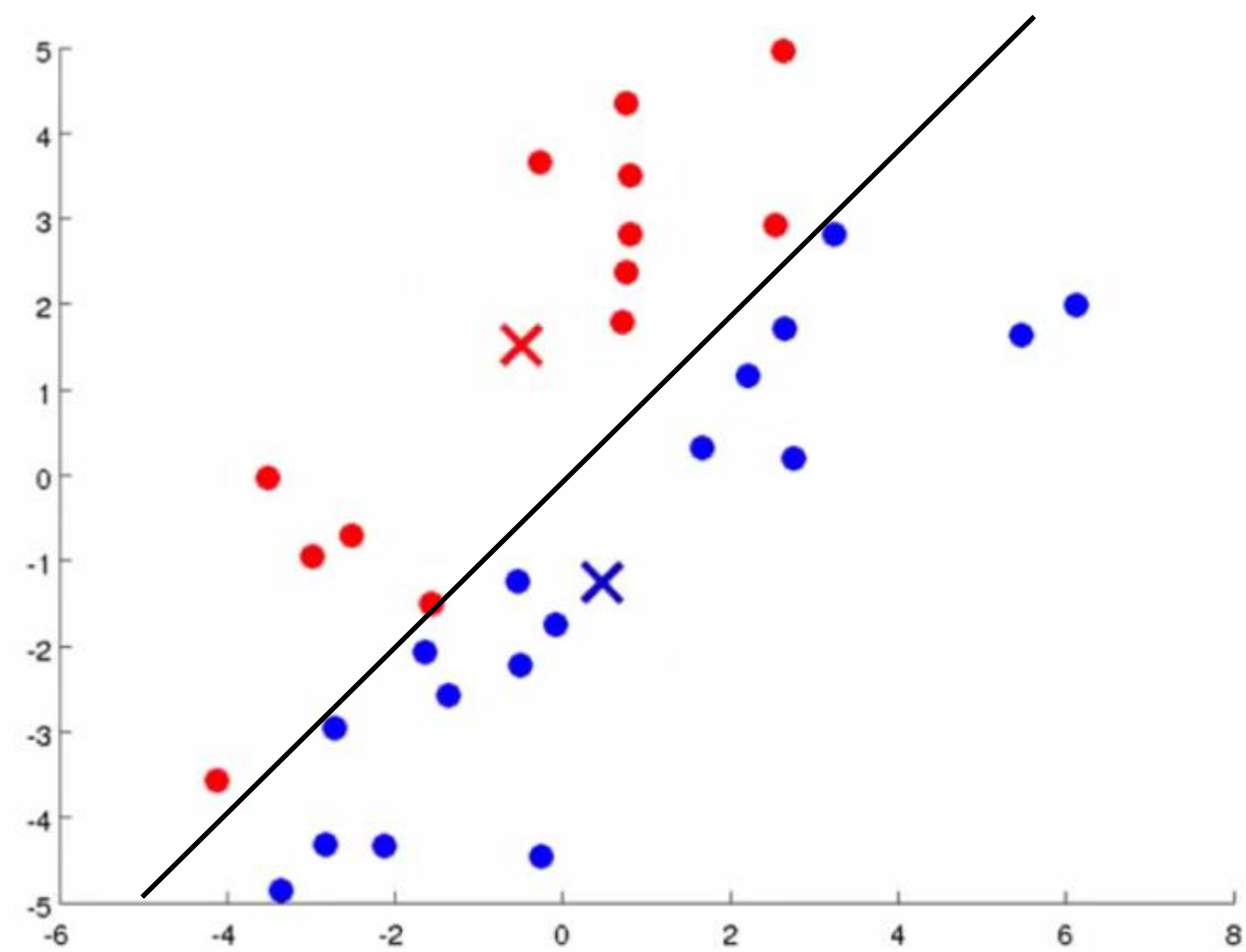
Step 2 : Assign each observation to the nearest centroid.



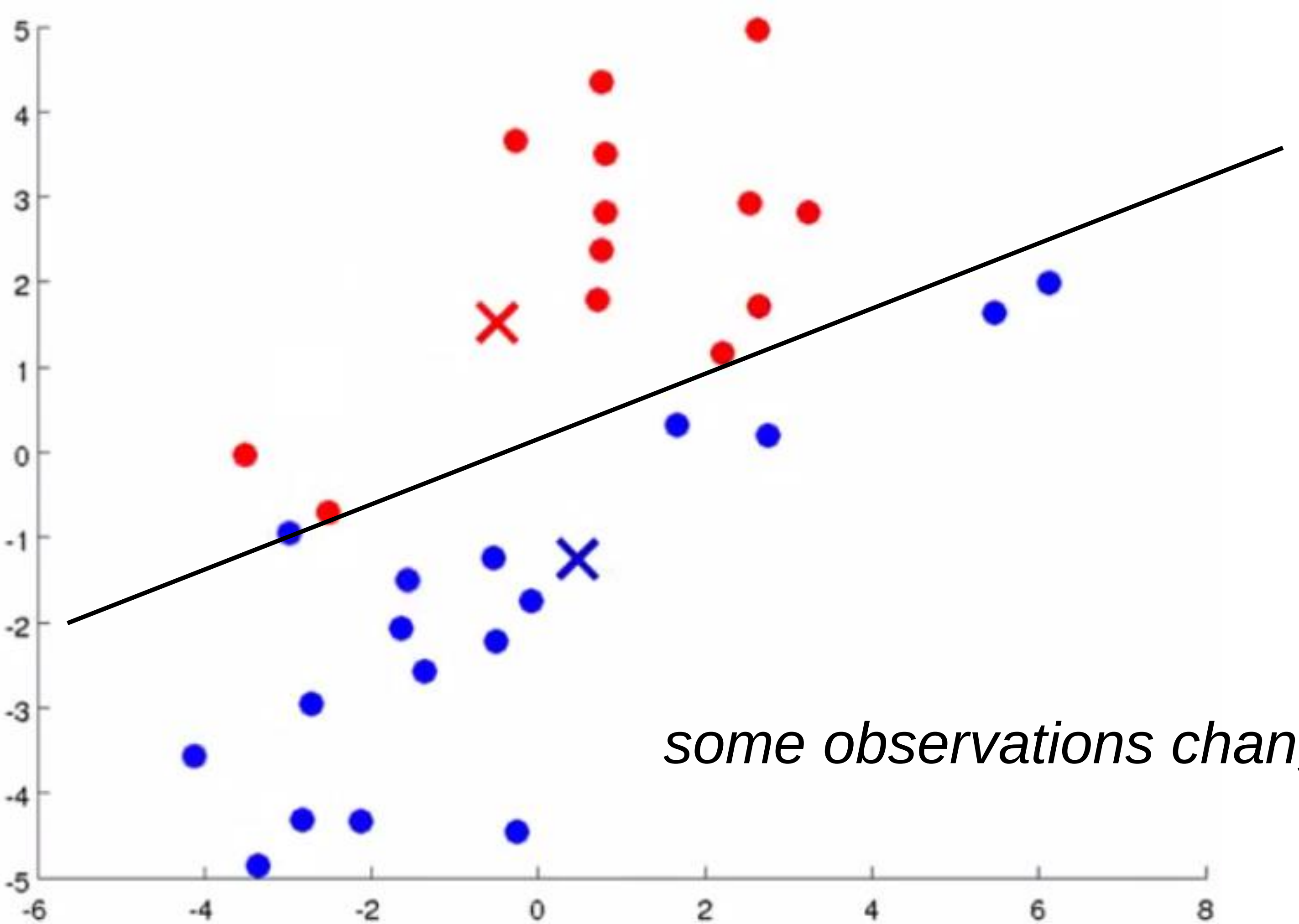
<u>**Step 2 completed**</u> **: Observations are colored as per the color of the closer cluster center**



