UNSUPERVISED LEARNING

Unsupervised Learning:

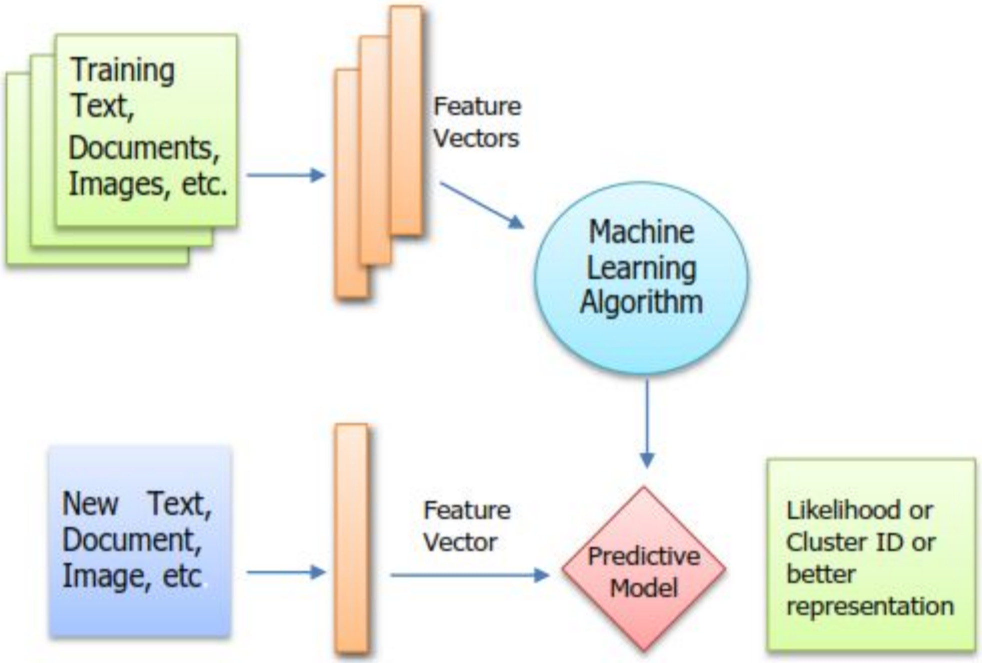
→ The model is not provided with the correct results during the training.

→ Can be used to cluster the input data in classes on the basis of their statistical properties only cluster significance and labeling.

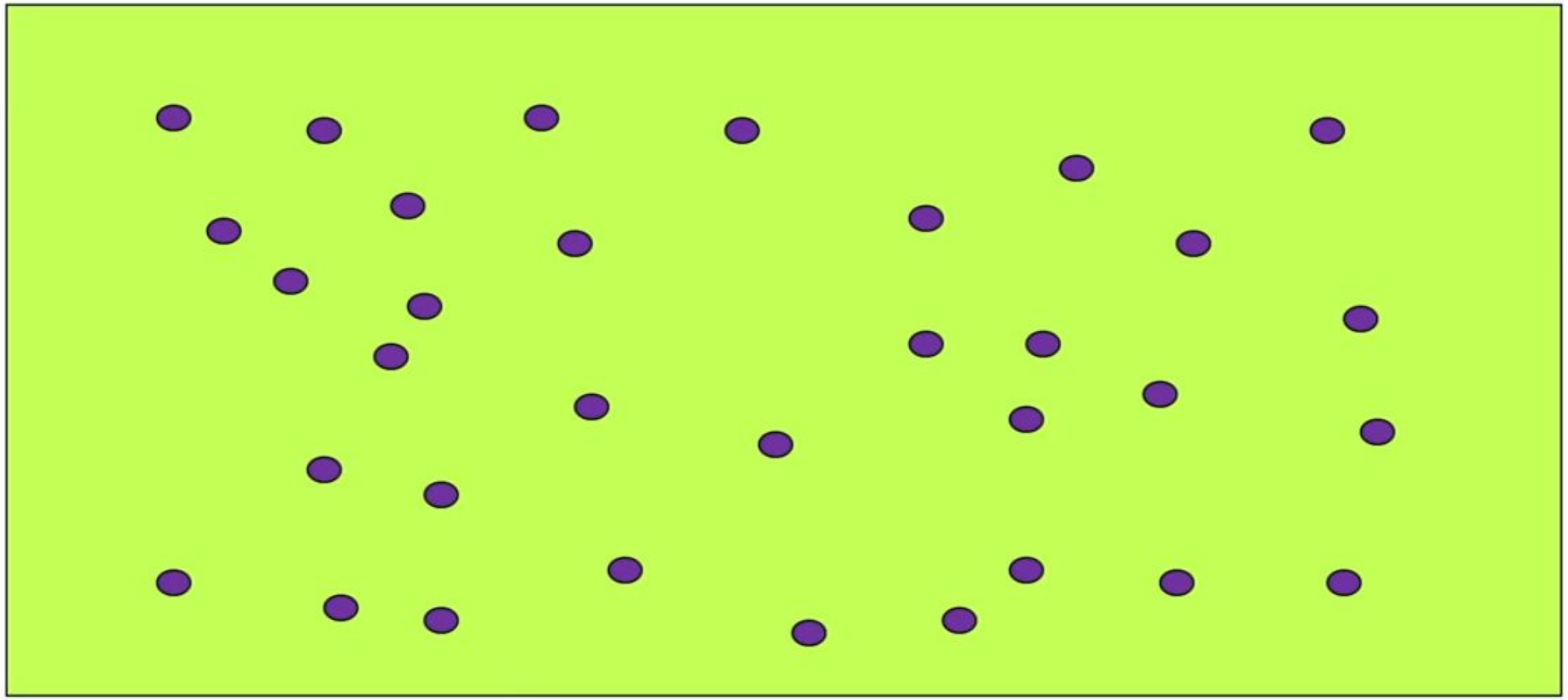
→ The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes.

