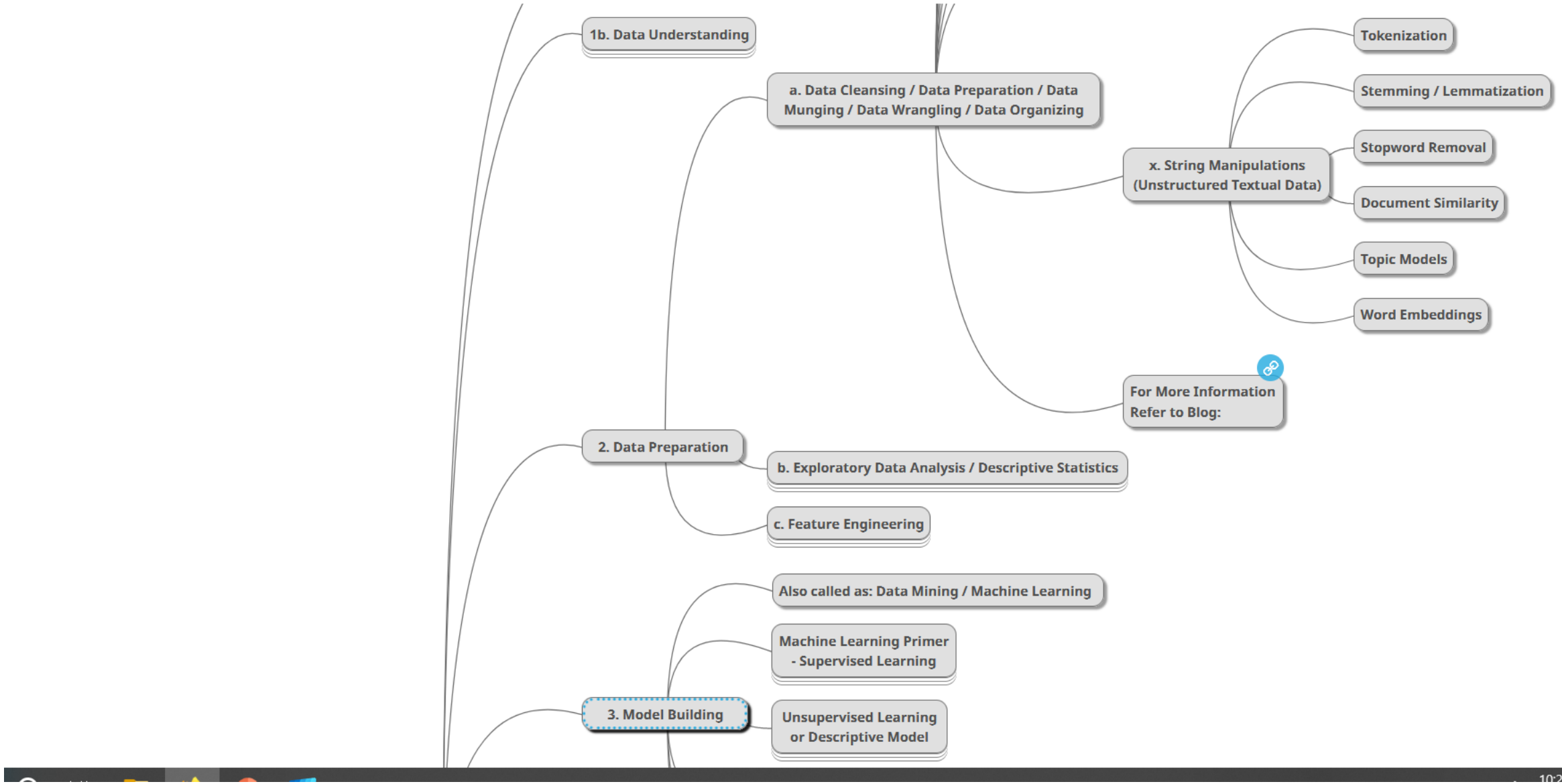
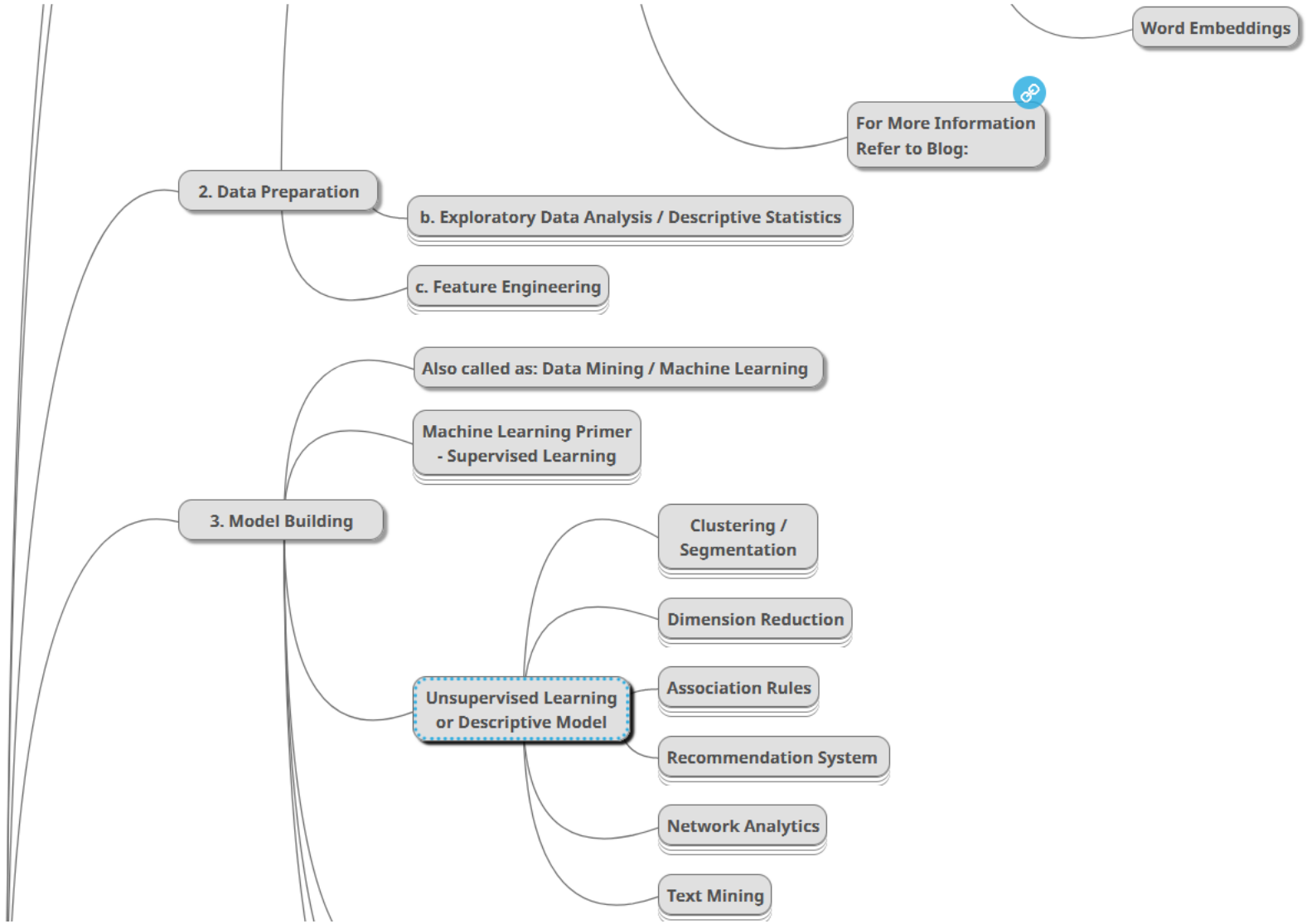
Data Mining

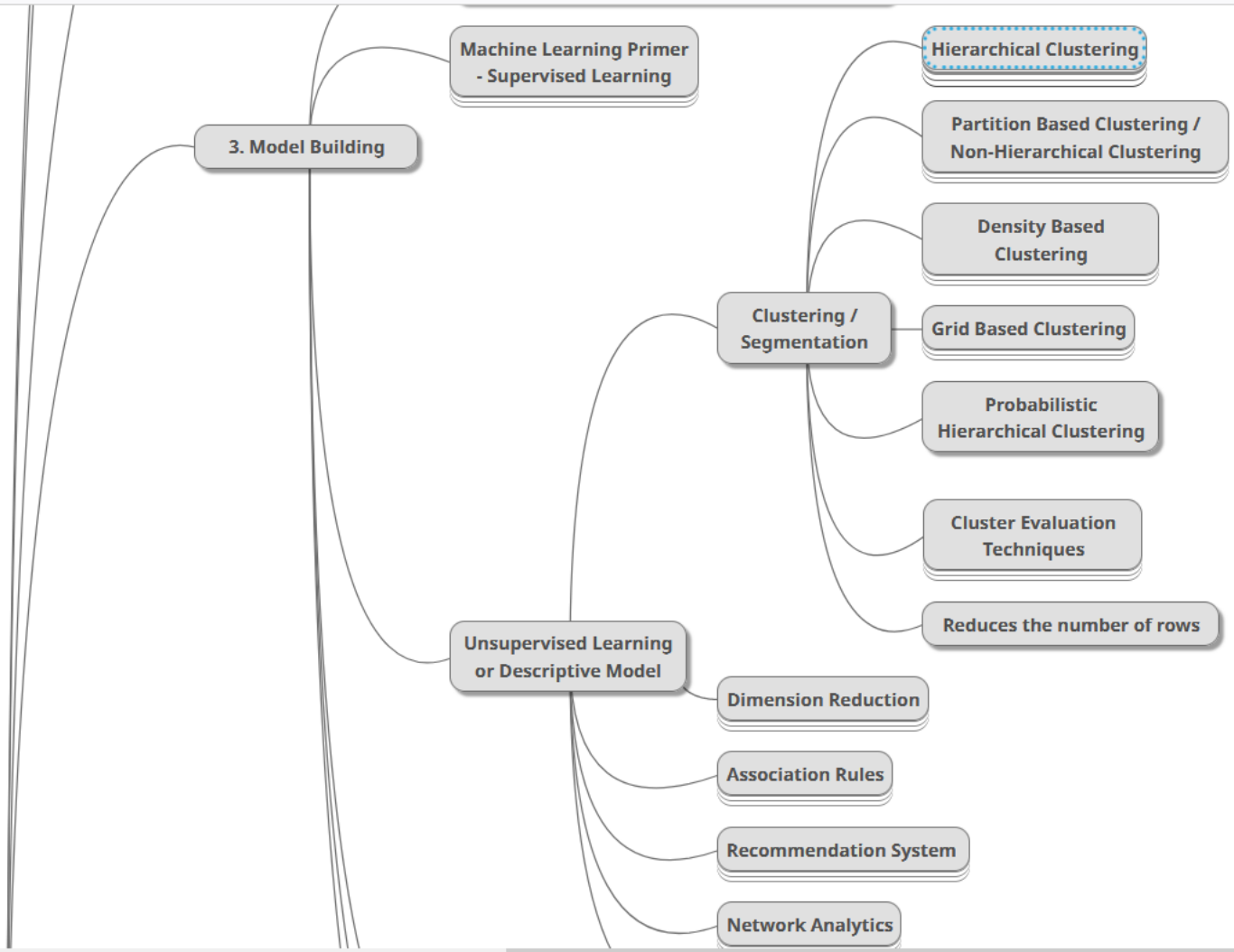
And Clustering

Please double-click to open



Please double-click to open




3. Model Building
Machine Learning Primer - Supervised Learning
Unsupervised Learning or Descriptive Model
Clustering / Segmentation
Hierarchical Clustering
Partition Based Clustering / Non-Hierarchical Clustering
Density Based Clustering
Grid Based Clustering
Probabilistic Hierarchical Clustering
Cluster Evaluation Techniques
Reduces the number of rows
Dimension Reduction
Association Rules
Recommendation System
Network Analytics

Machine learning primer

"How do we create computer programs that improve with experience?"

Tom Mitchell

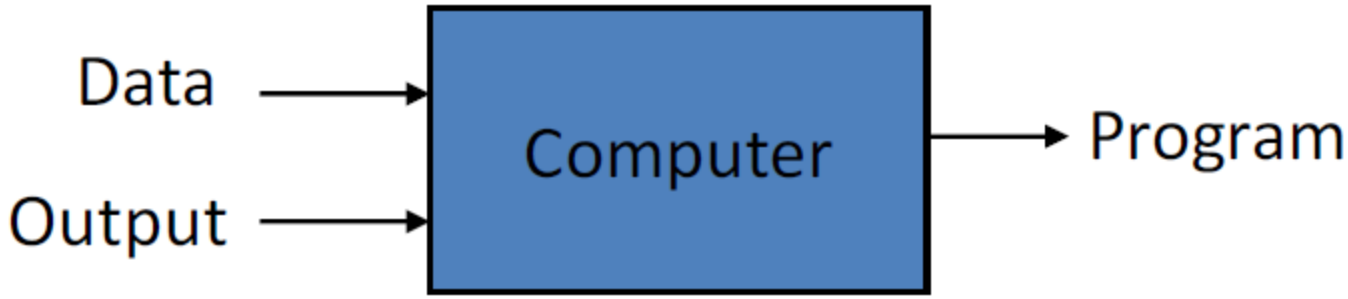
http://videolectures.net/mlas06_mitchell_itm/

"A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. "

Tom Mitchell. Machine Learning 1997.

Traditional Programming

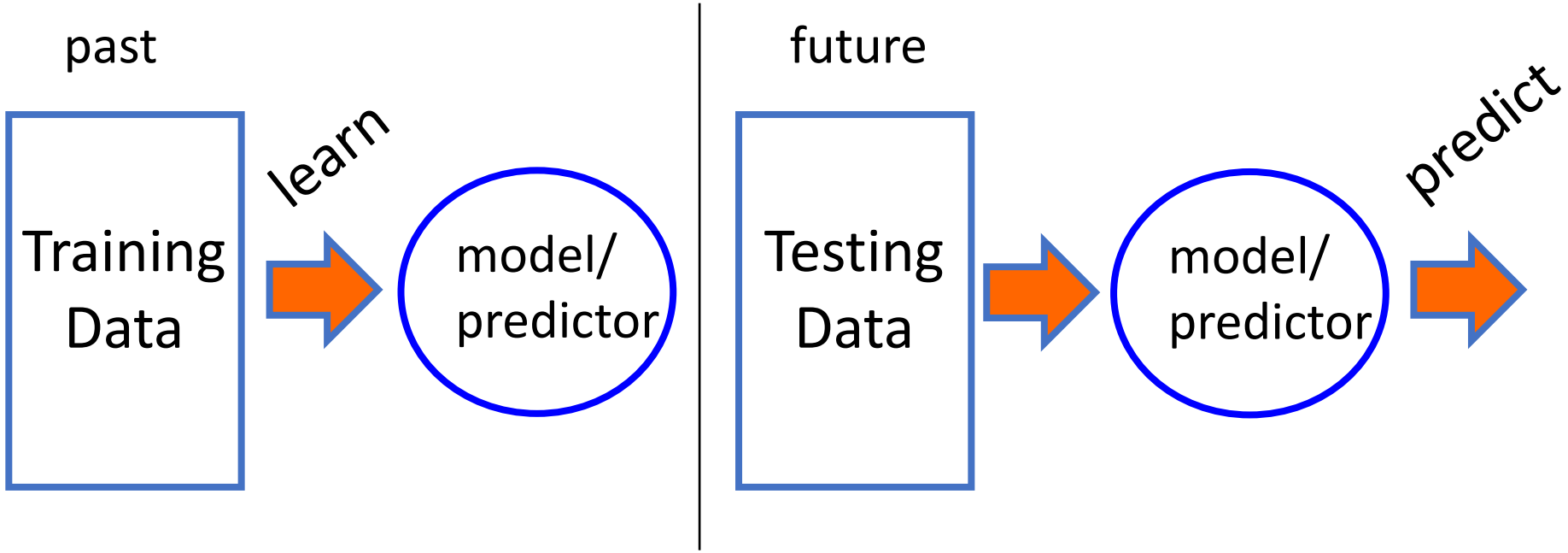
Machine Learning



Machine Learning is…

Machine learning is about predicting the future based on the past.