<u>Step 3</u> :Compute the centroid of red observations and centroid of blue observations

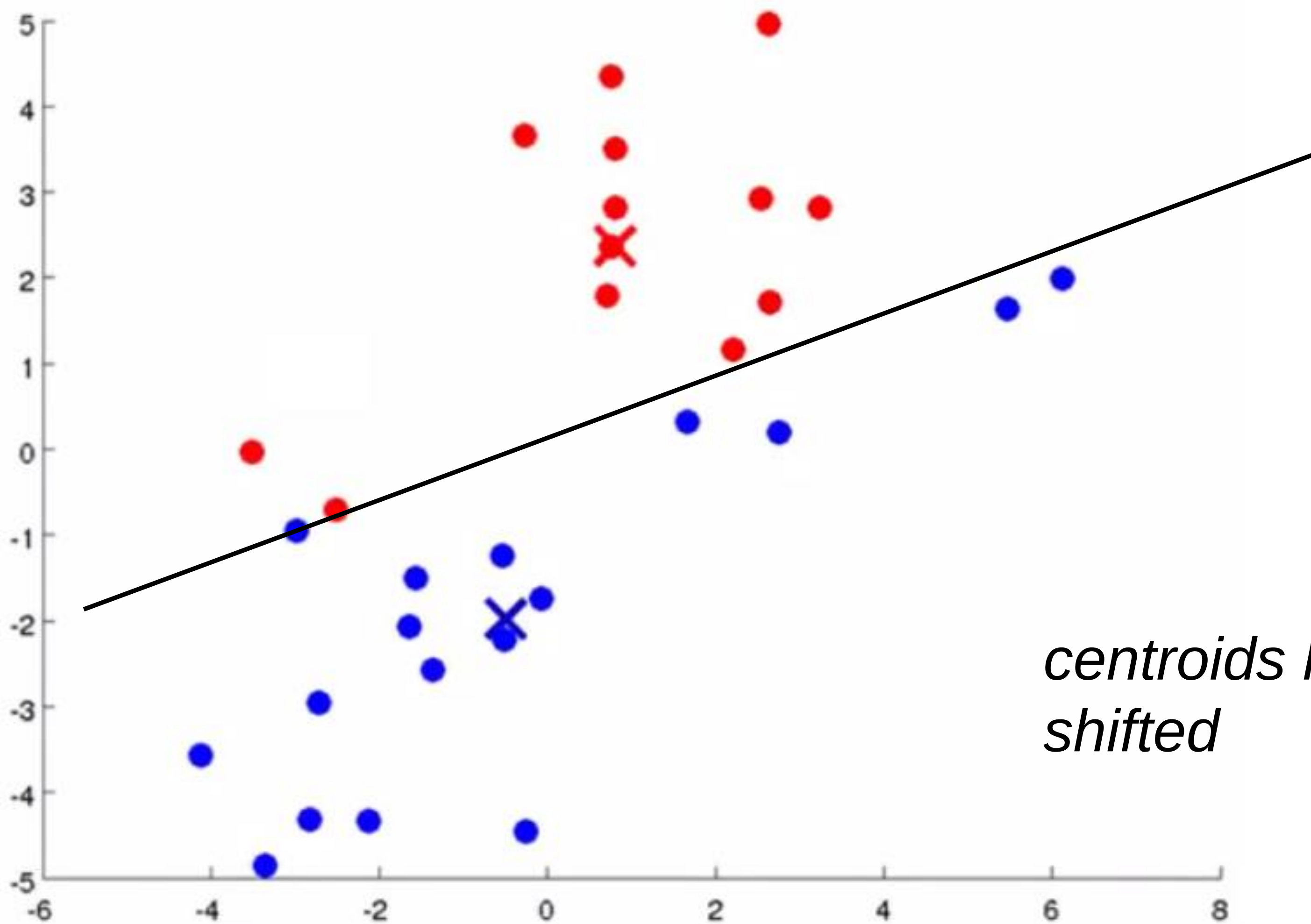


Repeat step 2 since centroids are updated



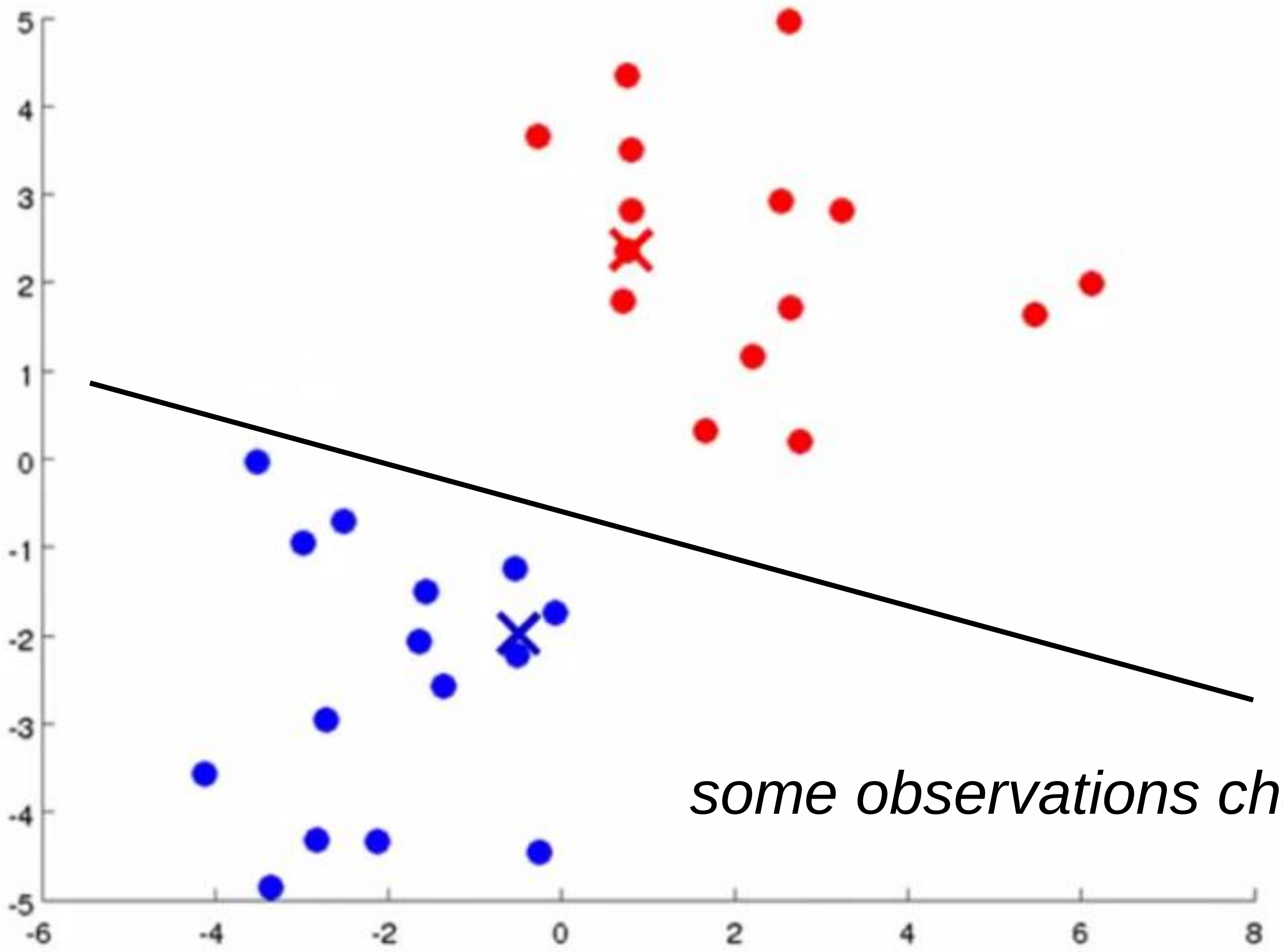
some observations change color

Repeat Step 3 (recompute centroids)