We don’t have outputs defined for the given inputs, but we need to group data into different clusters for classification, we use Unsupervised Learning

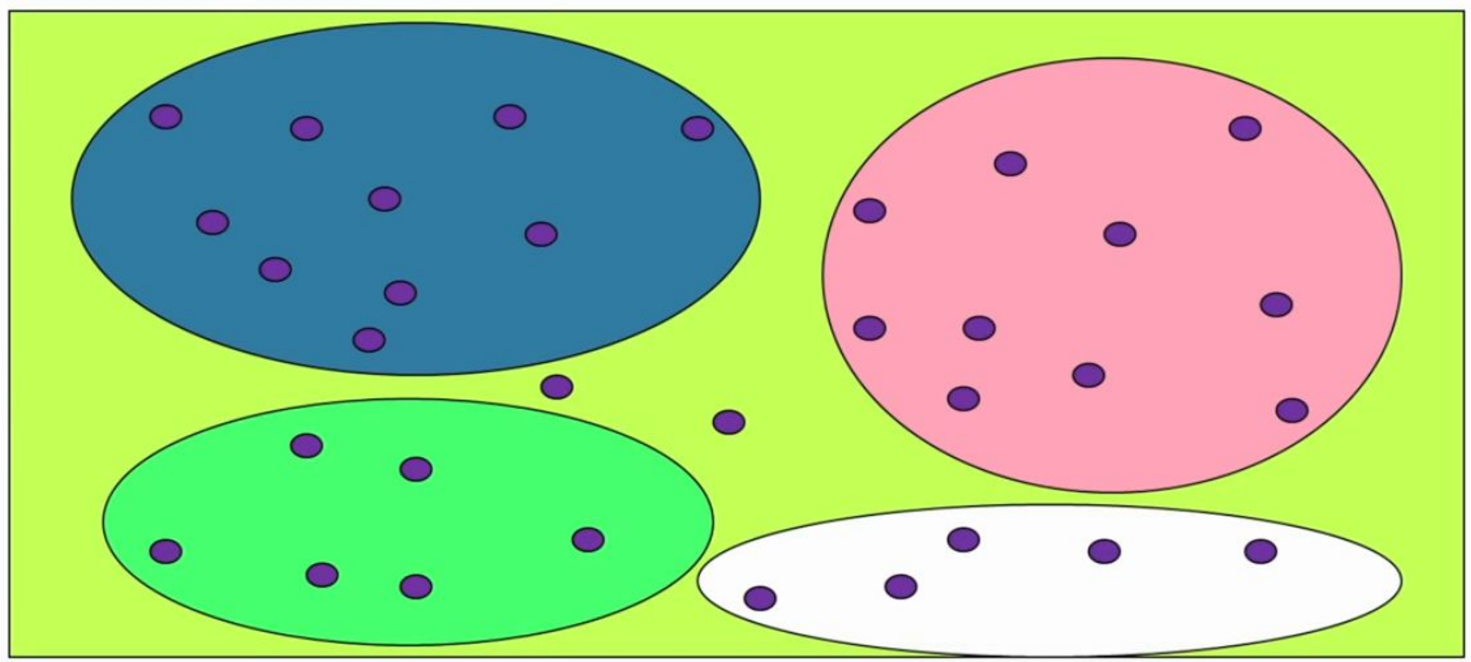
The flowchart:



Unlabelled Training Data



Possible Clusters



We are finding cohesive groups of points and points which are outliers
I need some form of bias: here I have assumed all clusters as ellipsoids
Not all points fall into clusters

Applications

- Customer data (targeted promotions for group of customers)
  - Discover classes of customers
- Image pixels (segmentation)
  - Discover regions
- Words
  - Synonyms
- Documents
  - Topics

The following scenarios implement Clustering:

→ A telephone company needs to establish its network by putting its towers in a particular region it has acquired. The location of putting these towers can be found by clustering algorithm so that all its users receive maximum signal strength.

→ Cisco wants to open its new office in California. The management wants to be cordial to its employees and want their office in a location so that its employees' commutation is reduced to minimum.

→ The Miami DEA wants to make its law enforcement more stringent and hence have decided to make their patrol vans stationed across the area so that the areas of high crime rates are in vicinity to the patrol vans.