-- Hal Daume III

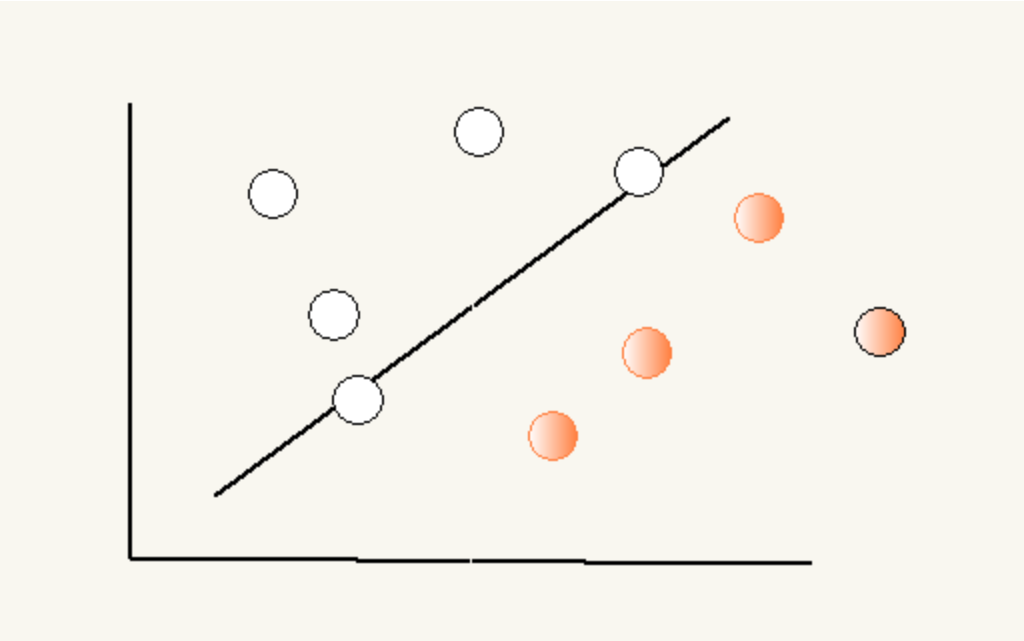


Terminology used in machine learning

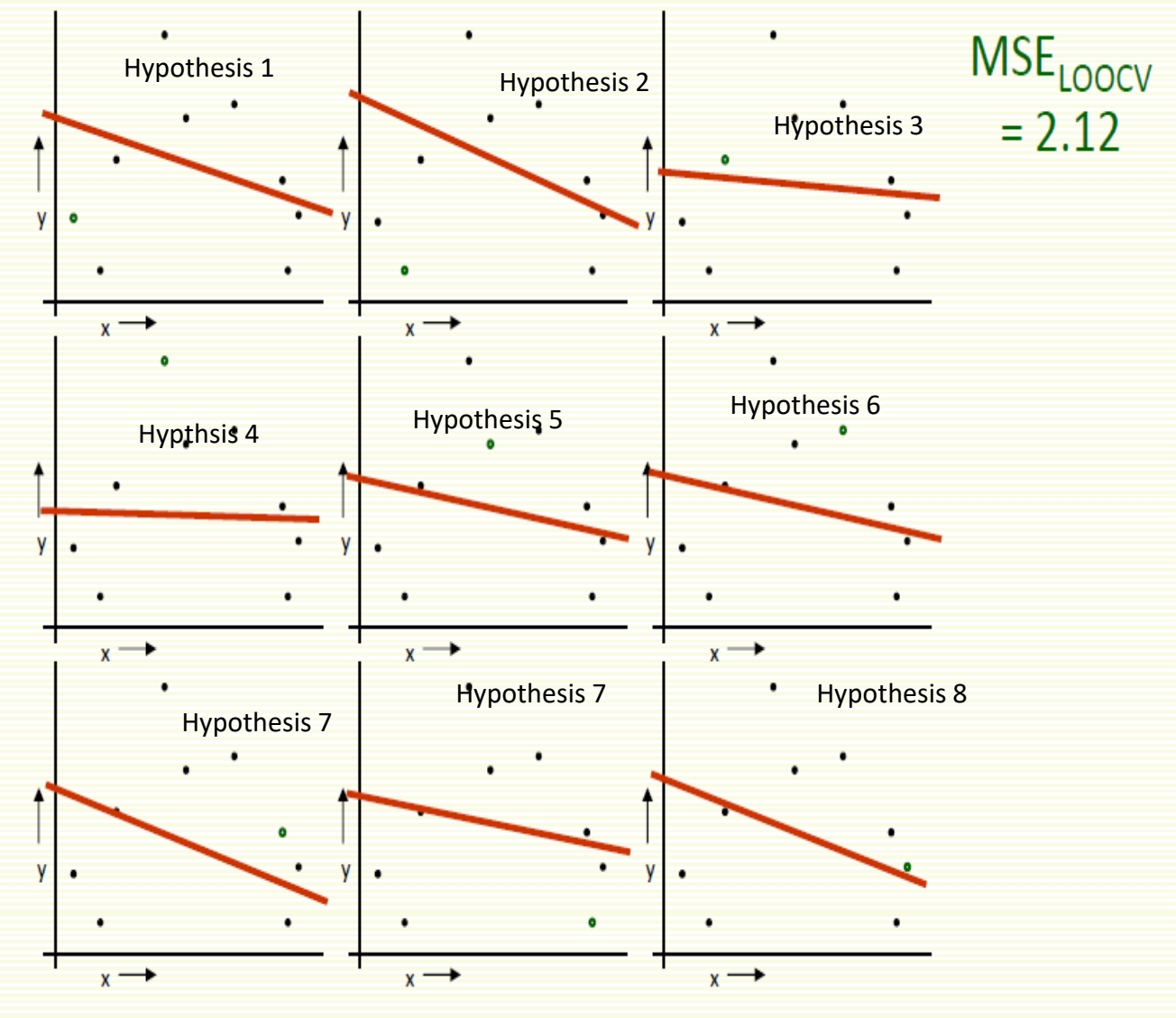
- **Training example**: a sample from x including its output from the target function
- **Target function**: the mapping function f from x to f(x)
- **Hypothesis**: approximation of f, a candidate function.
- **Concept**: A boolean target function, positive examples and negative examples for the 1/0 class values.
- **Classifier**: Learning program outputs a classifier that can be used to classify.
- **Learner**: Process that creates the classifier.
- **Hypothesis space**: set of possible approximations of f that the algorithm can create.
- **Version space**: subset of the hypothesis space that is consistent with the observed data.

Target function

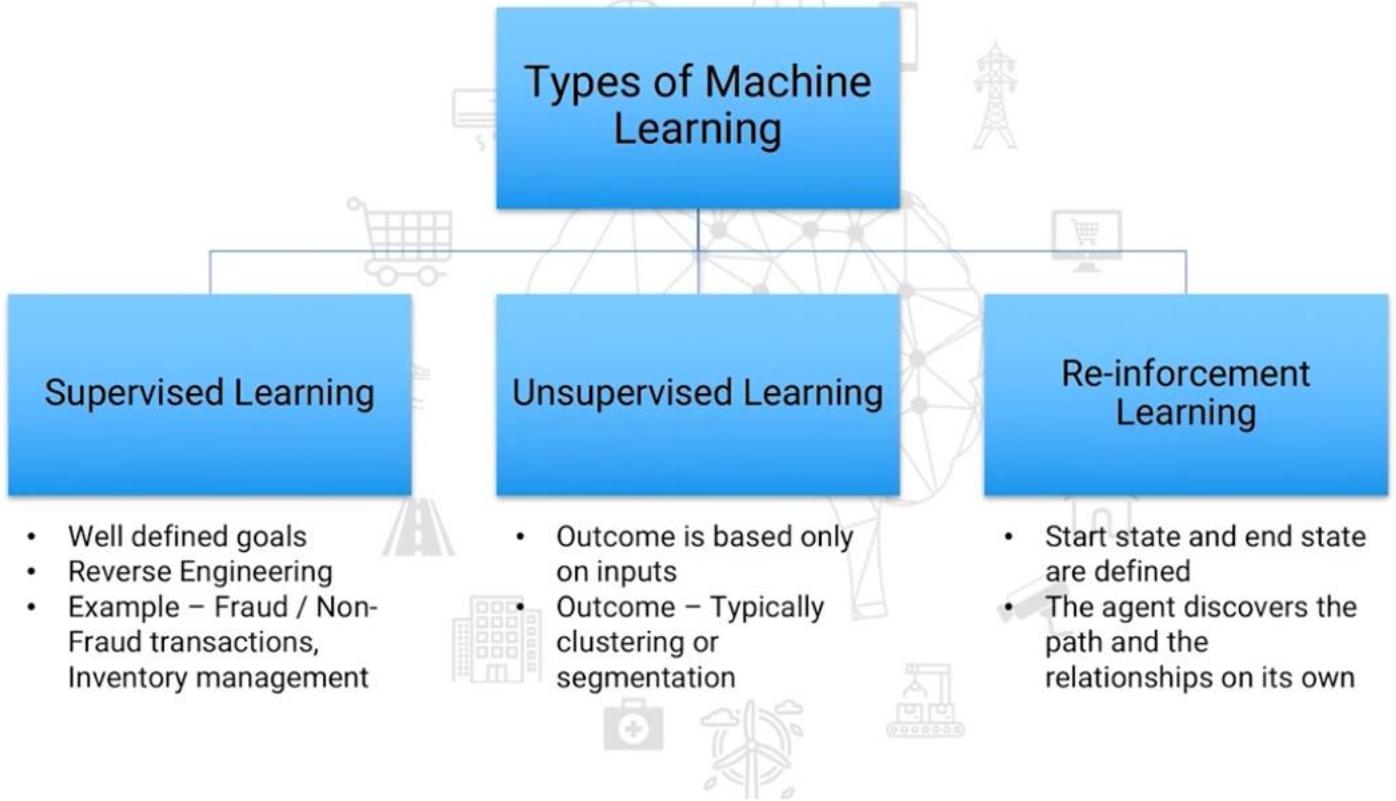
- The line which cover the all the points ,equation of this plane is called target function.



Hypothesis


Hypothesis 1
Hypothesis 2
Hypothesis 3
MSE LOOCV = 2.12
Hypthsis 4
Hypothesis 5
Hypothesis 6
Hypothesis 7
Hypothesis 7
Hypothesis 8
y
x

Types of Machine Learning



Supervised learning

examples

label

$label_1$

$label_3$

$label_4$

$label_5$

labeled examples

Supervised learning: given labeled examples

Supervised learning

label

$label_1$

$label_3$

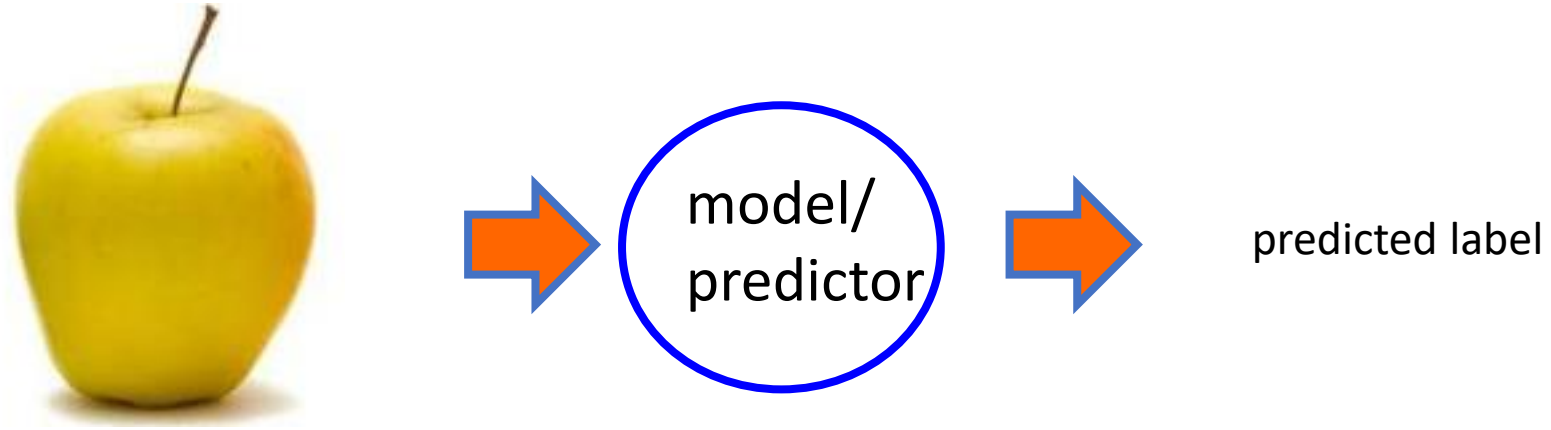
$label_4$

$label_5$

model/
predictor

Supervised learning: given labeled examples

Supervised learning



Supervised learning: learn to predict new example

Supervised learning: classification

label

apple

apple

Classification: a finite set of labels

banana

banana

Supervised learning: given labeled examples

Supervised Learning

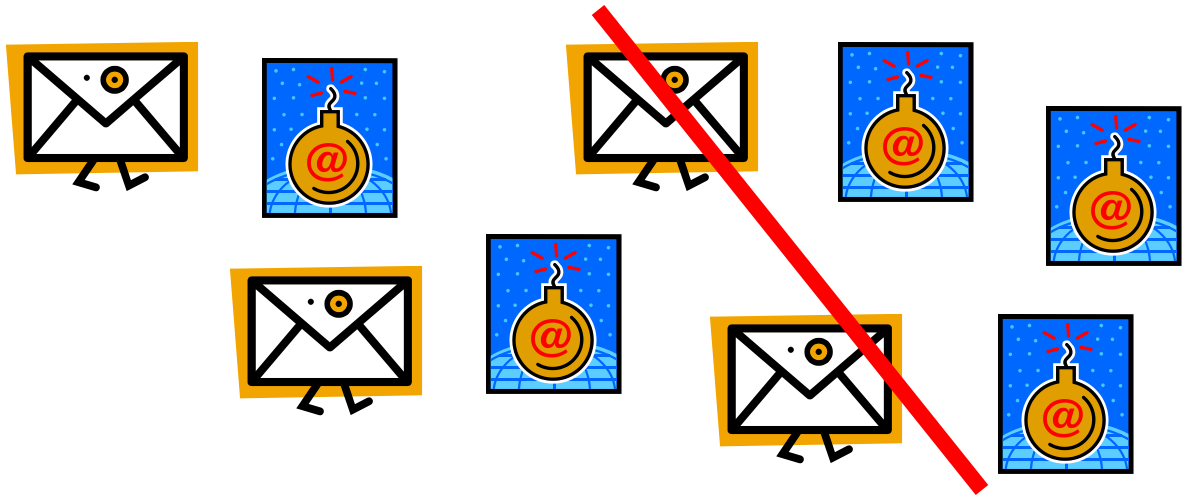
Supervised Learning = learning from labeled data. Dominant paradigm in Machine Learning.

- E.g, say you want to train an email classifier to distinguish spam from important messages
- Take sample S of data, labeled according to whether they were/weren't spam.
- Train a classifier (like SVM, decision tree, etc) on S. Make sure it's not overfitting.
- Use to classify new emails.

Supervised Learning

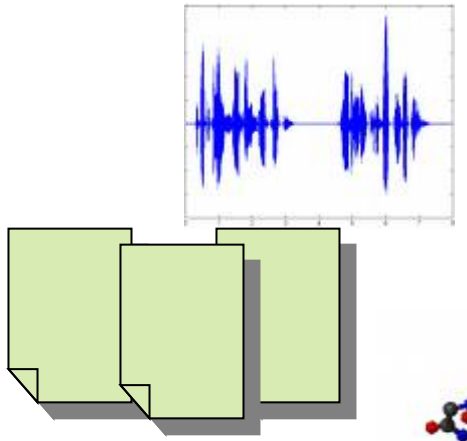
Supervised Learning = learning from labeled data. Dominant paradigm in Machine Learning.