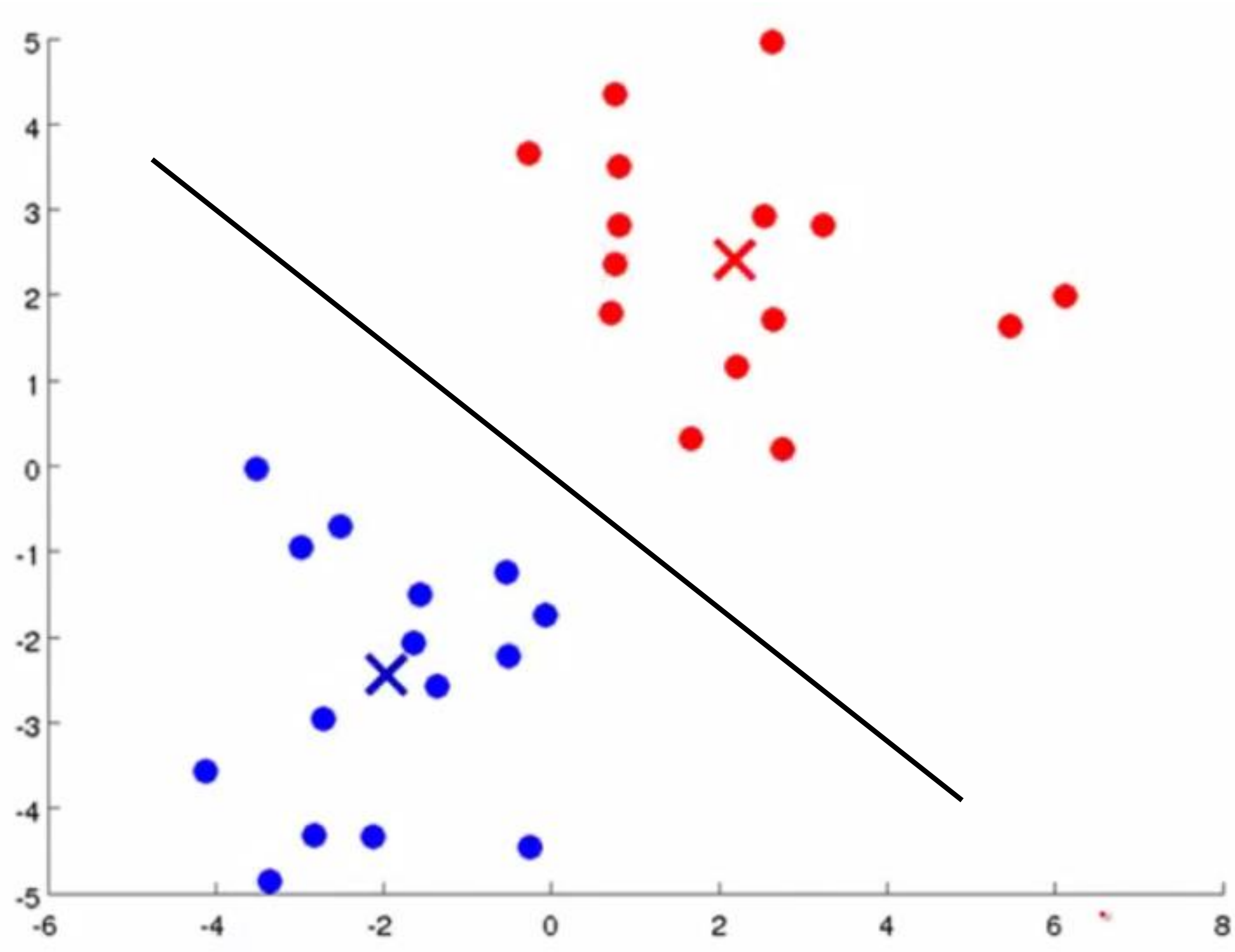
centroids have shifted

Repeat step 2 since centroids are updated



some observations change color

Repeat step 2 & 3 until cluster centers stabilize



K-Means Algorithm - Summary

- Given *k*, the *k-means* algorithm works as follows:
  1. Randomly choose *k* data points (seeds) to be the initial centroids, cluster centers
  2. Assign each data point to the closest centroid
  3. Re-compute the centroids using the current cluster memberships.
  4. If a convergence criterion is not met, or **if some clusters don't get any points**, go to 2.

Stopping/Convergence Criterion

1. No (or minimum) re-assignments of data points to different clusters,

2. No (or minimum) change of centroids, or

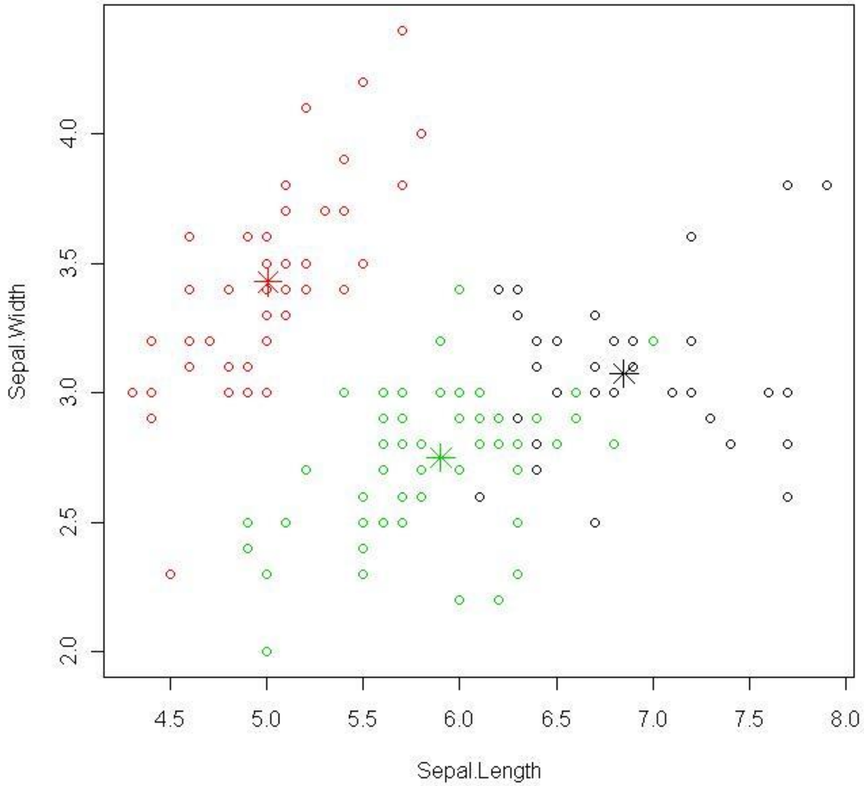
3. Minimum decrease in the **sum of squared error** (SSE),