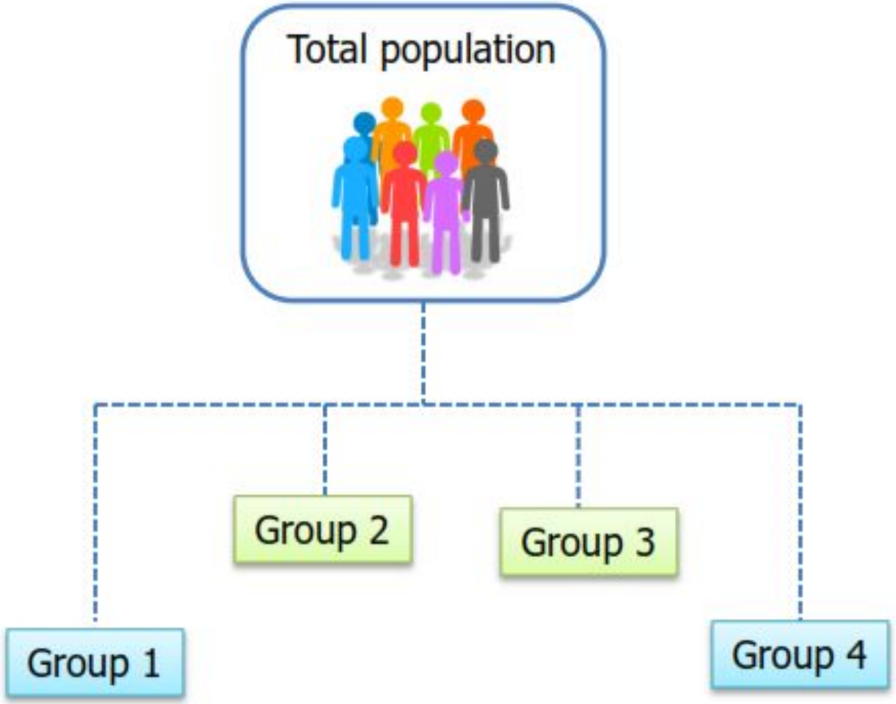
→ A Hospital Care chain wants to open a series of Emergency-Care wards, keeping in mind the factor of maximum accident prone areas in a region.

Performance tuning technique number 41:
clustering to reduce overheads

Why Clustering?

Organizing data into clusters such that there is:

→ High intra-cluster similarity
→ Low inter-cluster similarity
→ Informally, finding natural groupings among objects



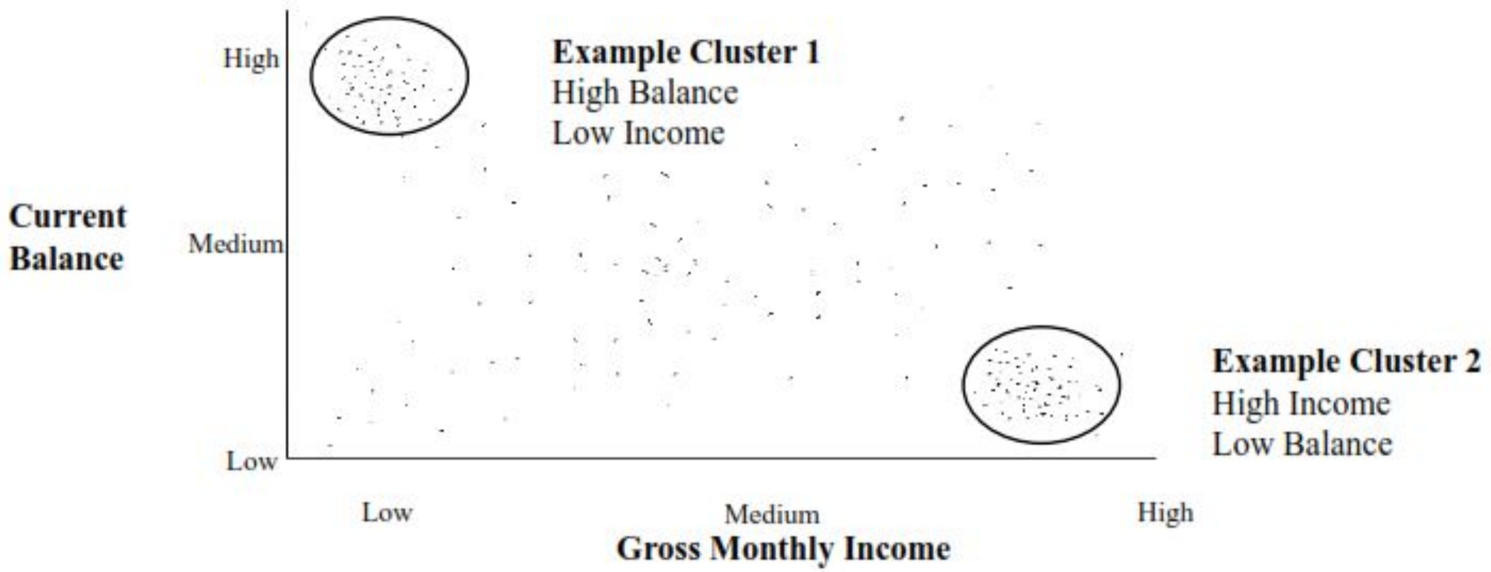
→ The process by which objects are classified into a number of groups so that they are as much dissimilar as possible from one group to another group, but as much similar as possible within each group.