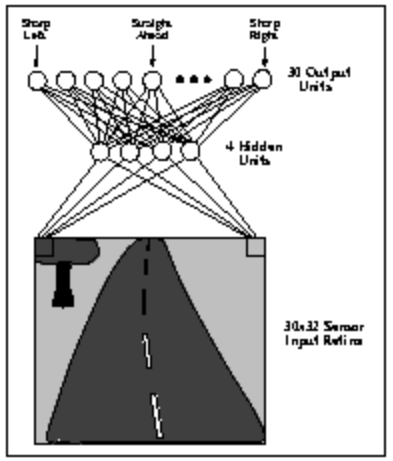
- E.g, say you want to train an email classifier to distinguish spam from important messages
- Take sample S of data, labeled according to whether they were/weren't spam.



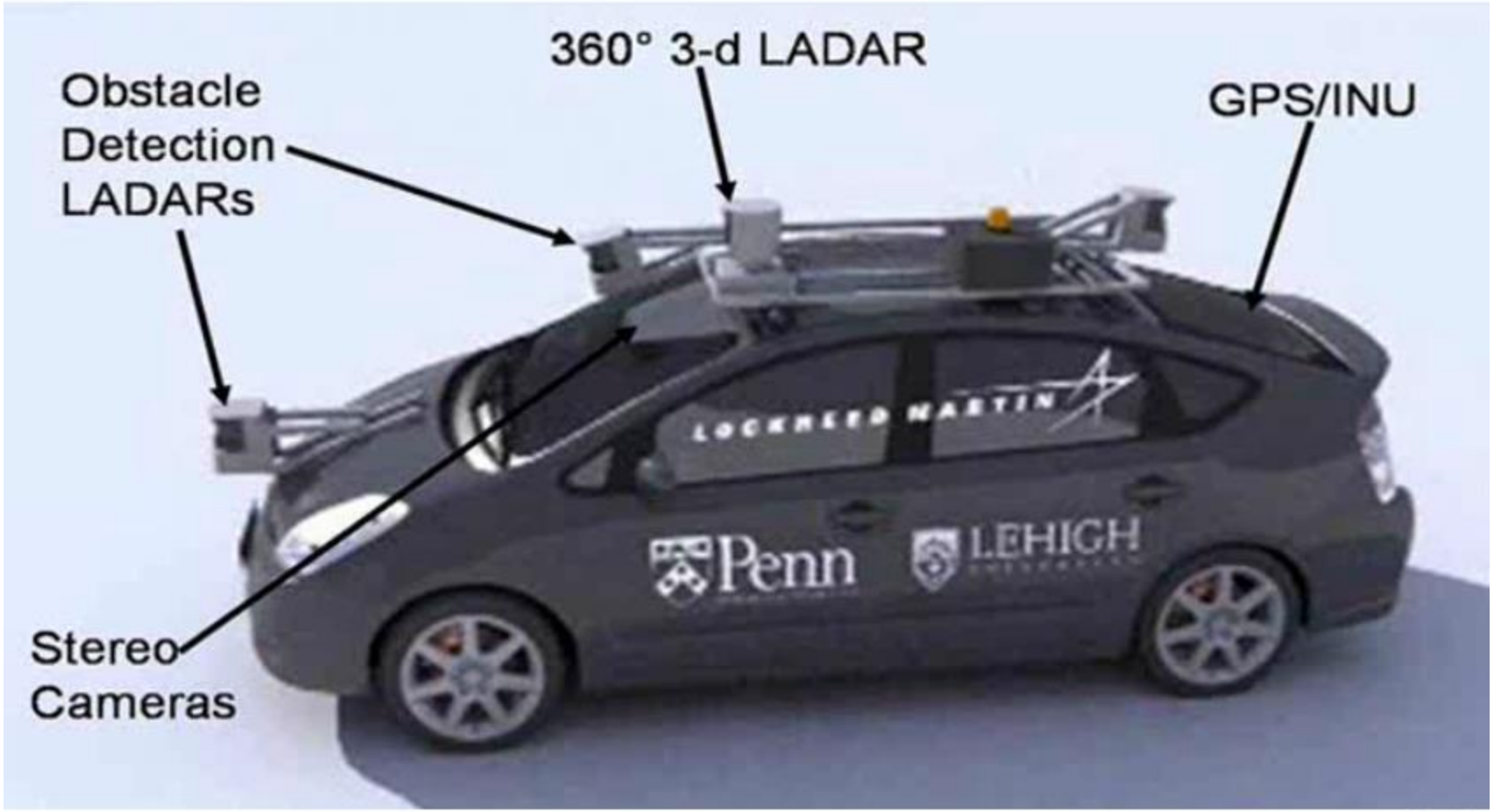
Basic paradigm has many successes

- recognize speech,
- steer a car,
- classify documents
- classify proteins
- recognizing faces, objects in images
- …

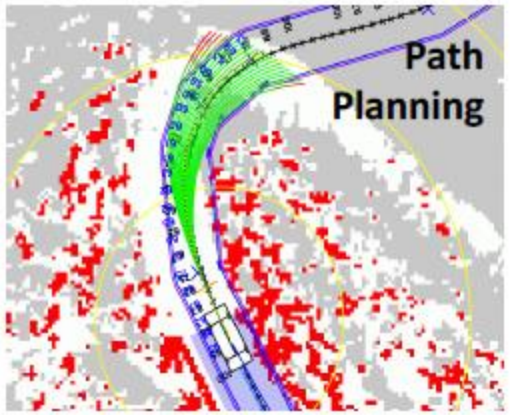




Autonomous Car Sensors



Autonomous Car Technology



Images and movies taken from Sebastian Thrun's multimedia website.

However, for many problems, labeled data can be rare or expensive.

Need to pay someone to do it, requires special testing,…

Unlabeled data is much cheaper.

However, for many problems, labeled data can be rare or expensive.

Need to pay someone to do it, requires special testing,…

Unlabeled data is much cheaper.

Speech

Customer modeling

Images

Protein sequences

Medical outcomes

Web pages

However, for many problems, labeled data can be rare or expensive.

Need to pay someone to do it, requires special testing,…

Unlabeled data is much cheaper.

Task: speech analysis [From Jerry Zhu]

- Switchboard dataset
- telephone conversation transcription
- 400 hours annotation time for each hour of speech

film ⇒ f ih_n uh_gl_n m

be all ⇒ bcl b iy iy_tr ao_tr ao l_dl

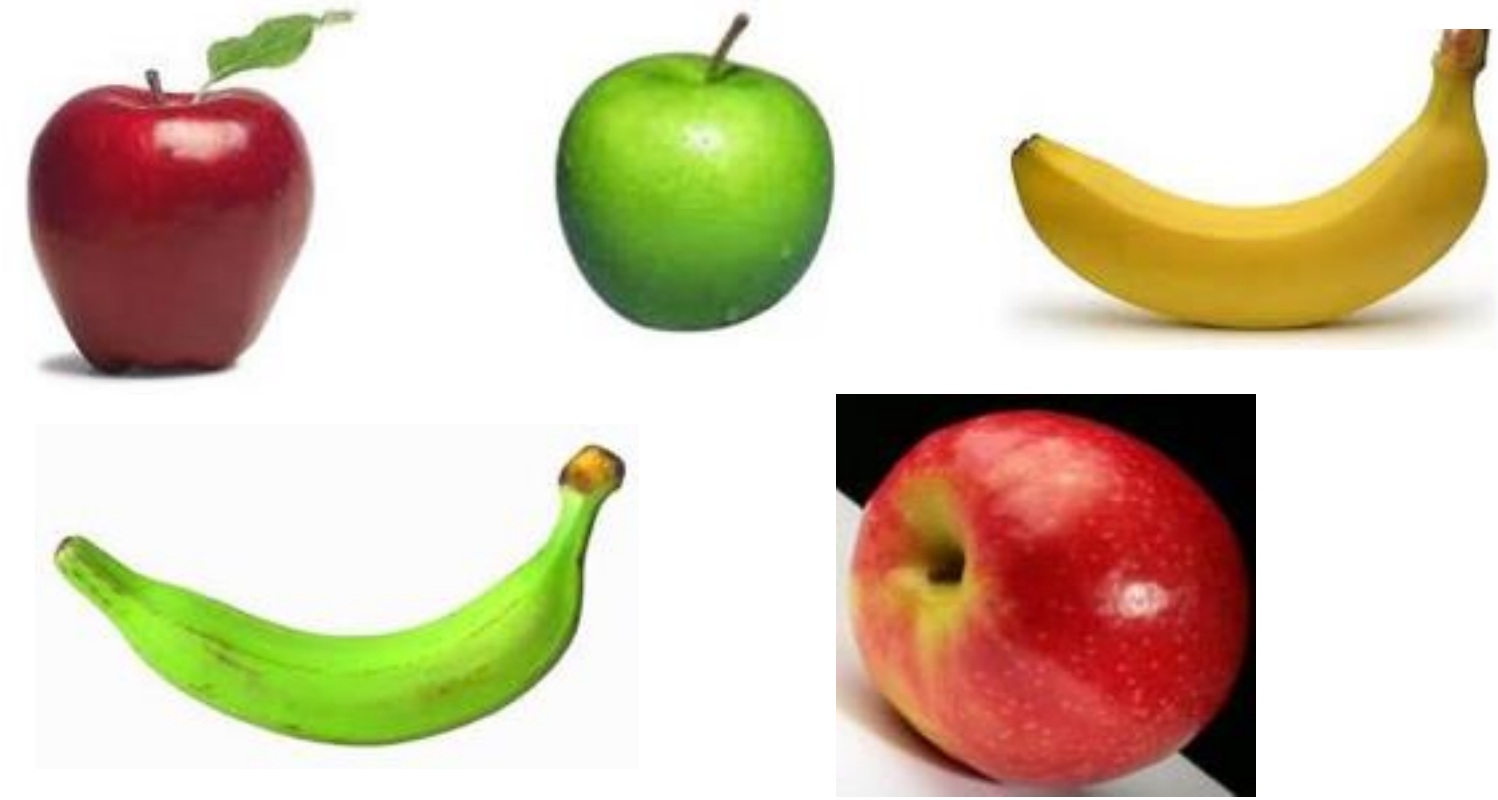
However, for many problems, labeled data can be rare or expensive.

Need to pay someone to do it, requires special testing,…

Unlabeled data is much cheaper.

Can we make use of cheap unlabeled data?

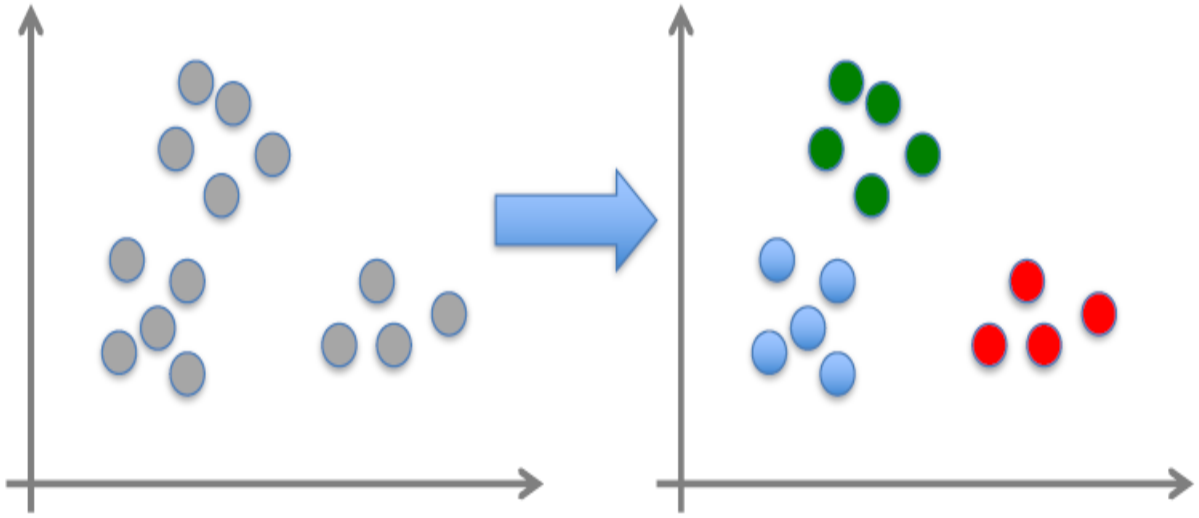
Unsupervised learning



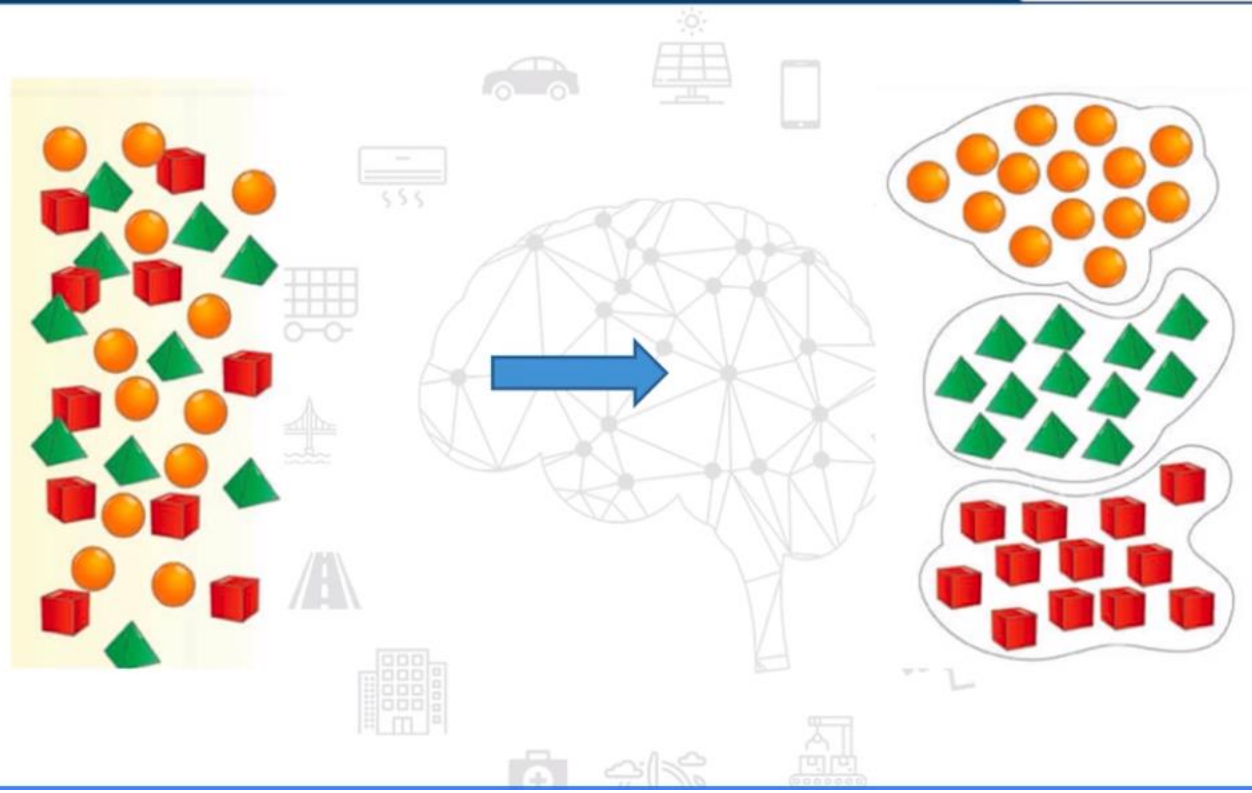
Unupervised learning: given data, i.e. examples, but no labels

Unsupervised Learning

- Given $x_1, x_2, \ldots, x_n$ (without labels)
- Output hidden structure behind the x's
  - E.g., clustering