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2 \qquad (1)$$

– C_i is the *j*th cluster, $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j

Example

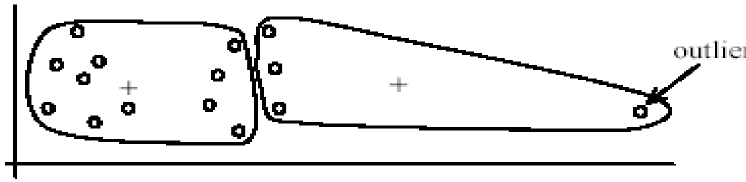
```
> (kc <- kmeans(newiris, 3))
K-means clustering with 3 clusters of sizes 38, 50, 62

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.006000    3.428000     1.462000    0.246000
3     5.901613    2.748387     4.393548    1.433871

Clustering vector:
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [30] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 1 3 3 3 3 3
 [59] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3
 [88] 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 1 1 1 3 1 1 1 1 1 1 3 3 1
[117] 1 1 1 3 1 3 1 3 1 1 3 3 1 1 1 1 1 3 1 1 1 1 3 1 1 1 3 1 1
[146] 1 3 1 1 3

Within cluster sum of squares by cluster:
[1] 23.87947 15.15100 39.82097

Available components:
[1] "cluster"  "centers"  "withinss" "size"
```



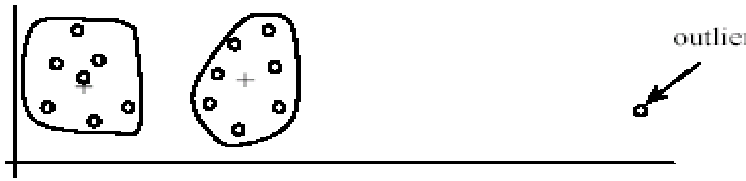

```
> table(iris$Species, kc$cluster)

              1  2  3
  setosa      0 50  0
  versicolor  2  0 48
  virginica  36  0 14
```

A Few Limitations of K-Means

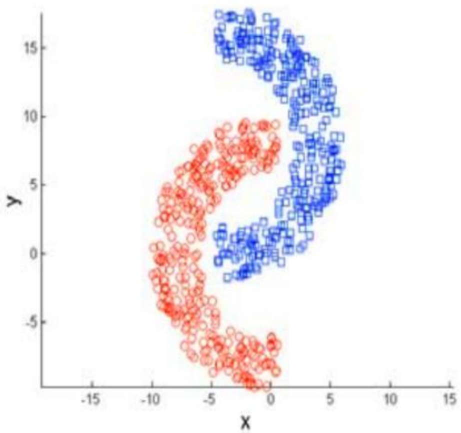
- Sensitivity to outliers in data
  - Detect and remove outliers before clustering
  - K-Medians is relatively more robust to outliers
- Cannot find arbitrary shaped clusters
  - May occur sometimes in nature and data