→ The objects in group 1 should be as similar as possible.

→ But there should be much difference between an object in group 1 and group 2.

→ The attributes of the objects are allowed to determine which objects should be grouped together.

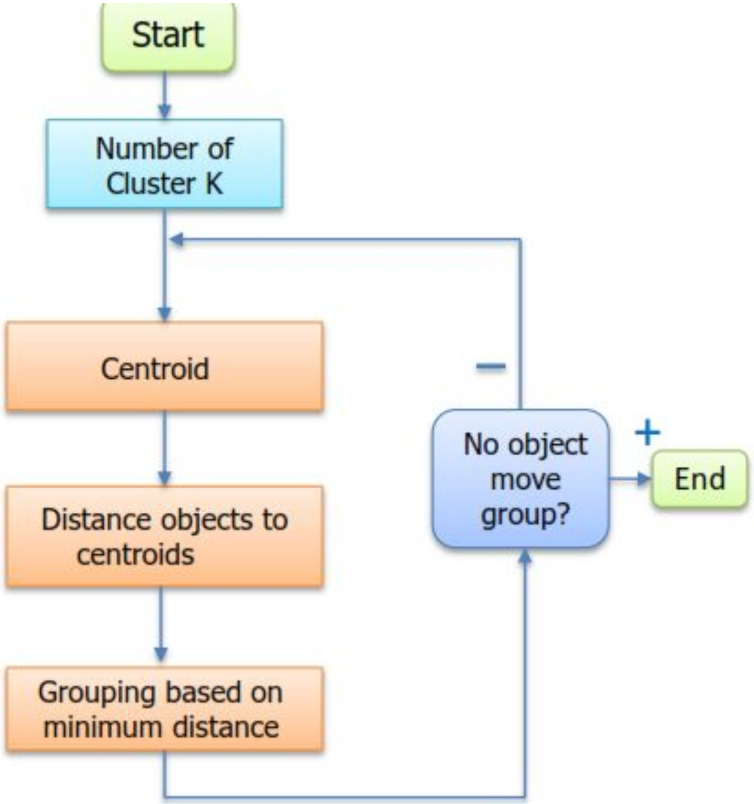
Basic concepts of Cluster Analysis using two variables



→ Cluster 1 and Cluster 2 are being differentiated by Income and Current Balance.

→ The objects in Cluster 1 have similar characteristics (High Income and Low balance), on the other hand the objects in Cluster 2 have the same characteristic (High Balance and Low Income).

→ But there are much differences between an object in Cluster 1 and an object in Cluster 2.



Iterate until *stable* (cluster centers converge):

1. Determine the centroid coordinate.

2. Determine the distance of each object to the centroids.

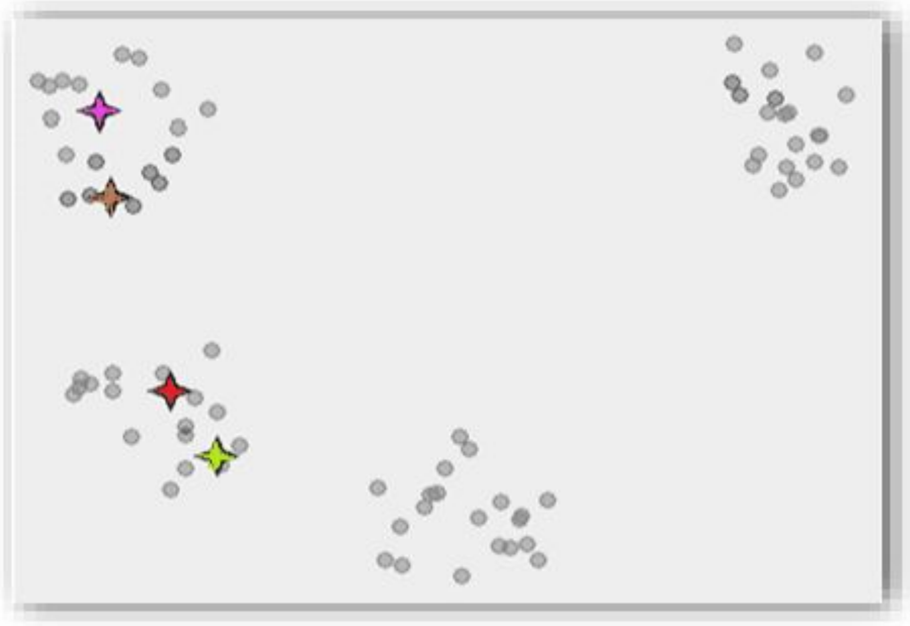
3. Group the object based on minimum distance (find the closest centroid)

Problem Statement:

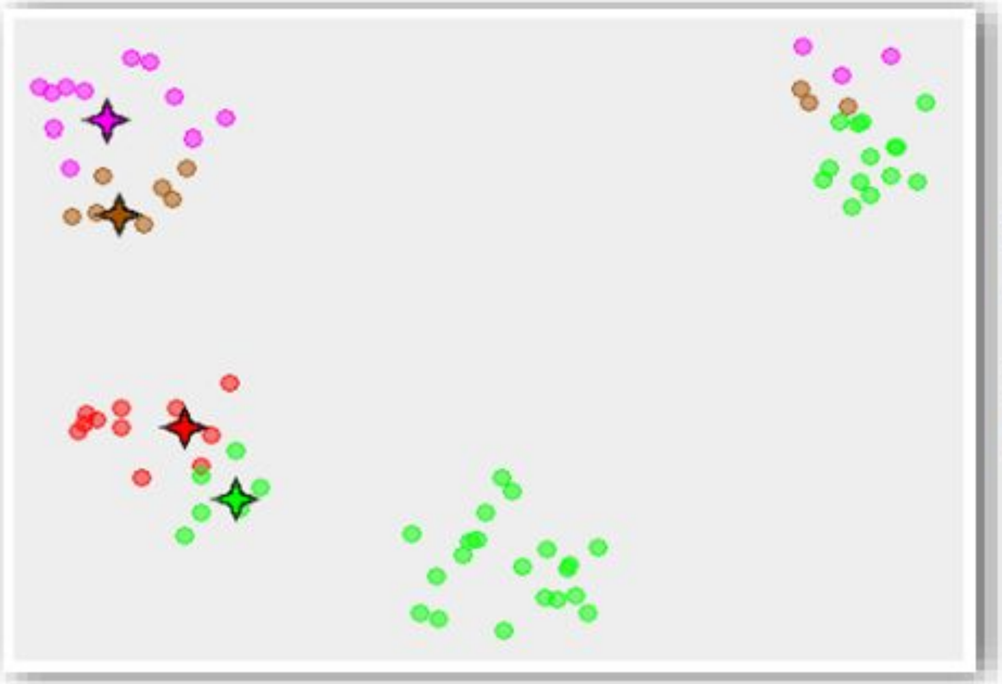
The newly appointed Governor has finally decided to do something for the society and wants to open a chain of schools across a particular region, keeping in mind the distance travelled by children is minimum, so that the percentage turnout is more.

Poor fella cannot decide himself and has asked its Data Science team to come up with the solution.

Bet, these guys have the solution to almost everything!!

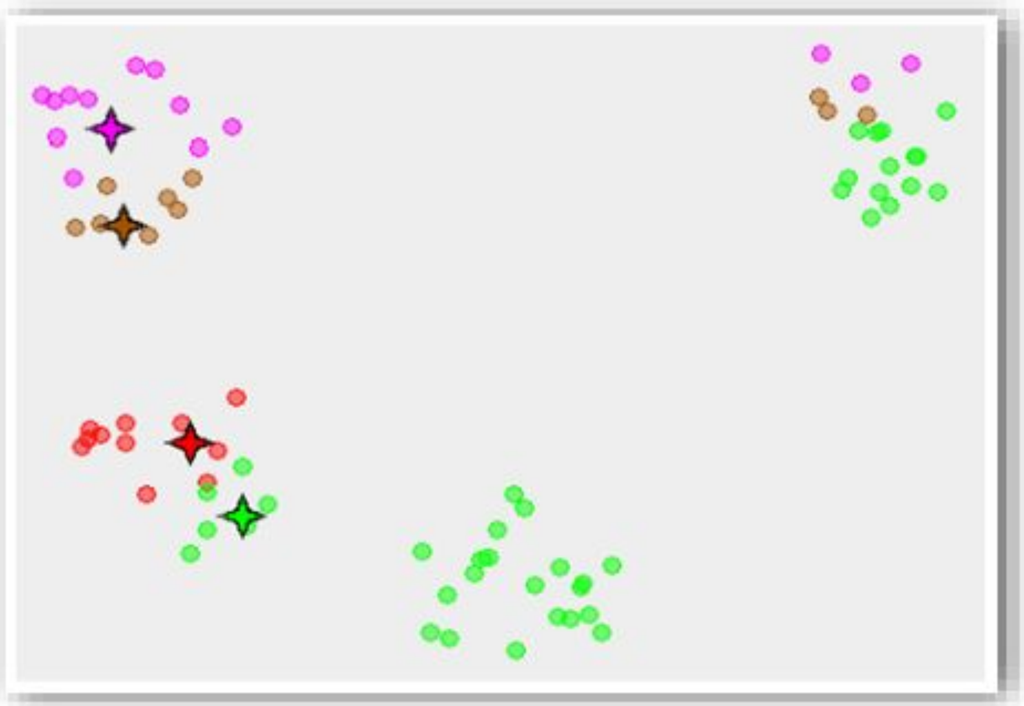


1. If k=4, we select 4 random points in the 2d space and assume them to be cluster centers for the clusters to be created.