Unsupervised learning at work



Unsupervised learning applications

learn clusters/groups without any label

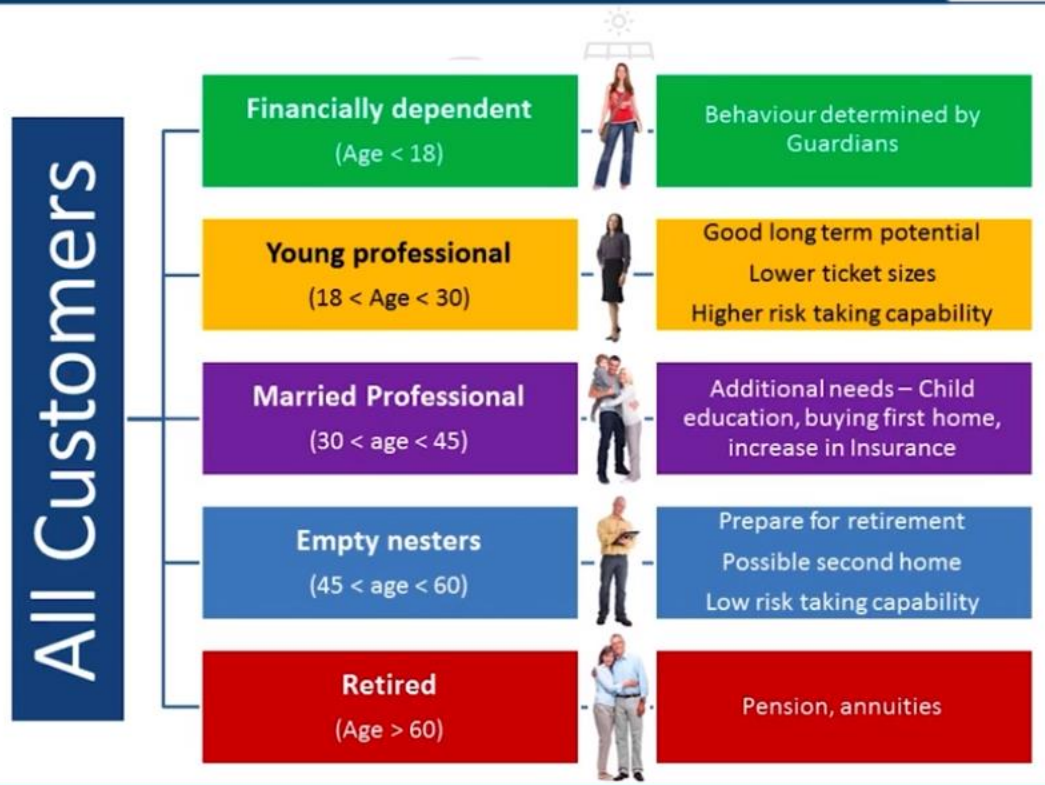
customer segmentation (i.e. grouping)

image compression

bioinformatics: learn motifs

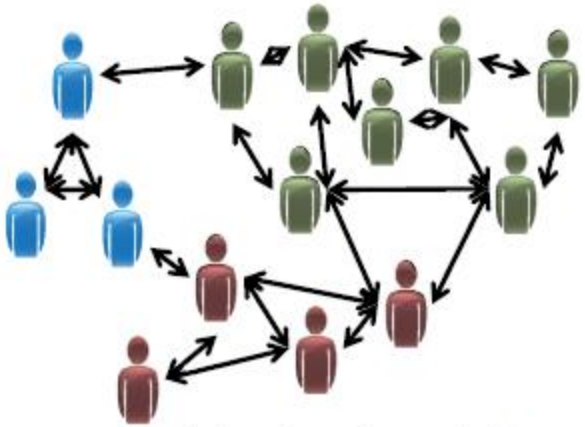
…

Example of Unsupervised Learning



Unsupervised Learning

Organize computing clusters



Social network analysis

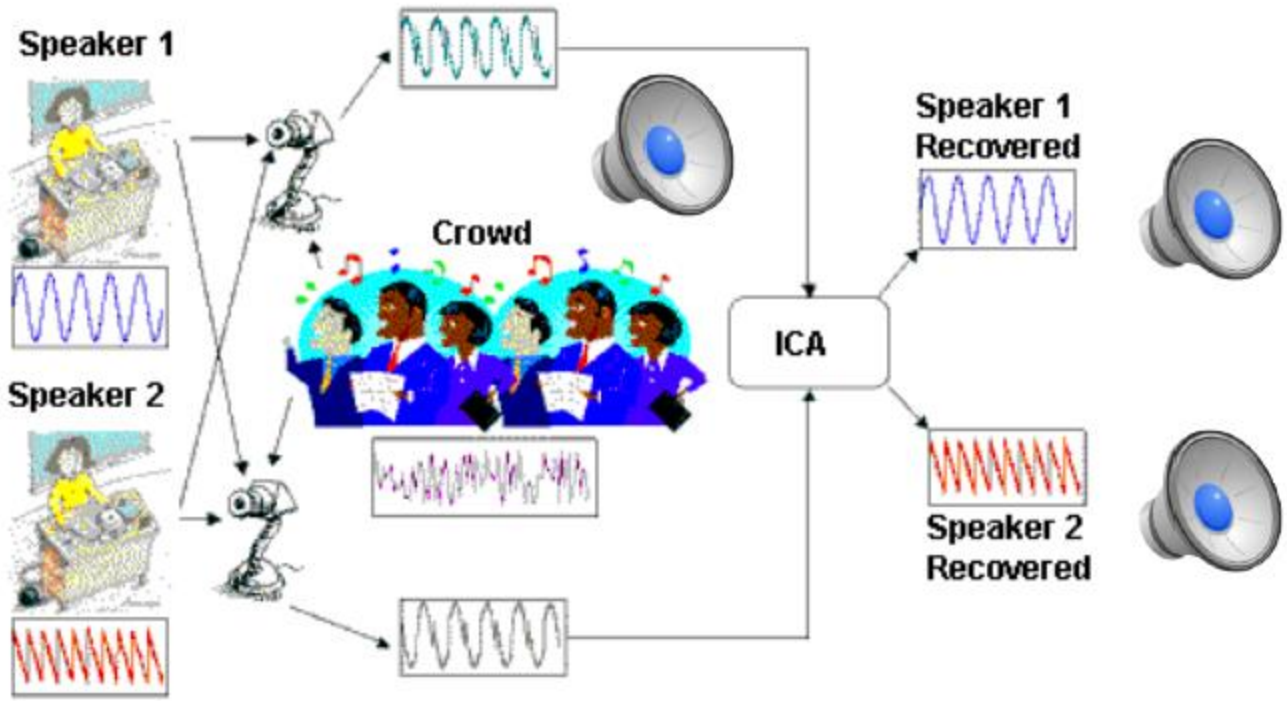
Market segmentation

Astronomical data analysis

lide credit: Andrew Ng

Unsupervised Learning

- Independent component analysis – separate a combined signal into its original sources



Humans vs. Machines - Unsupervised

- Google passes the "purring test" (ICML'12)
  - 16K cores watching 10M youtube stills for 3 days
  - completely unsupervised: the cat has just appeared as a useful concept to represent