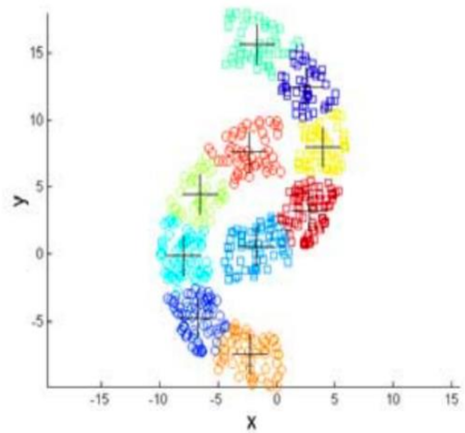


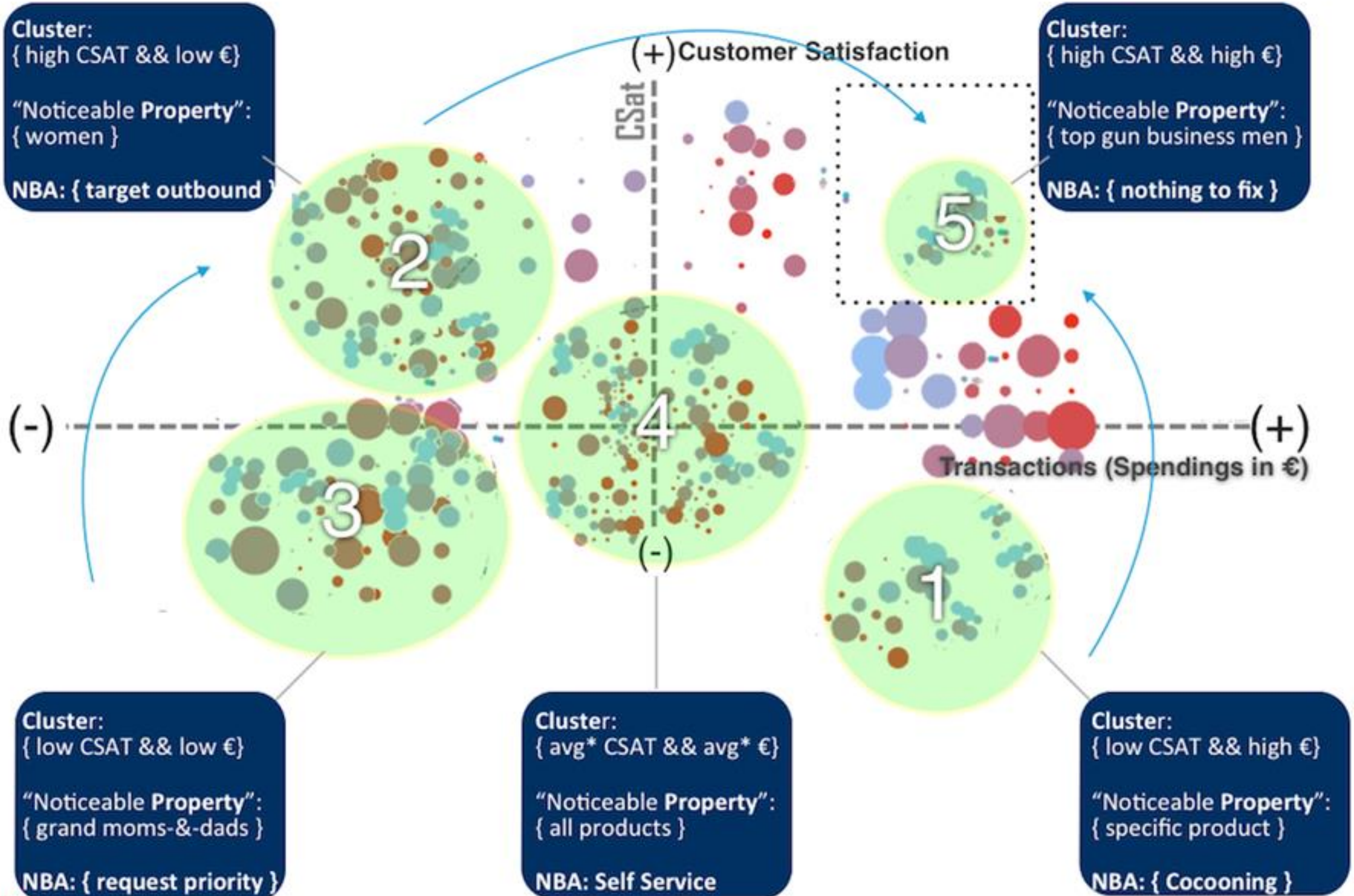
(A): Undesirable clusters



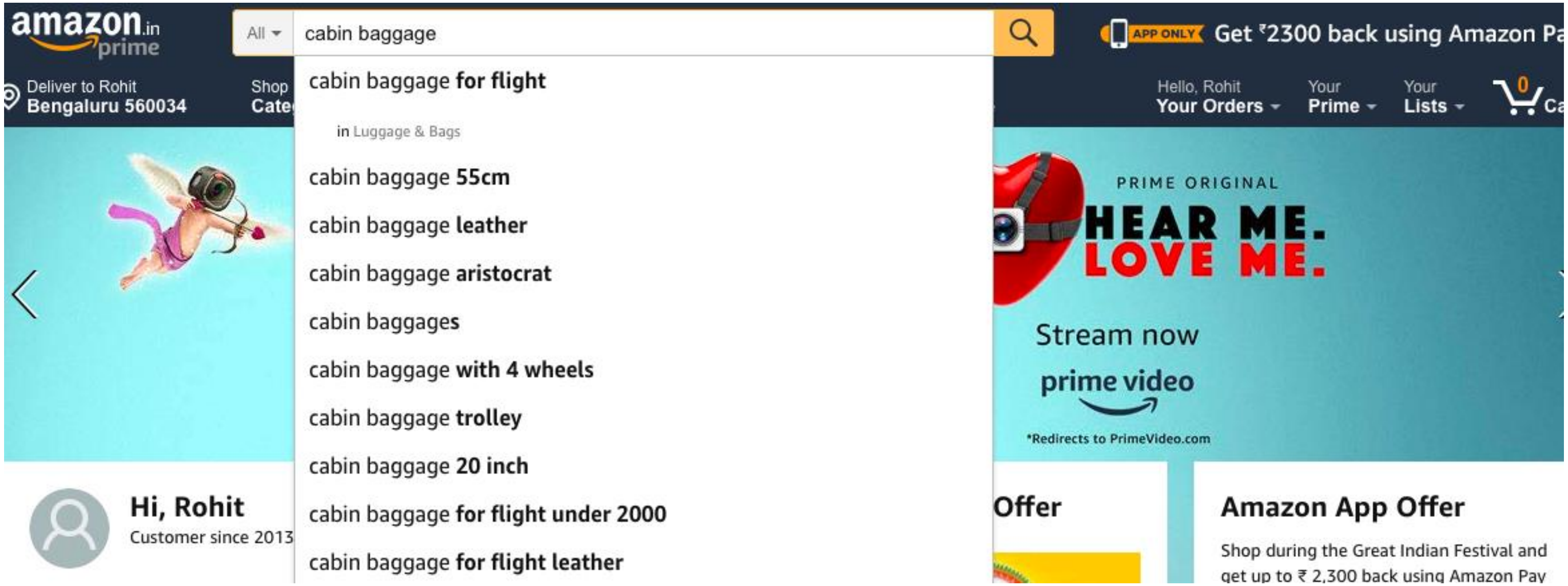
(B): Ideal clusters






Cluster:
{ high CSAT && low €}
"Noticeable Property":
{ women }
NBA: { target outbound }
(+) Customer Satisfaction
CSat
Cluster:
{ high CSAT && high €}
"Noticeable Property":
{ top gun business men }
NBA: { nothing to fix }
2
5
4
3
1
(-)
(+)
Transactions (Spendings in €)
(-)
Cluster:
{ low CSAT && low €}
"Noticeable Property":
{ grand moms-&-dads }
NBA: { request priority }
Cluster:
{ avg* CSAT && avg* €}
"Noticeable Property":
{ all products }
NBA: Self Service
Cluster:
{ low CSAT && high €}
"Noticeable Property":
{ specific product }
NBA: { Cocooning }

Clustering search queries



PRACTICAL CONSIDERATIONS

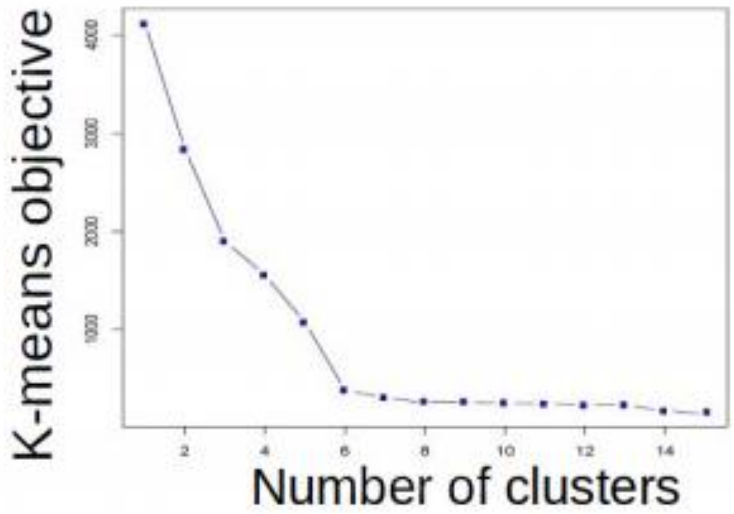
K-Means and K-Medoids

Stability Check of the Clusters

- To check the stability of the clusters take a random sample of 95% of records.
- Compute the clusters.
- If the clusters formed are very similar to the original, then the clusters are fine.

Choosing the value of K

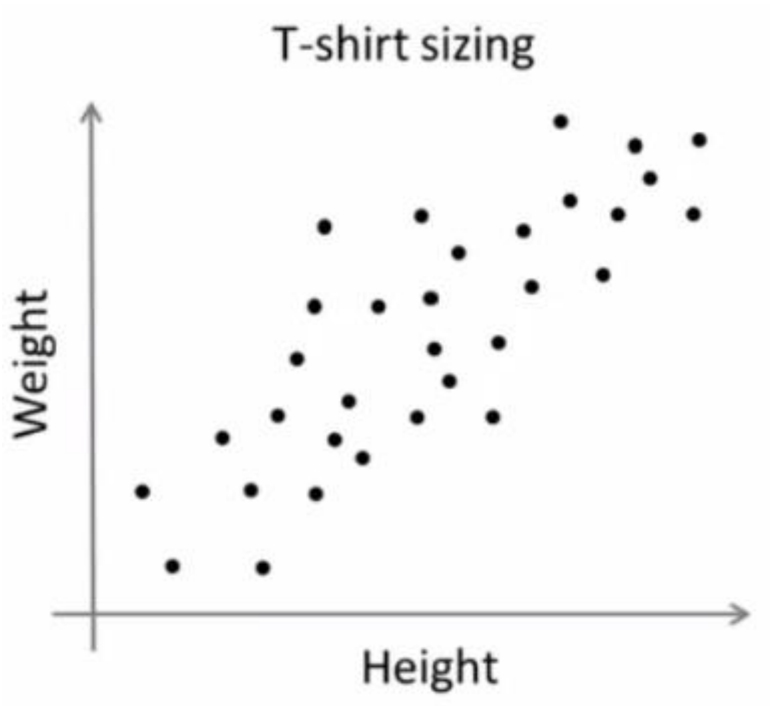
- One way to select K for the K-means algorithm is to try different values of K, plot the K-means objective versus K, and look at the "elbow-point"