2. We take up a random data point from the space and find out its distance from all the 4 clusters centers.

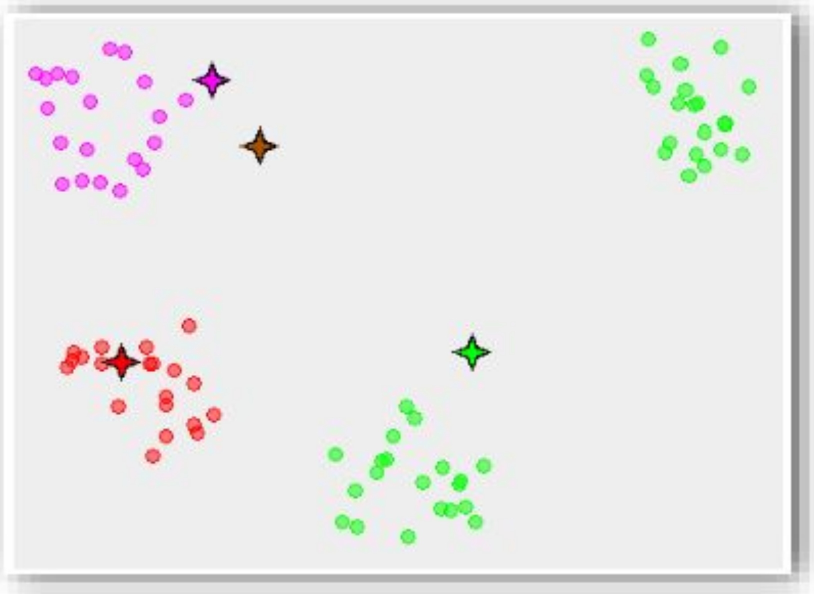
If the data point is closest to the pink cluster center, it is colored pink.



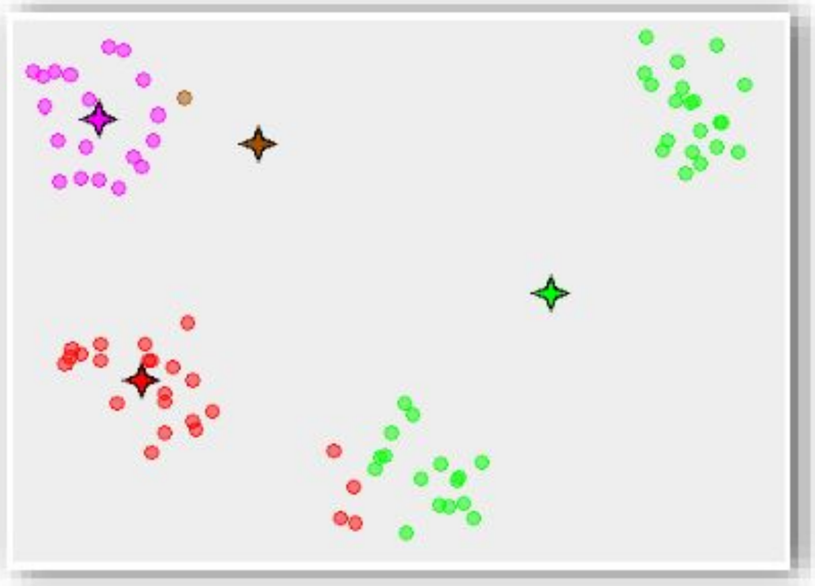
3. Now we calculate the centroid of all the pink points and assign that point as the cluster center for that cluster.

Similarly, we calculate centroids for all the 4 colored(clustered) points and assign the new centroids as the cluster centers.