Re-inforcement Learning

Observe how you are walking

⇩

Attempt to stand up

⇩

Manage to be balanced

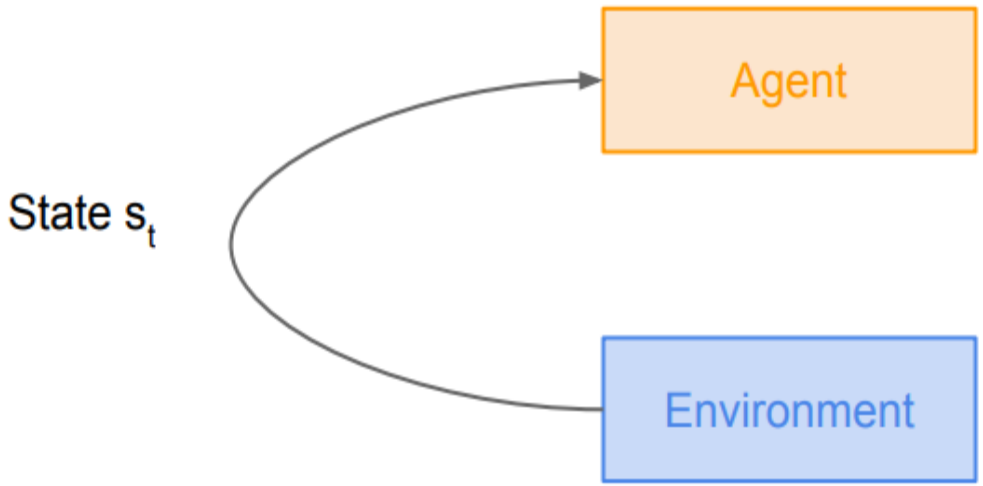
⇩

Decide which foot to put forward

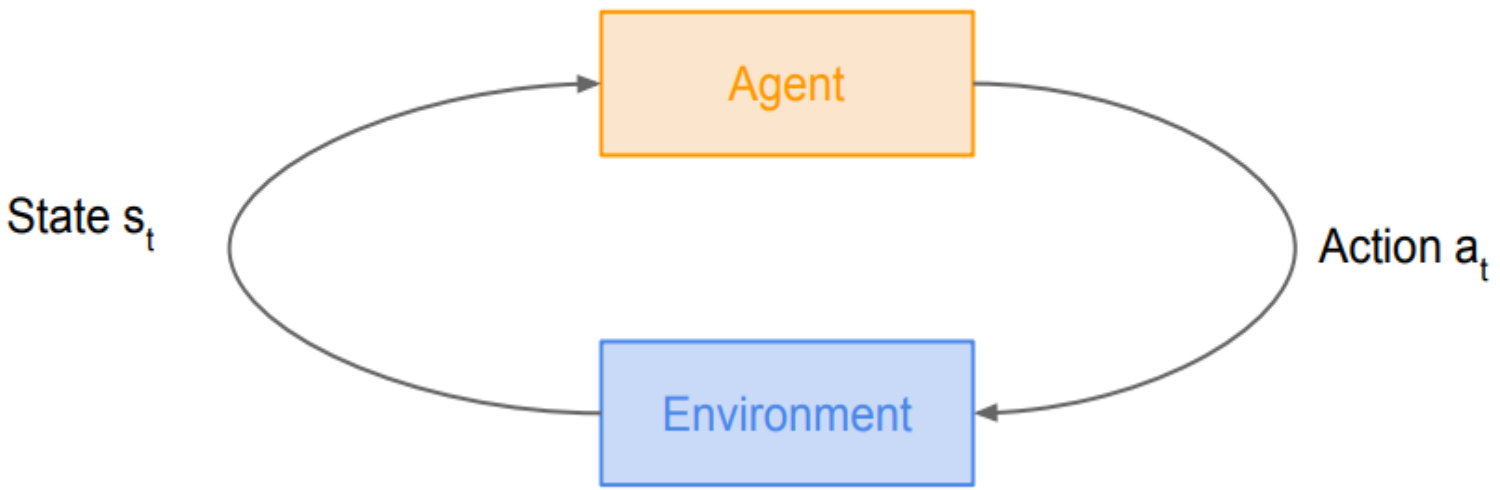
Re-inforcement Learning

Reinforcement Learning

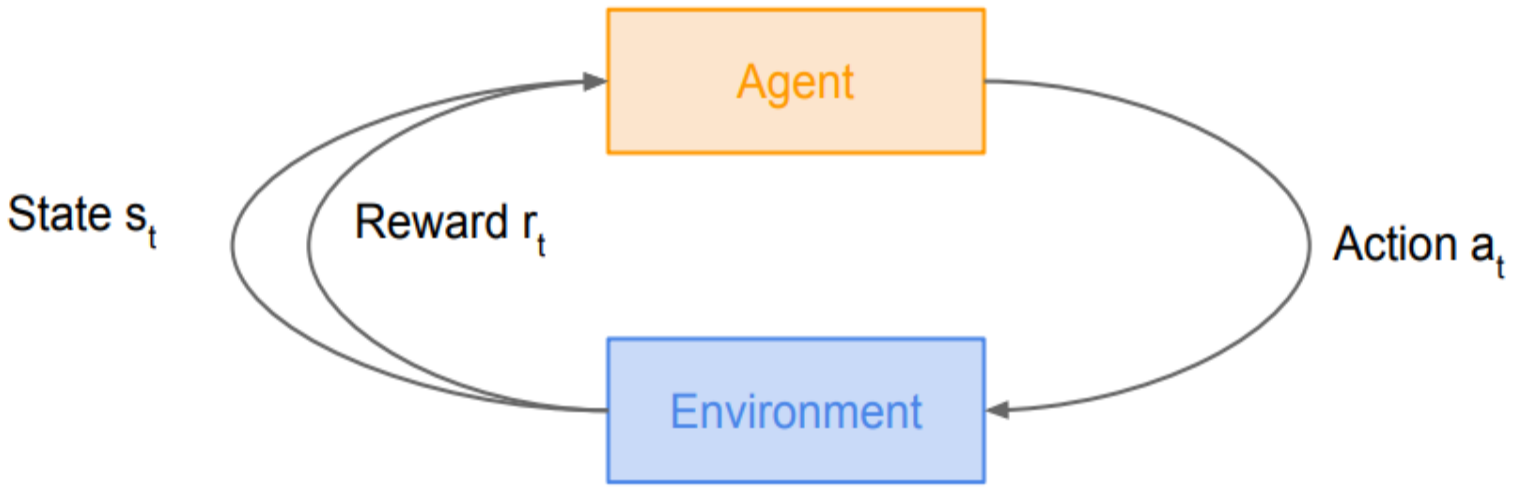
Reinforcement Learning

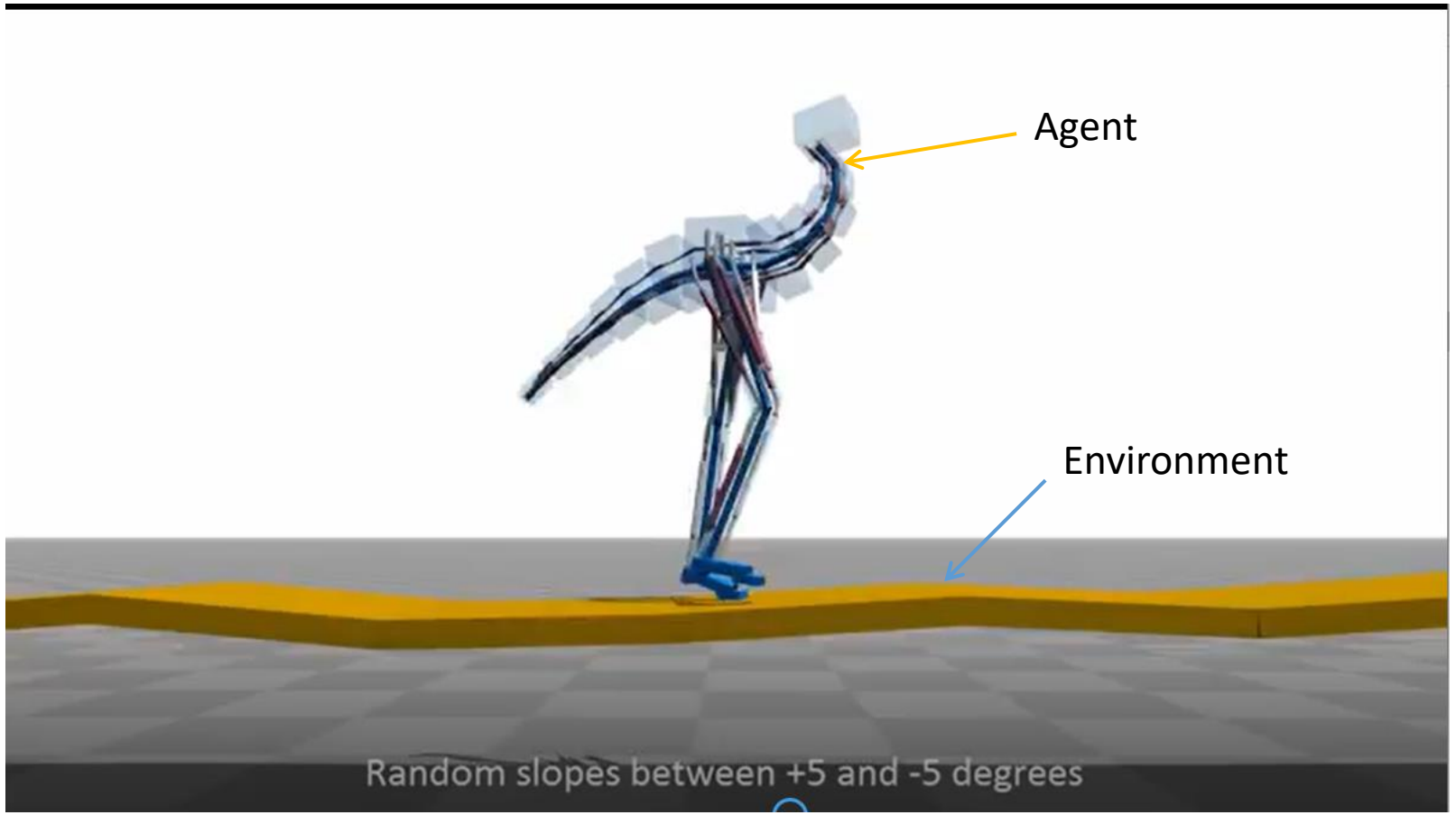


Reinforcement Learning



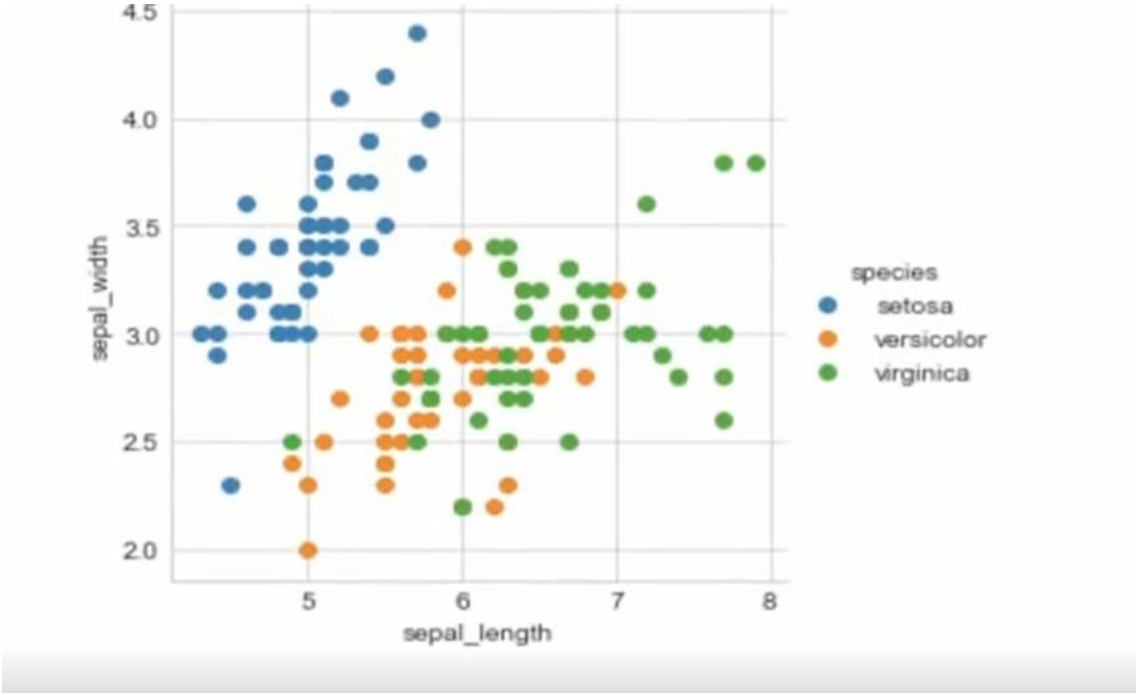
Reinforcement Learning




Agent
Environment
Random slopes between +5 and -5 degrees

Clustering

- When we have single variable: binning
- When we have two or more variables then clustering



- **What is Clustering?**
- A bank wants to give credit card offers to its customers. Currently, they look at the details of each customer and based on this information, decide which offer should be given to which customer.
- Now, the bank can potentially have millions of customers. Does it make sense to look at the details of each customer separately and then make a decision? Certainly not! It is a manual process and will take a huge amount of time.

- So what can the bank do? One option is to segment its customers into different groups. For instance, the bank can group the customers based on their income:

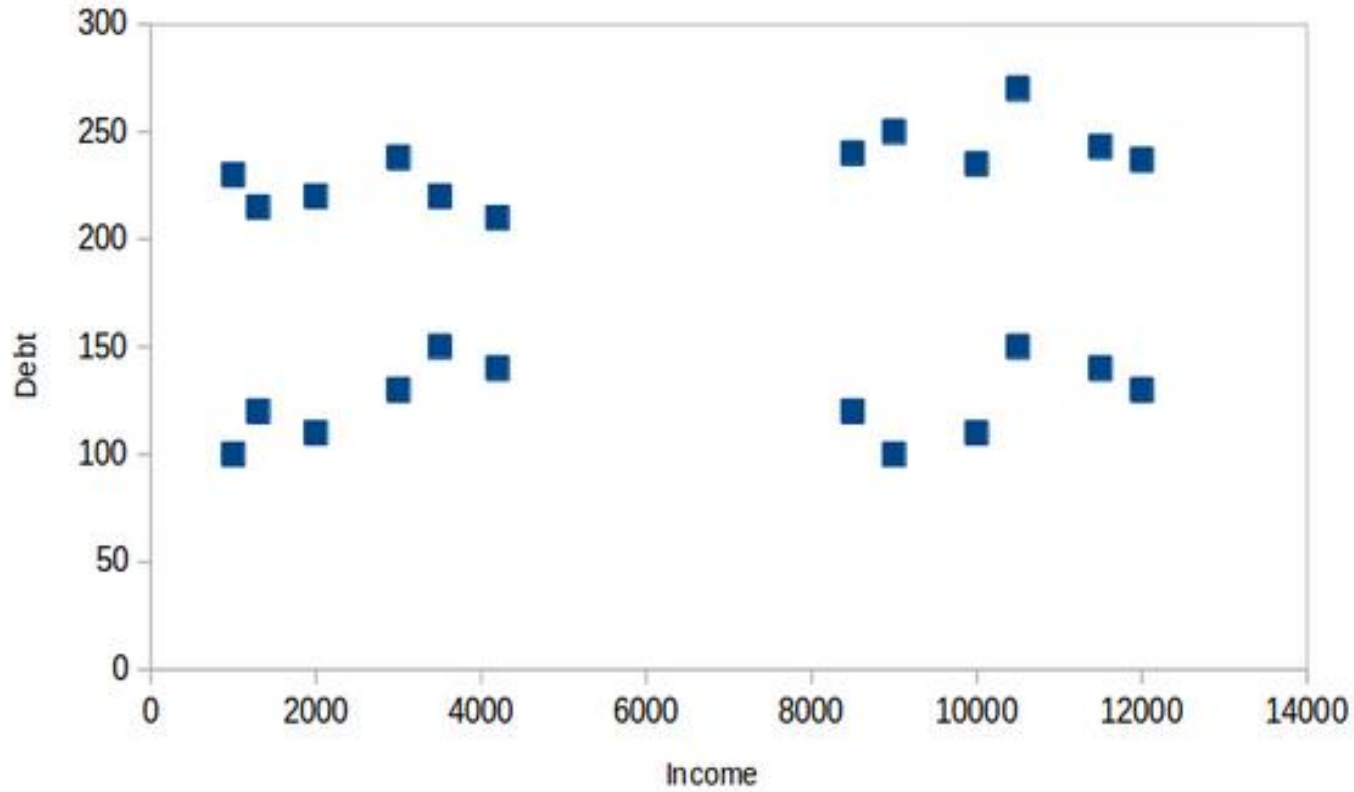
- *Clustering is the process of dividing the entire data into groups (also known as clusters) based on the patterns in the data.*

- A project where your task is to predict whether a loan will be approved or not:
- in the loan approval problem, we have to predict the *Loan_Status* depending on the *Gender, marital status, the income of the customers, etc*.

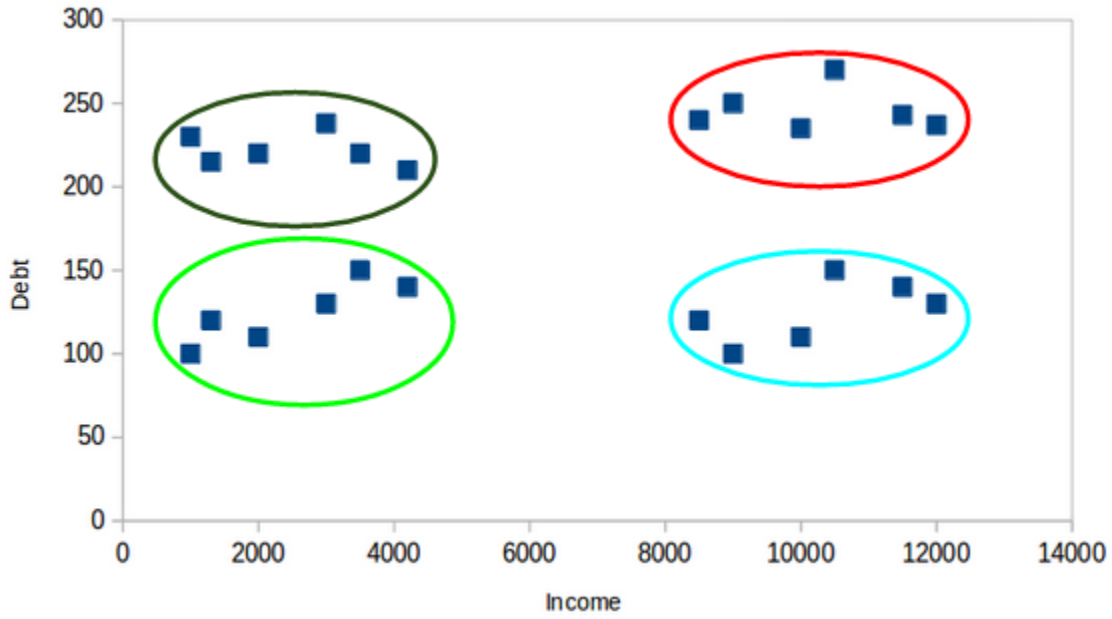
Loan_ID	Gender	Married	ApplicantIncome	LoanAmount	Loan_Status
LP001002	Male	No	5849	130.0	Y
LP001003	Male	Yes	4583	128.0	N
LP001005	Male	Yes	3000	66.0	Y
LP001006	Male	Yes	2583	120.0	Y
LP001008	Male	No	6000	141.0	Y

- **In clustering, we do not have a target to predict. We look at the data and then try to club similar observations and form different groups. Hence it is an unsupervised learning problem.**

- We’ll take the same bank as before who wants to segment its customers.
- For simplicity purposes, let’s say the bank only wants to use the income and debt to make the segmentation.
- They collected the customer data and used a scatter plot to visualize it:

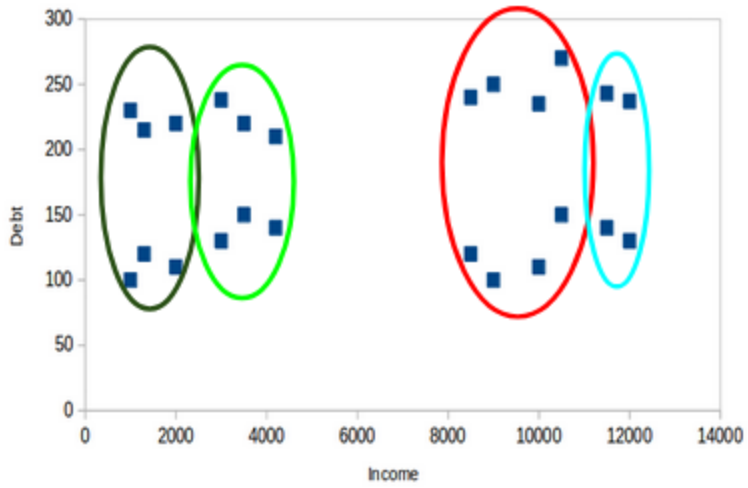

300
250
200
150
100
50
0
Debt
0
2000
4000
6000
8000
10000
12000
14000
Income

- On the X-axis, we have the income of the customer and the y-axis represents the amount of debt. Here, we can clearly visualize that these customers can be segmented into 4 different clusters as shown below:
-

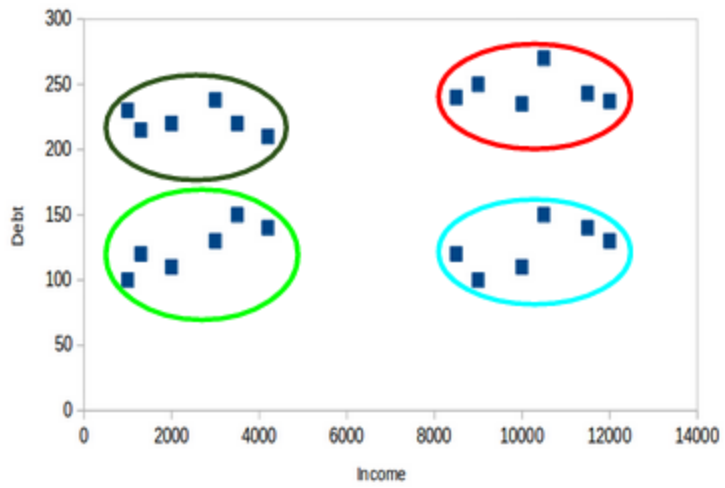


- This is how clustering helps to create segments (clusters) from the data. The bank can further use these clusters to make strategies and offer discounts to its customers.
- **Property 1**
- **All the data points in a cluster should be similar to each other.**

- **Property 2**
- **The data points from different clusters should be as different as possible.**

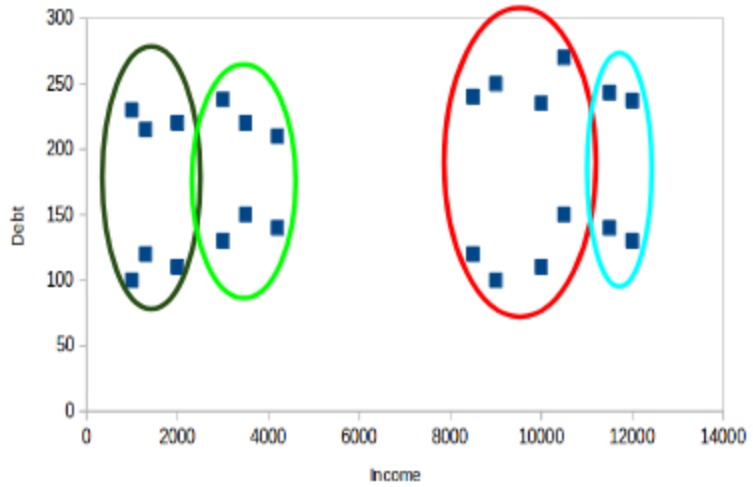


Case - I



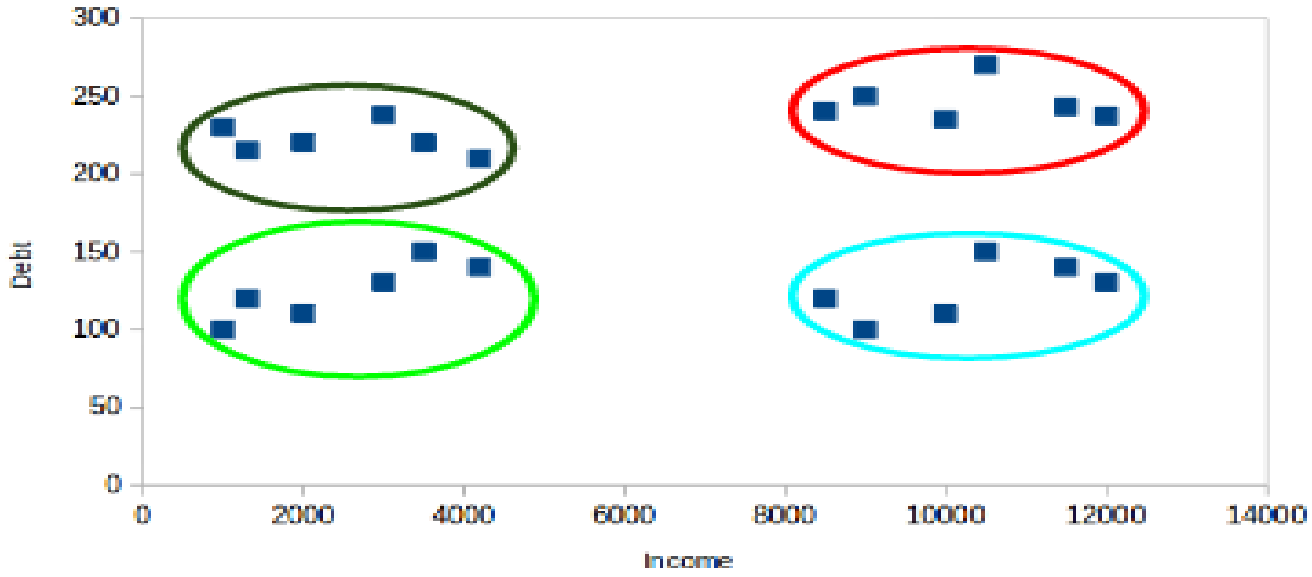
Case - II

Customers in the red and blue clusters are quite similar to each other. The top four points in the red cluster share similar properties as that of the top two customers in the blue cluster. They have high income and high debt value. Here, we have clustered them differently. Whereas, if you look at case II:



Case - I

Points in the red cluster are completely different from the customers in the blue cluster. All the customers in the red cluster have high income and high debt and customers in the blue cluster have high income and low debt value. Clearly we have a better clustering of customers in this case.