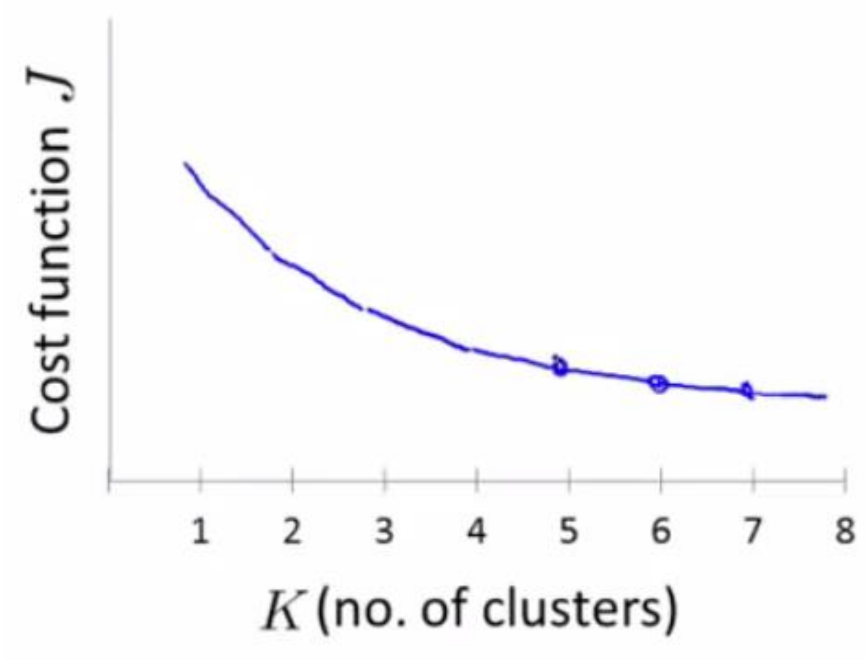


- For the above plot, $K = 6$ is the elbow point

Choosing the value of K

What happens if there are no distinct clusters?





K-Means vs K-Medoids

- The k-means algorithm is sensitive to outliers!
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.
- Uses L1 distance aka Manhattan distance

What is the problem with Medoids?

- More robust than k-means, in the presence of noise and outliers, because a medoid is less influenced by outliers or other extreme values than a mean

- Works efficiently for small data sets but does not **scale well** for large data sets.
  - $O(k(n\text{-}k)^2)$ for each iteration

where n is # of data,k is # of clusters

Large Data Sets

- Select a small % of data, run K-means or K-medoids
  - CLARA and CLARANS (Ng and Han 1994, 2002)
- Parallel and Efficient implementations of K-means / K-medoids

http://www.math.unipd.it/~dulli/corso04/ng94efficient.pdf
https://anuradhasrinivas.files.wordpress.com/2013/04/lesson8-clustering.pdf
http://www.vlfeat.org/overview/kmeans.html
http://repository.cmu.edu/cgi/viewcontent.cgi?article=2397&context=compsci
http://www.cs.ucsb.edu/~veronika/MAE/Global_Kernel_K-Means.pdf

DISTANCE MEASURES RE-VISITED

Distance measures for non-numeric attributes

Categorical Attributes

Option 1: Create dummies and use the same metric you use for numeric attributes

Attribute
Mysore
Delhi
Bangalore

→

Attribute	a1	a2	a3
Mysore	1	0	0
Delhi	0	1	0
Bangalore	0	0	1

Issue? Mysore and Bangalore are just as dissimilar as Delhi and Mysore

Categorical Attributes

Option 2: Use Hamming distance

		Data point j		
		1	0	
Data point i	1	a	b	$a+b$
	0	c	d	$c+d$
		$a+c$	$b+d$	$a+b+c+d$

$$Hamming\ distance = \frac{\#of\ dissimilar\ attributes}{\#of\ dissimilar + \#of\ similar} = \frac{b+c}{b+c+a+d}$$

Asymmetric Binary Attributes

- Asymmetric: if one of the states is more important or more valuable than the other.
  - By convention, state 1 represents the more important state, which is typically the rare or infrequent state.
  - Jaccard coefficient is a popular measure
  - We can have some variations, adding weights

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{b+c}{a+b+c}$$

Dissimilarity Between Binary Variables

- Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

  - Gender is a symmetric attribute
  - The remaining attributes are asymmetric binary
  - Let the values Y and P be set to 1, and the value N be set to 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Ordinal Variables

Employee performance rating scale

Performance Rating	Description	Rating guideline
1	Low performer	Bottom 10 %
2	Average performer	Next 50%
3	Above average performer	Next 30%
4	Exceptional performer	Top 10%

Need a custom distance metric

Best implemented as a Look-up table

What would be a suitable distance metric between the ratings ?

Look Up Matrix for Ordinal with 3 States

Performance Rating	Description	Rating guideline
1	Low performer	Bottom 10 %
2	Average performer	Next 50%
3	Above average performer	Next 30%
4	Exceptional performer	Top 10%

Rating	1	2	3	4
1	0	4	6	8
2	4	0	2	6
3	6	2	0	3
4	8	6	3	0

Mix of attribute types: Which distance measure to use?

- Gower Distance
  - Idea: Use distance measure between 0 and 1 for each feature *f*
  - Aggregate over features:

$$d(i,j) = \frac{1}{p}\sum_{i=1}^{p} d_{ij}^{(f)}$$

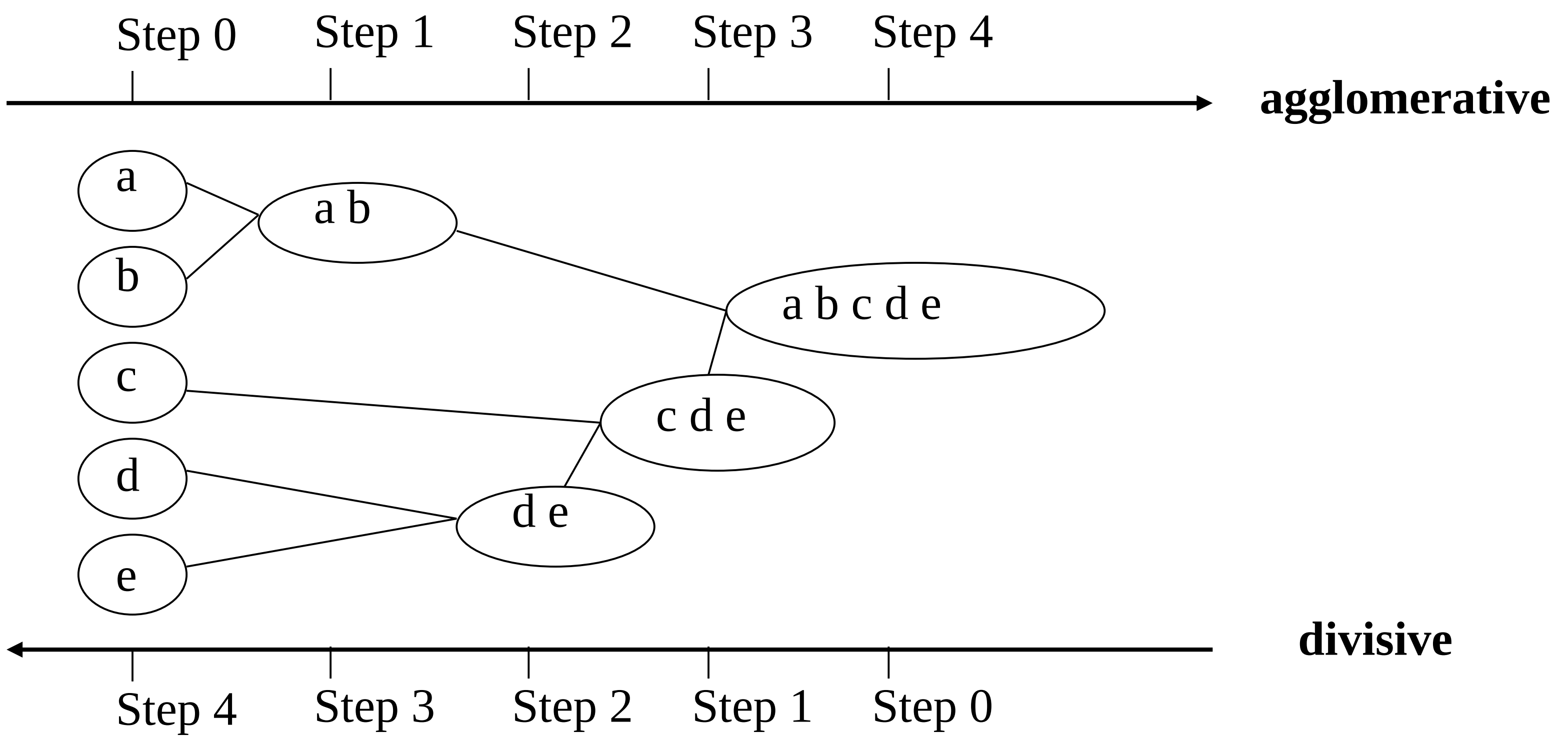
HIERARCHICAL (AGGLOMERATIVE) CLUSTERING

Distance measures for non-numeric attributes

BACK TO MODELS

Hierarchical Clustering

- Use distances between pairs of data points as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



Example of Agglomerative Clustering

	BOS	NY	DC	MIA	CHI	SEA	SF	LA	DEN
BOS	0	206	429	1504	963	2976	3095	2979	1949
NY	206	0	233	1308	802	2815	2934	2786	1771
DC	429	233	0	1075	671	2684	2799	2631	1616
MIA	1504	1308	1075	0	1329	3273	3053	2687	2037
CHI	963	802	671	1329	0	2013	2142	2054	996
SEA	2976	2815	2684	3273	2013	0	808	1131	1307
SF	3095	2934	2799	3053	2142	808	0	379	1235
LA	2979	2786	2631	2687	2054	1131	379	0	1059
DEN	1949	1771	1616	2037	996	1307	1235	1059	0

	BOS/NY	DC	MIA	CHI	SEA	SF	LA	DEN
BOS/NY	0	223	1308	802	2815	2934	2786	1771
DC	223	0	1075	671	2684	2799	2631	1616
MIA	1308	1075	0	1329	3273	3053	2687	2037
CHI	802	671	1329	0	2013	2142	2054	996
SEA	2815	2684	3273	2013	0	808	1131	1307
SF	2934	2799	3053	2142	808	0	379	1235
LA	2786	2631	2687	2054	1131	379	0	1059
DEN	1771	1616	2037	996	1307	1235	1059	0

	BOS/NY/DC	MIA	CHI	SEA	SF	LA	DEN
BOS/NY/DC	0	1075	671	2684	2799	2631	1616
MIA	1075	0	1329	3273	3053	2687	2037
CHI	671	1329	0	2013	2142	2054	996
SEA	2684	3273	2013	0	808	1131	1307
SF	2799	3053	2142	808	0	379	1235
LA	2631	2687	2054	1131	379	0	1059
DEN	1616	2037	996	1307	1235	1059	0

	BOS/ NY/DC	MIA	CHI	SEA	SF/LA	DEN
BOS/NY/DC	0	1075	671	2684	2631	1616
MIA	1075	0	1329	3273	2687	2037
CHI	671	1329	0	2013	2054	996
SEA	2684	3273	2013	0	808	1307
SF/LA	2631	2687	2054	808	0	1059
DEN	1616	2037	996	1307	1059	0

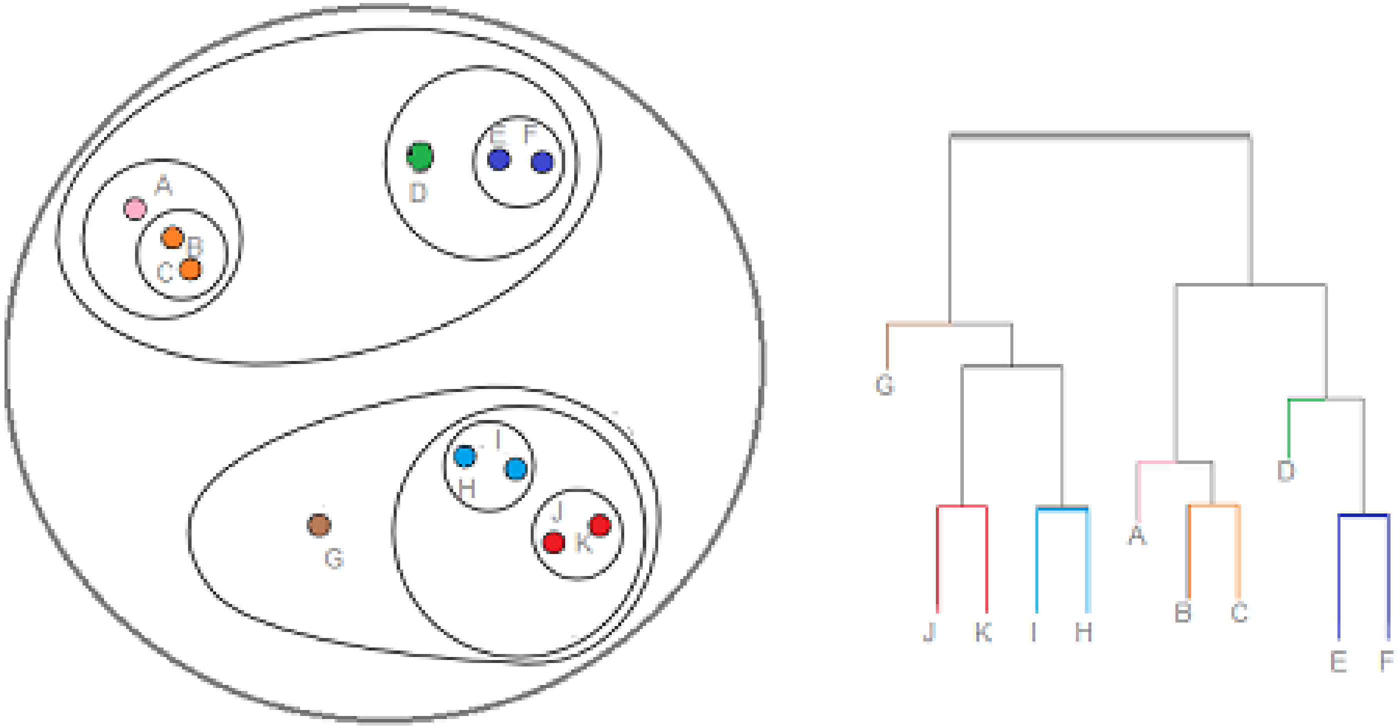
	BOS/NY/DC/ CHI	MIA	SEA	SF/LA	DEN
BOS/NY/DC/CHI	0	1075	2013	2054	996
MIA	1075	0	3273	2687	2037
SEA	2013	3273	0	808	1307
SF/LA	2054	2687	808	0	1059
DEN	996	2037	1307	1059	0

	BOS/NY/DC/CHI	MIA	SF/LA/SEA	DEN
BOS/NY/DC/CHI	0	1075	2013	996
MIA	1075	0	2687	2037
SF/LA/SEA	2054	2687	0	1059
DEN	996	2037	1059	0

	BOS/NY /DC/CHI/DEN	MIA	SF/LA/SEA
BOS/NY/DC/CHI/DEN	0	1075	1059
MIA	1075	0	2687
SF/LA/SEA	1059	2687	0

	BOS/NY /DC/CHI /DEN/SF /LA/SEA	MIA
BOS/NY/DC/CHI/DEN/SF/LA/SEA	0	1075
MIA	1075	0

Hierarchical Clustering



Decomposes data objects into several levels of nested partitioning (tree of clusters).