4. Step-2 and step-3 are run iteratively, unless the cluster centers converge at a point and no longer move.

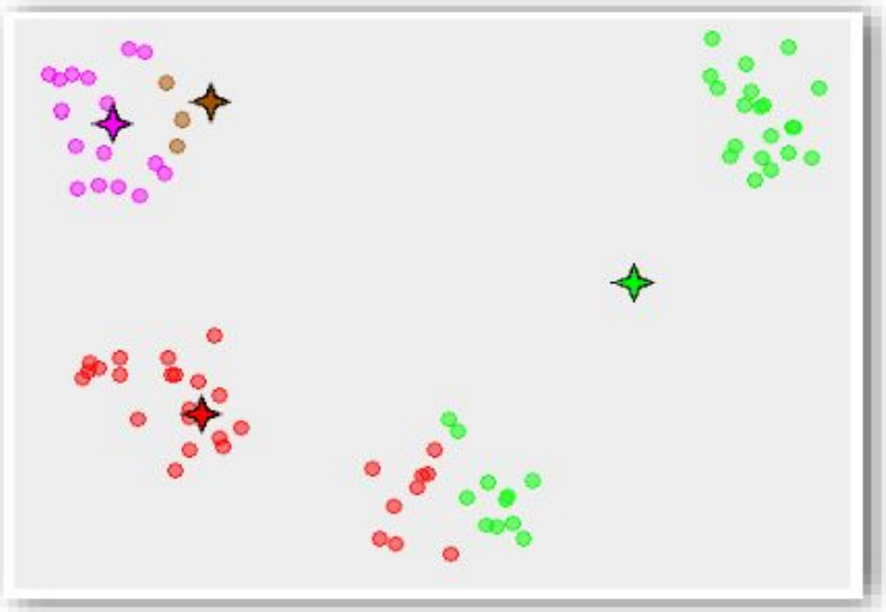


Iteration-1

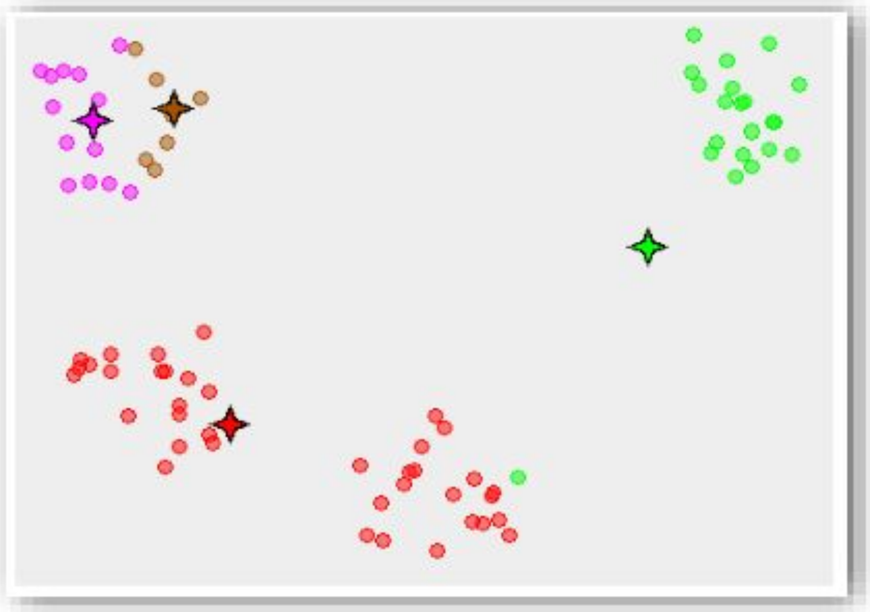


Iteration-2

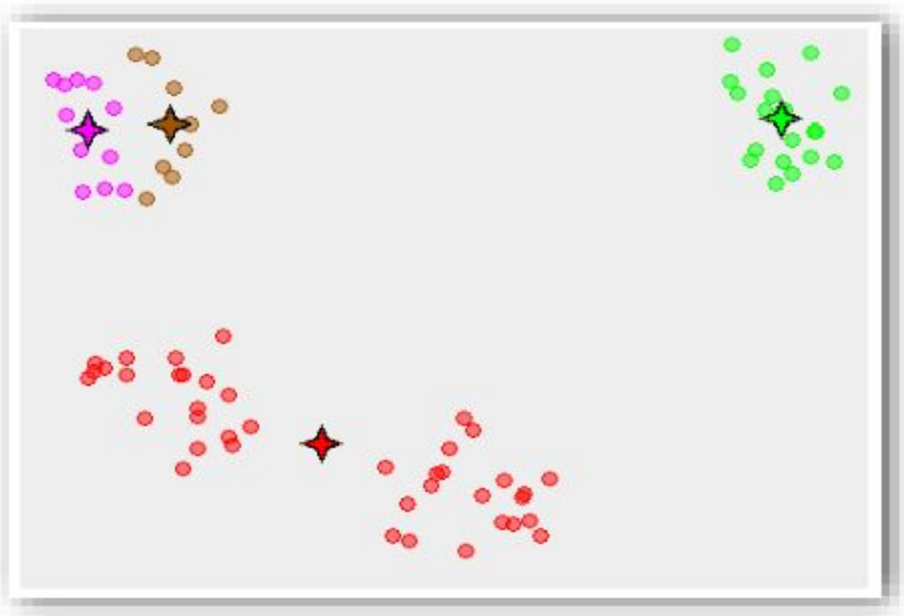
5. We can see that the cluster centers are still not converged so we go ahead and iterate it more.



Iteration-3



Iteration-4



Finally, after multiple iterations, we reach a stage where the cluster centers coverge and the clusters look like as:

Here we have performed:

Iterations: 5

Mathematical Formula

$D=\{\mathbf{x_1},\mathbf{x_2},\ldots,\mathbf{x_i},\ldots,\mathbf{x_m}\}$ → data set of m records

$\mathbf{x_i}=(x_{i1},x_{i2},\ldots,x_{in})$ → each record is an n-dimensional vector

$$\mathbf{c_i} = \text{cluster}(\mathbf{x_i}) = \arg\min_j \left\| x_i - \mu_j \right\|^2$$

$$\textit{Distortion} = \sum_{i=1}^{m}(x_i - c_i)^2 = \sum_{j=1}^{k} \sum_{i \in OwnedBy(\mu_j)} (x_i - \mu_j)^2$$

(within cluster sum of squares)