Case - II

Types of clustering:

1. **Hierarchical algorithms**: these find successive clusters using previously established clusters.

   1. Agglomerative ("bottom-up"): Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.

   2. Divisive ("top-down"): Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

2. **Partitional clustering:** Partitional algorithms determine all clusters at once. They include:

   - ***K*-means and derivatives**
   - Fuzzy *c*-means clustering
   - QT clustering algorithm

Common Distance measures:

- *Distance measure* will determine how the *similarity* of two elements is calculated and it will influence the shape of the clusters.

  They include:

1. The Euclidean distance (also called 2-norm distance) is given by:

$$d(x, y) = \sum_{i=1}^{p} |x_i - y_i|$$

2. The Manhattan distance (also called taxicab norm or 1-norm) is given by:

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

3.The maximum norm is given by:

$$d(x, y) = \max_{1 \le i \le p} |x_i - y_i|$$

4. The Mahalanobis distance corrects data for different scales and correlations in the variables.

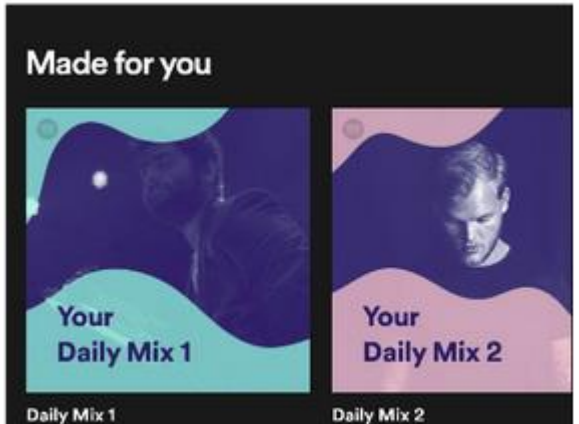
5. Inner product space: The angle between two vectors can be used as a distance measure when clustering high dimensional data

6. Hamming distance (sometimes edit distance) measures the minimum number of substitutions required to change one member into another.

Applications of Clustering in Real-World Scenarios

- **Customer Segmentation**
- **Document Clustering**
- This is another common application of clustering. Let’s say you have multiple documents and you need to cluster similar documents together. Clustering helps us group these documents such that similar documents are in the same clusters.
- **Image Segmentation**
- We can also use clustering to perform image segmentation. Here, we try to club similar pixels in t er. We can apply clustering to create clusters having simila ne group.
-

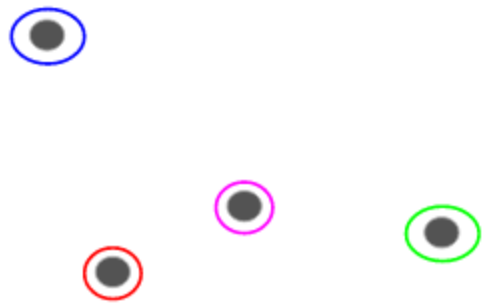
- **Recommendation Engines**
- Clustering can also be used in recommendation engines. Let’s say you want to recommend songs to your friends. You can look at the songs liked by that person and then use clustering to find similar songs and finally recommend the most similar songs.



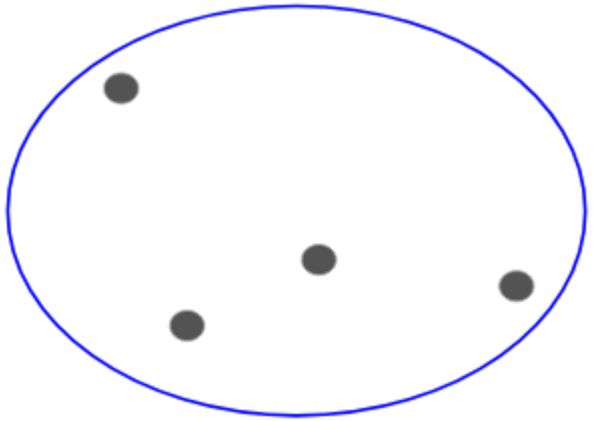
What is Hierarchical Clustering?

- Let’s say we have the below points and we want to cluster them into groups:
-

- We can assign each of these points to a separate cluster:
-

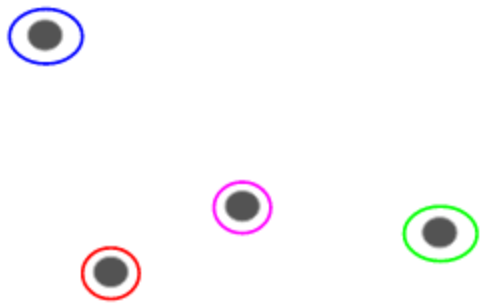


- Now, based on the similarity of these clusters, we can combine the most similar clusters together and repeat this process until only a single cluster is left:
- We are essentially building a hierarchy of clusters. That's why this algorithm is called hierarchical clustering.

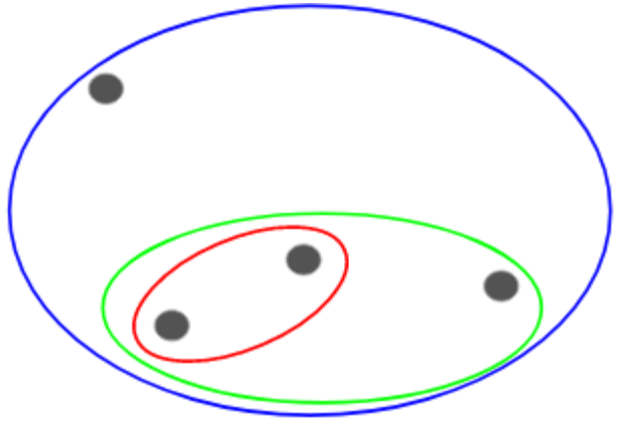


Agglomerative Hierarchical Clustering

- We assign each point to an individual cluster in this technique. Suppose there are 4 data points. We will assign each of these points to a cluster and hence will have 4 clusters in the beginning:

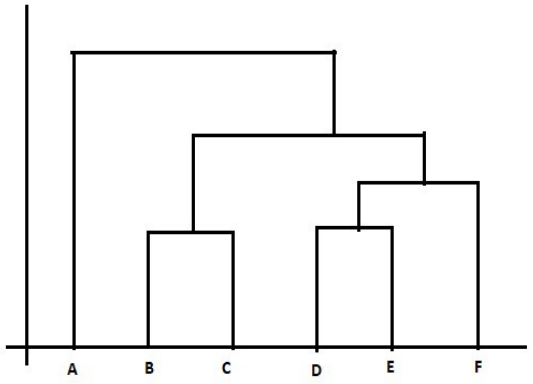


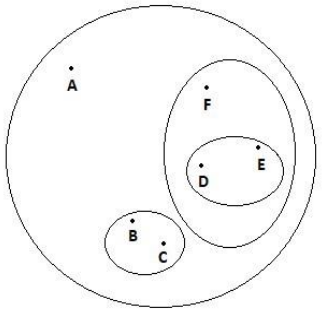
- Then, at each iteration, we merge the closest pair of clusters and repeat this step until only a single cluster is left:

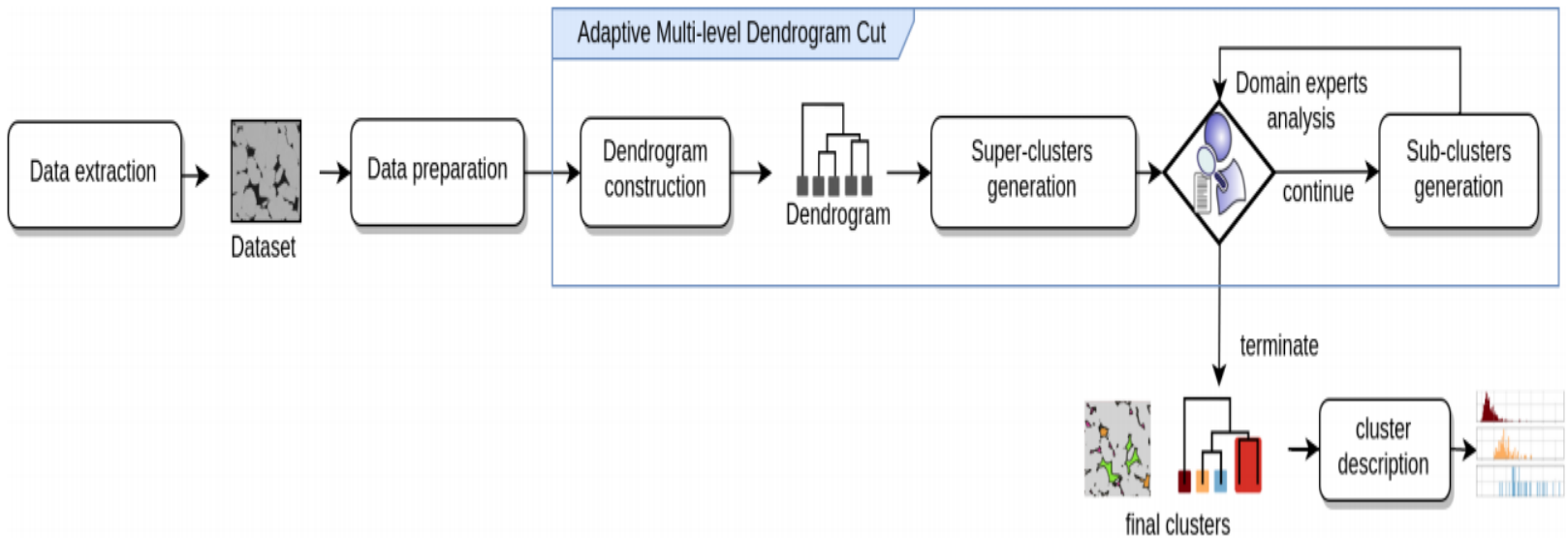


We are merging (or adding) the clusters at each step, right? Hence, this type of clustering is also known as **additive hierarchical clustering.**

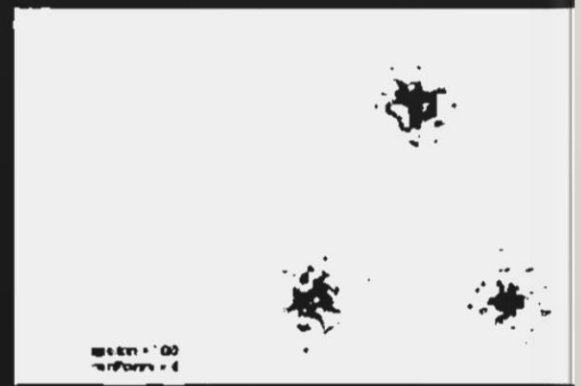
- The Hierarchical clustering Technique can be visualized using a **Dendrogram.**
- A **Dendrogram** is a tree-like diagram that records the sequences of merges or splits.






Adaptive Multi-level Dendrogram Cut
Data extraction
Dataset
Data preparation
Dendrogram construction
Dendrogram
Super-clusters generation
Domain experts analysis
continue
Sub-clusters generation
terminate
final clusters
cluster description

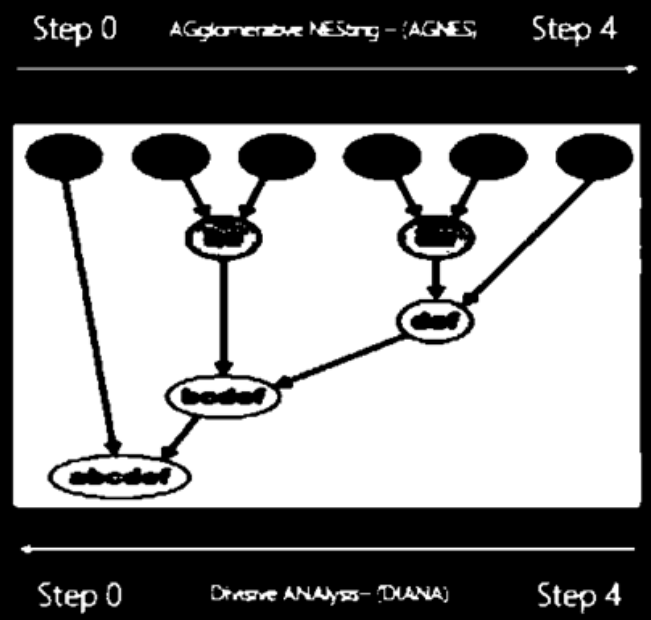
- Cluster Analysis also called as Data Segmentation is an Exploratory Method
- Clustering identifies homogeneous groups of records
- Similar items should be grouped together into homogeneous groups. Within group similarity should be more, i.e., distance among the records/items should be less (within the cluster). High intra-class similarity (Cohesive within clusters)



- Dissimilar items should be placed into heterogeneous groups. Between group dissimilarity should be more, i.e., distance between any two groups should be more. Low inter-class similarity (Distinctive between clusters)

Case	Sex	Glasses	Moustache	Smile	Hat
1	m	y	n	y	n
2	f	n	n	y	n
3	m	y	n	n	n
4	m	n	n	n	n
5	m	n	n	y	n
6	m	n	y	n	y
7	m	y	n	y	n
8	m	n	n	y	n
9	m	y	y	y	n
10	f	n	n	n	n
11	m	n	y	n	n
12	f	n	n	n	n

- Hierarchical Clustering – Agglomerative or Divisive Clustering
  - Begin with 'n' records and find all the distances between records
  - Merge similar records or group of records until all form a large group



Other Hierarchical Clustering Algorithms:

- BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)
- CURE (Clustering Using REpresentatives)
- CHAMELEON

d_{ij} = Distance between records ‘i’ & ‘j’

Distance Requirements:

- Non-negative ($d_{ij} > 0$)
- $d_{ii} = 0$
- Symmetry ($d_{ij} = d_{ji}$)
- Triangle inequality ($d_{ij} + d_{jk} \geq d_{ik}$)