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.

K-Means vs. Hierarchical

- Flat clustering produces a single partitioning
- Flat clustering needs the number of clusters to be specified
- Flat clustering is usually more efficient run-time wise

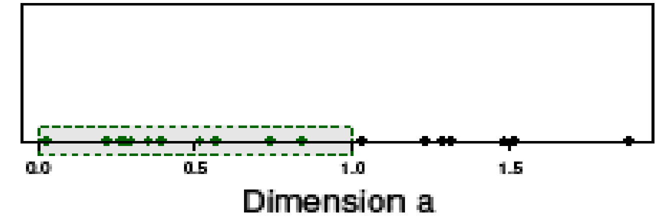
- Hierarchical Clustering can give different partitionings depending on the level-of-resolution we are looking at
- Hierarchical clustering doesn't need the number of clusters to be specified
- Hierarchical clustering can be slow (has to make several merge/split decisions)

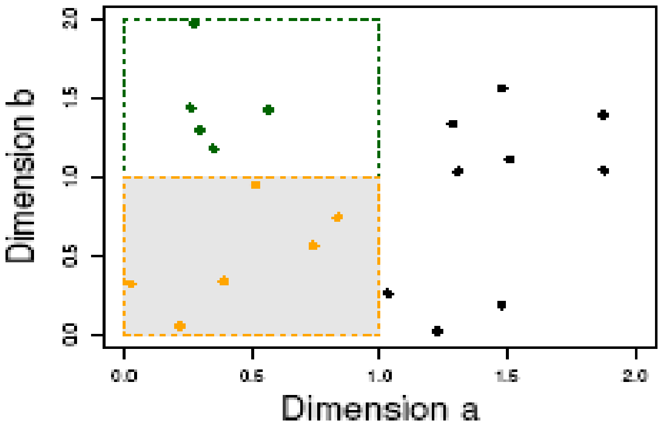
MISCELLANEOUS CONCEPTS

The Curse of Dimensionality

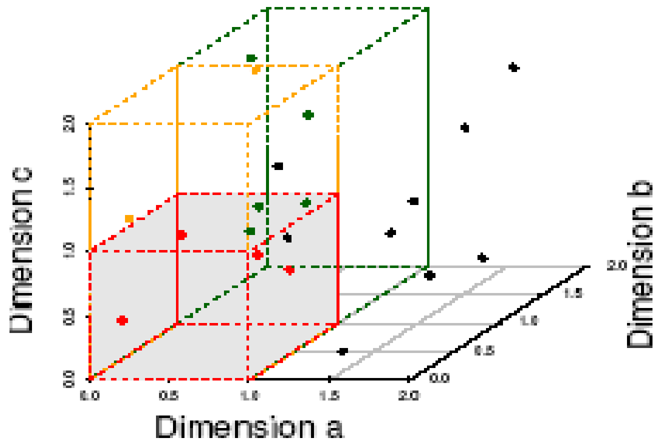
(graphs adapted from Parsons et al. KDD Explorations 2004)

- Data in only one dimension is relatively packed
- Adding a dimension “stretch” the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measure becomes meaningless—due to equi-distance





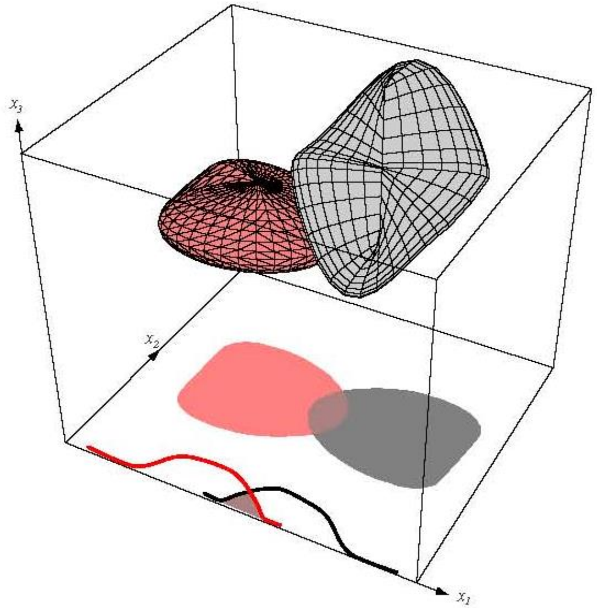
(b) 6 Objects in One Unit Bin



(c) 4 Objects in One Unit Bin

Data Preparation : Dimensionality Reduction

- If data x lies in high dimensional space, then an enormous amount of data is required to learn distributions or decision rules.
- The Main Idea
  - Reduce the dimensionality of the space
  - Project the d-dimensional points in a k-dim space
  - k << d
  - distances are preserved as well as possible
- Solve the problem in low dimensions



Clustering customers - - Attributes used in various industry scenarios

- The columns are whatever information is available at hand, typically :
  - Retail :
    - Spending in each product category (Electronics, Fashion..)
    - Frequency & Recency of visits/purchases.
  - Telecom
    - Voice usage (minutes), data usage (GB), prepaid recharge frequency, avg. recharge denomination..
  - When demographics/KYC is available (e.g. banking)
    - attributes like age, income, gender, marital status…

Inspire...Educate...Transform.

HYDERABAD

Office and Classrooms
Plot 63/A, Floors 1&2, Road # 13, Film Nagar,
Jubilee Hills, Hyderabad - 500 033
+91-9701685511 (Individuals)
+91-9618483483 (Corporates)

BENGALURU

Office
Incubex, #728, Grace Platina, 4th Floor, CMH Road,
Indira Nagar, 1st Stage, Bengaluru – 560038
+91-9502334561 (Individuals)
+91-9502799088 (Corporates)

Classroom
KnowledgeHut Solutions Pvt. Ltd., Reliable Plaza,
Jakkasandra Main Road, Teacher's Colony, 14th Main
Road, Sector – 5, HSR Layout, Bengaluru - 560102

Social Media

Web:	http://www.insofe.edu.in
Facebook:	https://www.facebook.com/insofe
Twitter:	https://twitter.com/Insofeedu
YouTube:	http://www.youtube.com/InsofeVideos
SlideShare:	http://www.slideshare.net/INSOFE
LinkedIn:	http://www.linkedin.com/company/international-school-of-engineering

This presentation may contain references to findings of various reports available in the public domain. INSOFE makes no representation as to their accuracy or that the organization subscribes to those findings.