Prospective Applicants

Dean

Univ	SAT	Top10	Accept	SFRatio	Expenses	GradRate
Brown	13?0	89	22	13	22?04	94
CalTech	14?5	100	25	6	63575	81
CMU	126C	62	59	9	25026	?2
Columbia	13?0	?6	24	12	31?0	88
Cornell	128C	83	33	13	21864	90
Dartmouth	1340	89	23	10	32162	95
Duke	13?5	90	30	12	31585	95
Georgetown	1255	?4	24	12	20126	92
Harvard	1400	91	14	1?	39525	9?
JohnsHopkins	1305	?5	44	?	58691	8?
MIT	138C	94	30	10	34870	91
Northwestern	126C	85	39	1?	28052	89
NotreDame	1255	81	42	13	15?22	94
PennState	108?	38	54	18	10?85	80
Princeton	1375	91	14	8	30220	95
Purdue	1005	28	90	19	9066	69
Stanford	136C	90	20	12	36450	93
TexasA&M	1075	49	67	25	8704	67
UCBerkeley	1240	95	40	17	15?40	?8
UChicago	129C	?5	50	13	38380	8?
UMichigan	1?50	65	68	16	15470	85
UPenn	1285	80	36	1?	27553	90
UVA	1225	??	44	14	13349	92
UWisconsin	1085	40	69	15	1?557	?1
Yale	1375	95	19	1?	43514	96

Distance between the records

Notation:

$x_i = \{x_{i1}, x_{i2}, \ldots\ldots\ldots x_{ip}\}$

$x_j = \{x_{j1}, x_{j2}, \ldots\ldots\ldots x_{jp}\}$

Brown = (1310, 89, 22, 13, 22704, 94)

CalTech = (1415, 100, 25, 6, 63575, 81)

$$d_{ij} = \sqrt{(x_{i1} \_ xj_1)^2 + (x_{i2} \_ x_{j2})^2 + \ldots\ldots + (x_{ip} - xjp)^2}$$

$$\sqrt{(1310-1415)^2 + (89-100)^2 + (22-25)^2 + (13-6)^2 + (22704-63575)^2 + (94-81)^2} = 40871.1391$$

Standardize the data:

Brown = (0.40199, 0.64423, -0.87189, 0.06884, -0.32471, 0.80372)

CalTech = (1.37098, 1.21025, -0.71981, -1.65218, 2.50865, -0.63150)

$$Z = \left[\frac{X - Mean}{Stdev}\right]$$

$$\sqrt{(0.40-1.37)^2 + (0.64-1.21)^2 + (-0.87-(-0.71))^2 + (0.06-(-1.65))^2 + (-0.32-2.50)^2 + (0.80-(-0.63))^2} = 3.56$$

Distance matrix

Manhattan Distance

$$d_{ij} = |x_{i1} \_ xj_1| + |x_{i2} - xj_2| + \ldots\ldots\ldots\ldots + |x_{ip} - xjp|$$

Statistical (Mahalanobis) Distance

$$d_{ij} = \sqrt{(x_i - xj)^T * S^{-1} * (x_i - xj)}$$

Where S is the covariance matrix

Distance between binary data

Similarity- based metrics based on 2 x 2 table of counts

	Graduate	Alcoholic	Vice President
Mr. A	Y	Y	Y
Mr. B	N	Y	N
Mr. C	N	N	Y

		Mr. C	
		N	Y
Mr. A	N	0	0
	Y	2	1

		Mr. C	
		N	Y
Mr. A	N	a	b
	Y	c	d

Binary Euclidean Distance: (b + c) / (a + b + c + d)
Simple Matching Coefficient: (a + d) / (a + b + c + d)
Jaccard's Index: d / (b + c + d)

For more than 2 categories, distance = 0 only if both items have same category. Otherwise it will be 1

Distance metrix for mixed data

Numerical & Categorical Data

Step 1: Standardize the numerical variables to [0, 1]
Step 2: Categorical data is converted into dummy variables using one-hot encoding
Step 3: Calculate Euclidean distance using entire data

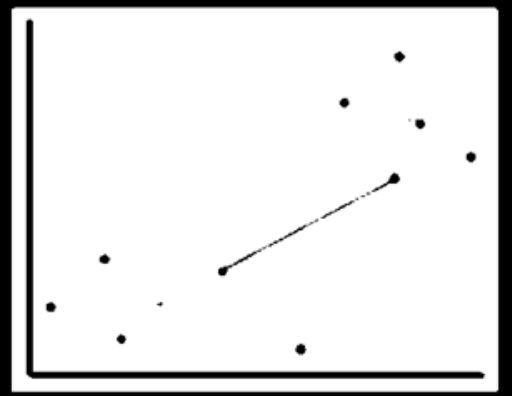
Gower's General Dissimilarity Coefficient

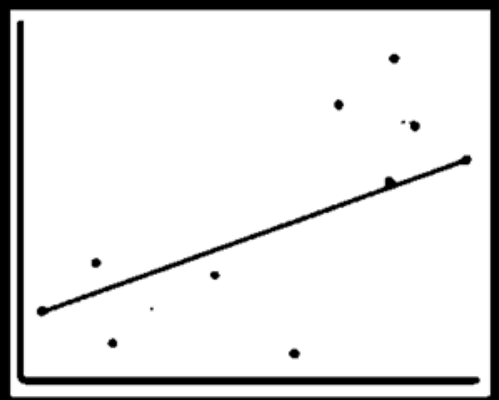
$$d_{ij} = \frac{\sum_k w_{ijk} d_{ijk}}{\sum_k w_{ijk}}$$

d_{ijk} = distance provided by k^{th} variable
w_{ijk} = usually 1 or 0 depending whether or not the comparison is valid for the k^{th} variable

Distances Between Clusters

- Single Linkage is also called as Nearest Neighbour
- Minimum distance between the members of the two clusters

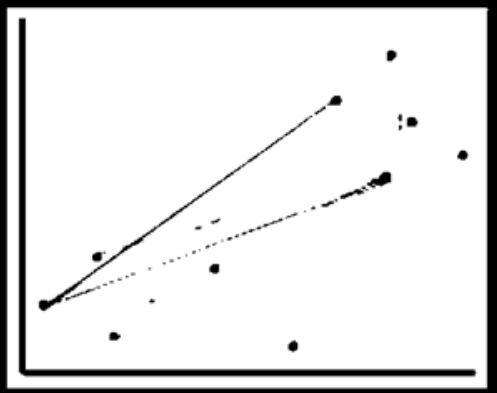




- Complete Linkage is also called as Farthest Neighbour
- Maximum distance between the members of the two clusters

Distances Between Clusters

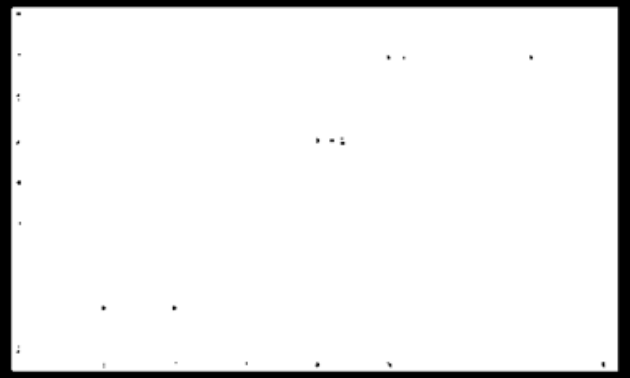
- Average Linkage
- Average of all distances between the members of the two clusters



- Centroid Linkage
- Distance between the centroids of two clusters

Hierarchical Clustering Process

Item	X1	X2
1	1	1
2	2	1
3	4	5
4	7	7
5	5	7



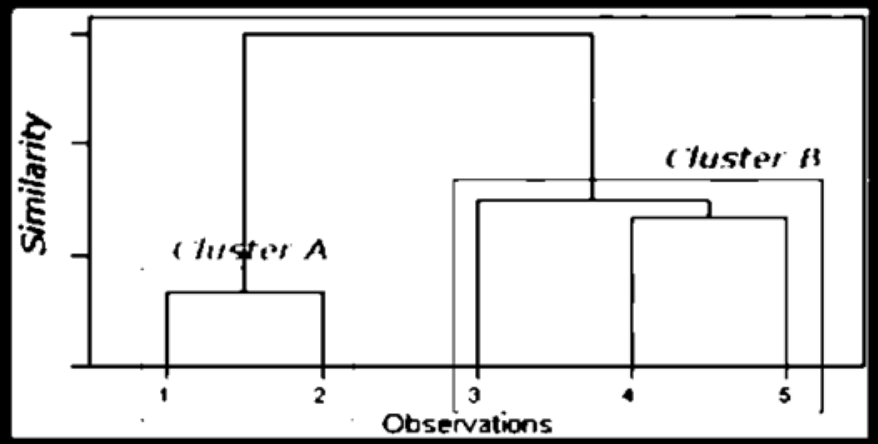
Euclidean Distance Matrix

	1	2	3	4	5
1	0.0				
2	1.0	0.0			
3	5.0	4.5	0.0		
4	8.5	7.8	3.6	0.0	
5	7.2	6.7	2.2	2.0	0.0

	A	3	4	5
A	0.0			
3	4.5	0.0		
4	7.8	3.6	0.0	
5	6.7	2.2	2.0	0.0

	A	3	B
A	0.0		
3	4.5	0.0	
B	6.7	2.2	0.0

	A	B
A	0.0	
B	4.5	0.0



- In hierarchical clustering the number of distance calculations will be C_2^n.
- If n is very large then Hierarchical clustering takes more time
- Hence Hierarchical clustering is not suitable for large data set
- On the contrary in K-means,we are defining number of cluster well before and hence less time required for computing.

Introduction to K-Means Clustering

- It states that the points within a cluster should be similar to each other. So, **our aim here is to minimize the distance between the points within a cluster.**
- *There is an algorithm that tries to minimize the distance of the points in a cluster with their centroid – the k-means clustering technique.*
- ***The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.***

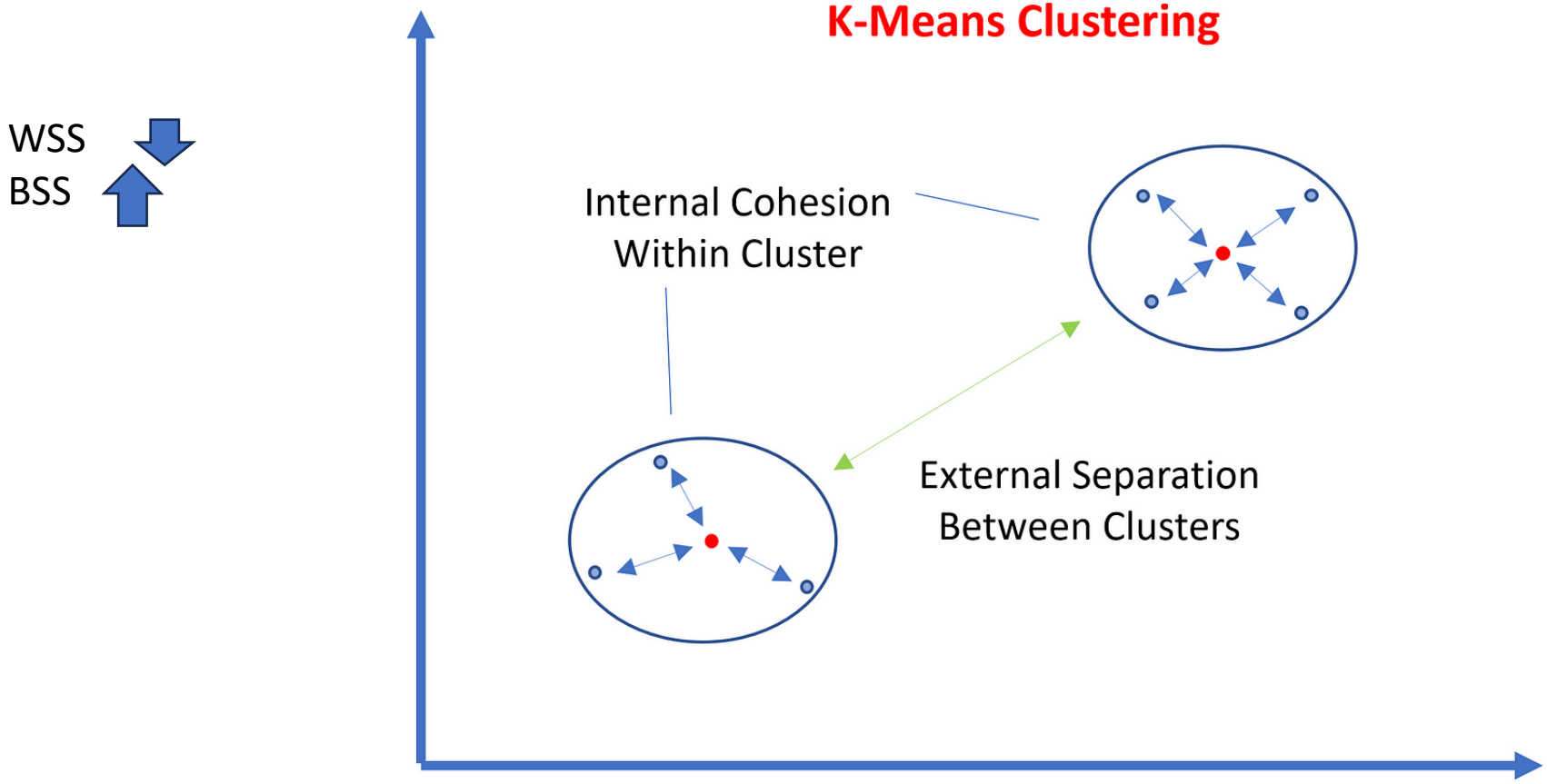
- Simply speaking k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group.
- K is positive integer number.
- The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

K-Means Clustering Steps

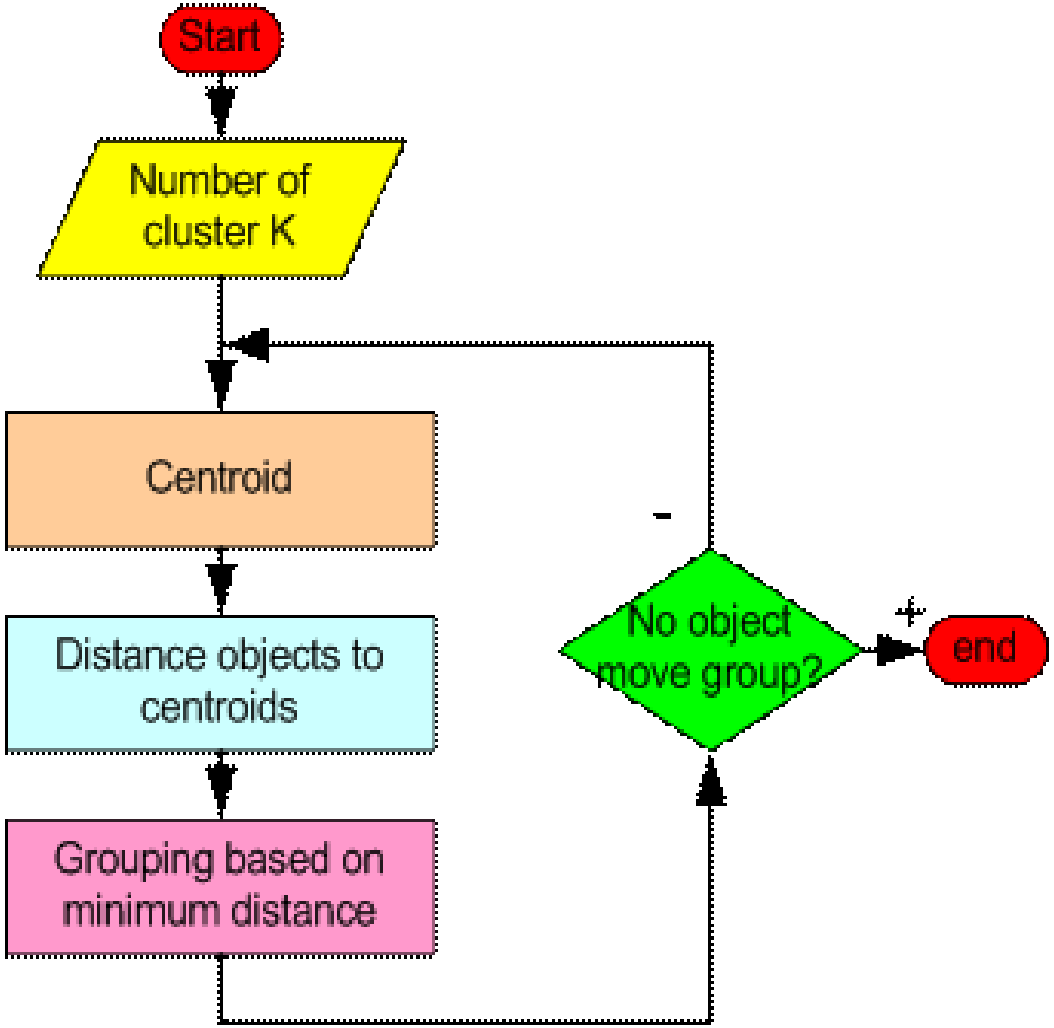
- K-means works based on optimization algorithm and minimizes the within sum of square (within cluster variance)
- Steps:

1. Select 'K' data points as 'K' cluster centroids
2. Find the distance between all the data points to all the 'K' centroids
3. Assign each of the record / data point to one of the K centroids based on nearest distance
4. Re-calculate the new centroids (K of them) by taking average of the points which form a cluster
5. Repeat the steps 3 & 4 until the algorithm stops converging
6. Give a logical name to each cluster

WSS :Within sum of squares
BSS: Between sum of square



How the K-Mean Clustering algorithm works?



- **Step 1:** Begin with a decision on the value of k = number of clusters .
- **Step 2**: Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly,or systematically as the following:

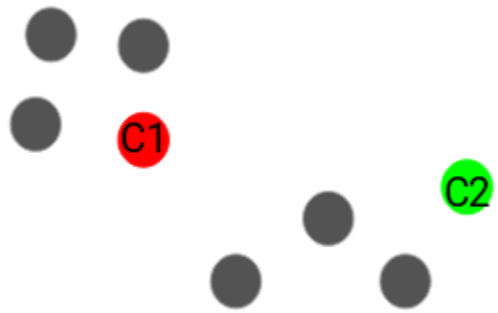
  1.Take the first k training sample as single- element clusters

  2. Assign each of the remaining (N-k) training sample to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.

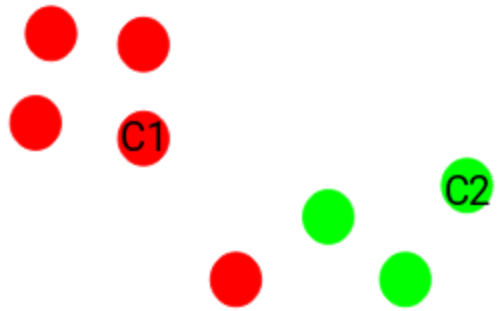
- **Step 3:** Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.
- **Step 4 .** Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

- Let’s now take an example to understand how K-Means actually works:
- We have these 8 points and we want to apply k-means to create clusters for these points. Here’s how we can do it.

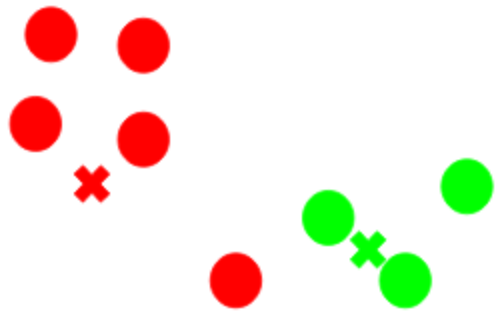
- **Step 1: Choose the number of clusters *k***
- The first step in k-means is to pick the number of clusters, k.
- **Step 2: Select k random points from the data as centroids**
- Next, we randomly select the centroid for each cluster. Let's say we want to have 2 clusters, so k is equal to 2 here. We then randomly select the centroid:Here, the red and green circles represent the centroid for these clusters.



- **Step 3: Assign all the points to the closest cluster centroid**
- Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.
- Here you can see that the points which are closer to the red point are assigned to the red cluster whereas the points which are closer to the green point are assigned to the green cluster.
-



- **Step 4: Recompute the centroids of newly formed clusters**
- Now, once we have assigned all of the points to either cluster, the next step is to compute the centroids of newly formed clusters:Here, the red and green crosses are the new centroids.



- **Step 5: Repeat steps 3 and 4**
- We then repeat steps 3 and 4:

Stopping Criteria for K-Means Clustering

There are essentially three stopping criteria that can be adopted to stop the K-means algorithm:

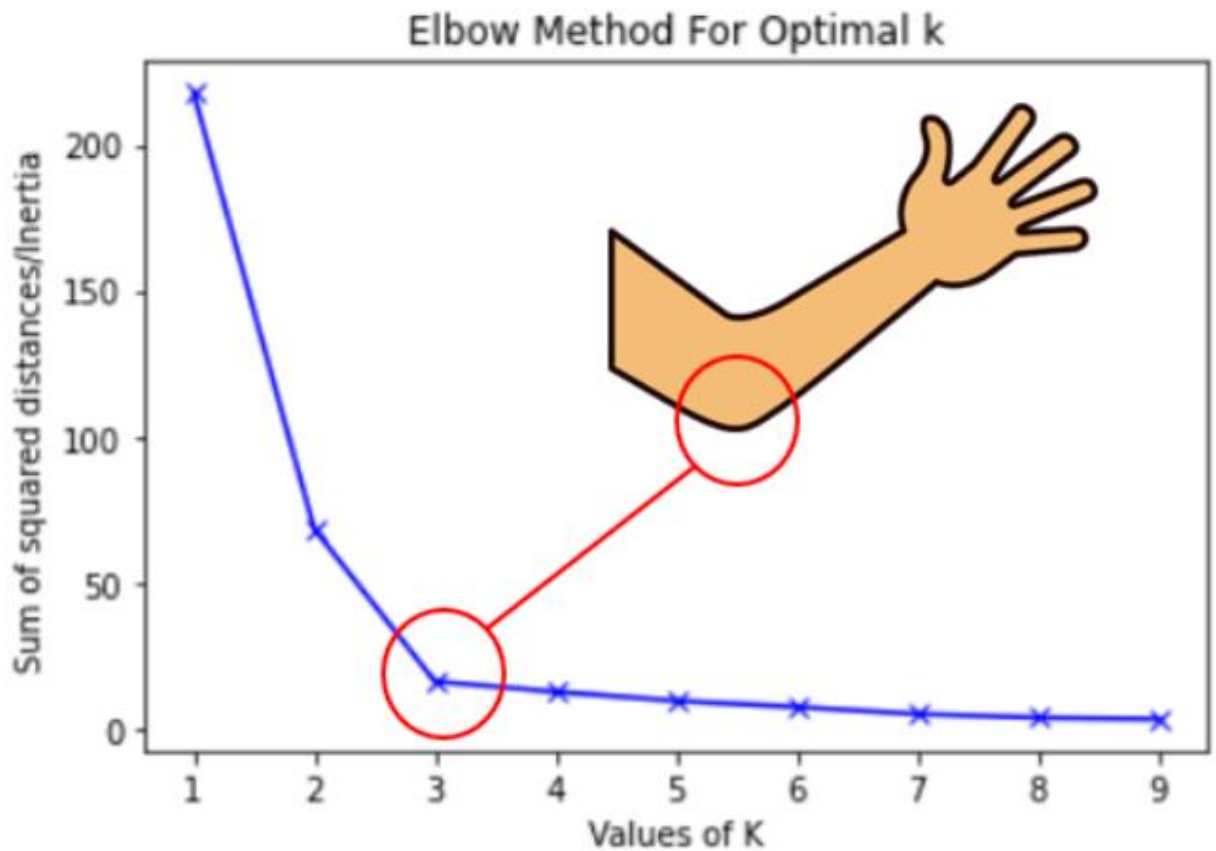
- Centroids of newly formed clusters do not change
- Points remain in the same cluster
- Maximum number of iterations are reached
- We can stop the algorithm if the centroids of newly formed clusters are not changing. Even after multiple iterations, if we are getting the same centroids for all the clusters, we can say that the algorithm is not learning any new pattern and it is a sign to stop the training.

Weaknesses of K-Mean Clustering

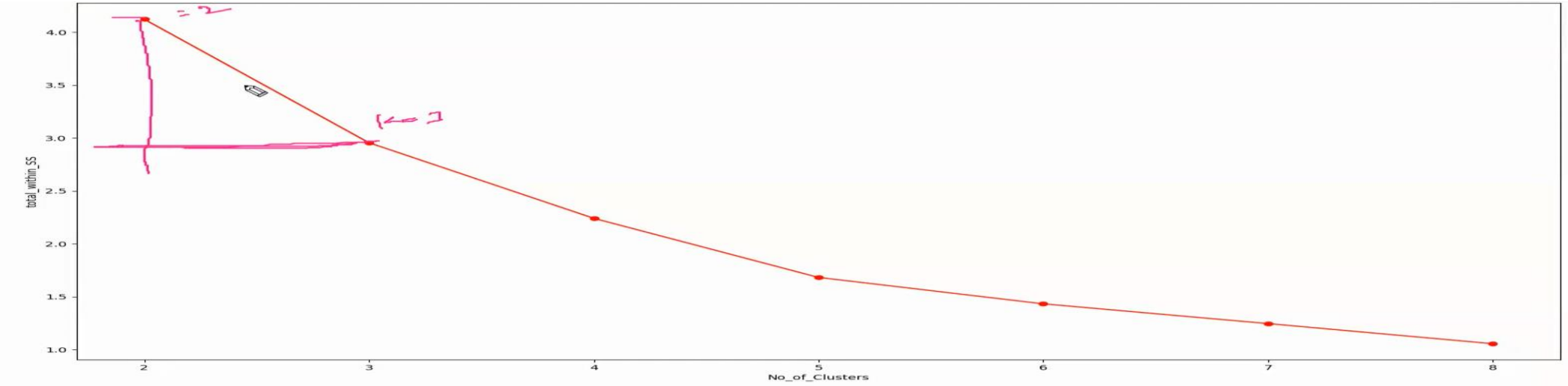
1. When the numbers of data are not so many, initial grouping will determine the cluster significantly.
2. The number of cluster, K, must be determined before hand. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments.
3. We never know the real cluster, using the same data, because if it is inputted in a different order it may produce different cluster if the number of data is few.
4. It is sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the *local optimum*.

K-Means Clustering

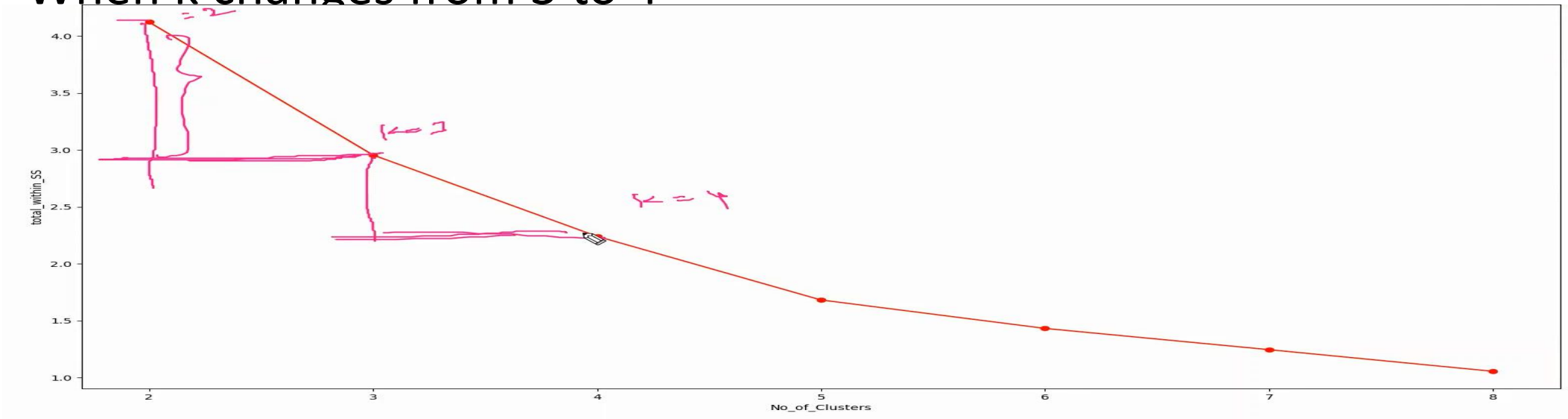
- Upfront the number of clusters must be decided based on:
  - $\sqrt{\frac{n}{2}}$ thumb rule
  - Scree Plot or Elbow Curve
  - Domain knowledge and customer requirements
  - Within Sum of Square & Between Sum of Square measures
- Useful for large datasets
- No Dendrogram

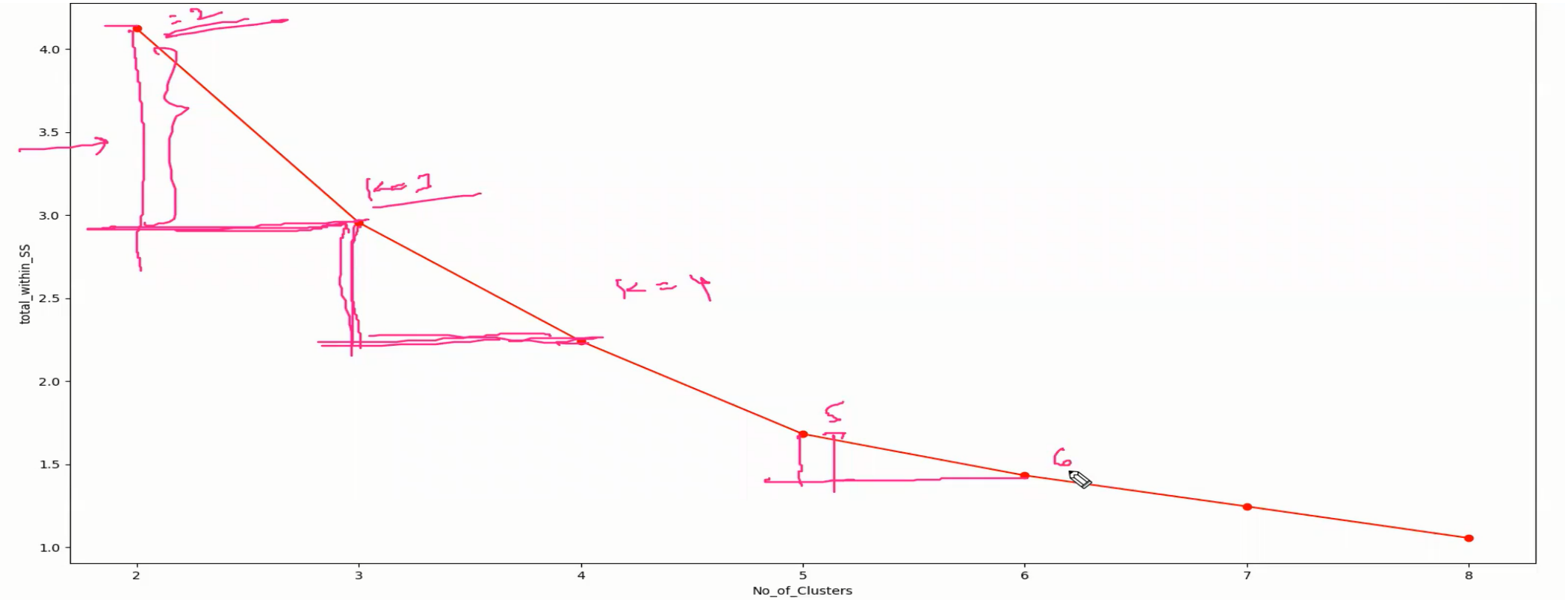


Line plot between K and inertia



- When k changes from 2 to 3,then decrease in kwss is higher than
- When k changes from 3 to 4





- When k values changes from 5 to 6 decrease in twss is considerably less,hence considered k=3

You can try k=3,4 and 5 whichever is giving better result ,you can finalize the value