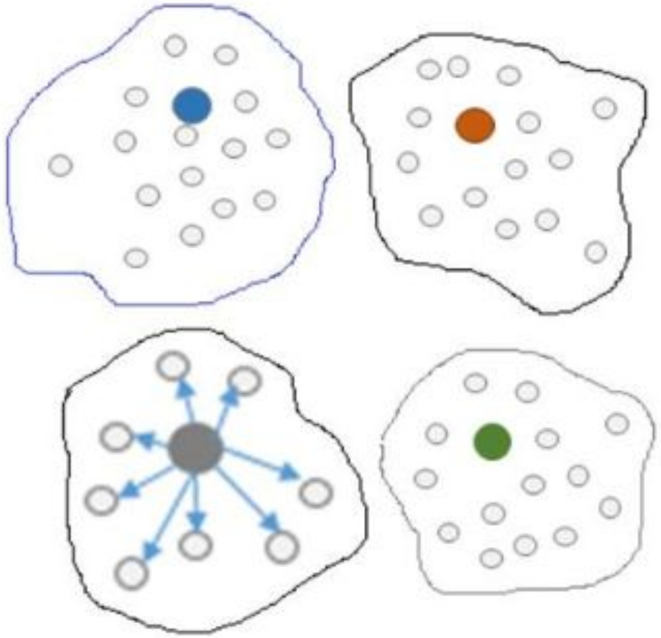
Owned By(.): set of records that belong to the specified cluster center

Goal: Find cluster centers that *minimize* Distortion

Solution can be found by setting the partial derivative of Distortion w.r.t. each cluster center to zero

$$\frac{\partial \text{ Distortion}}{\partial \mu_j} = \frac{\partial}{\partial \mu_j} \sum_{i \in OwnedBy(\mu_j)} (x_i - \mu_j)^2 = -2 \sum_{i \in OwnedBy(\mu_j)} (x_i - \mu_j) = 0 \text{ (for minimum)}$$

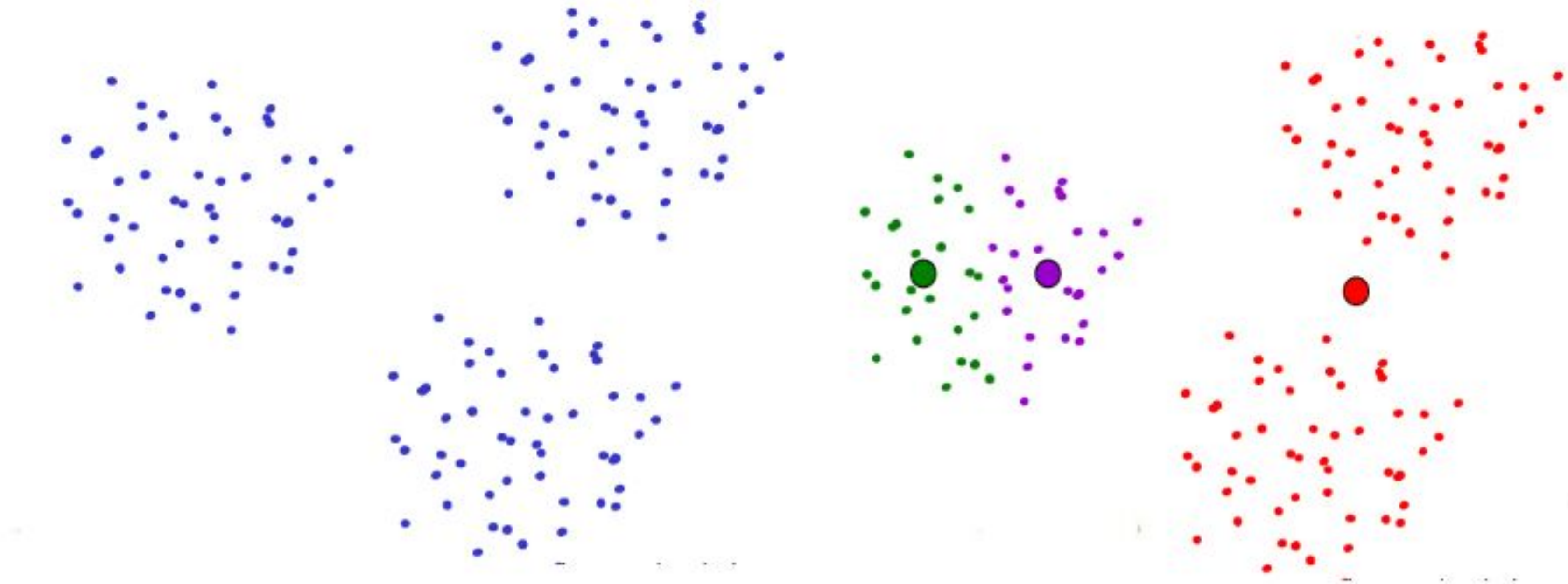
$$\Rightarrow \mu_j = \frac{1}{|OwnedBy(\mu_j)|} \sum_{i \in OwnedBy(\mu_j)} x_i$$



Not necessarily!

We might get stuck in local minimum, and not a global minimum

Try to come up with a converged solution, but does not have minimum distortion:



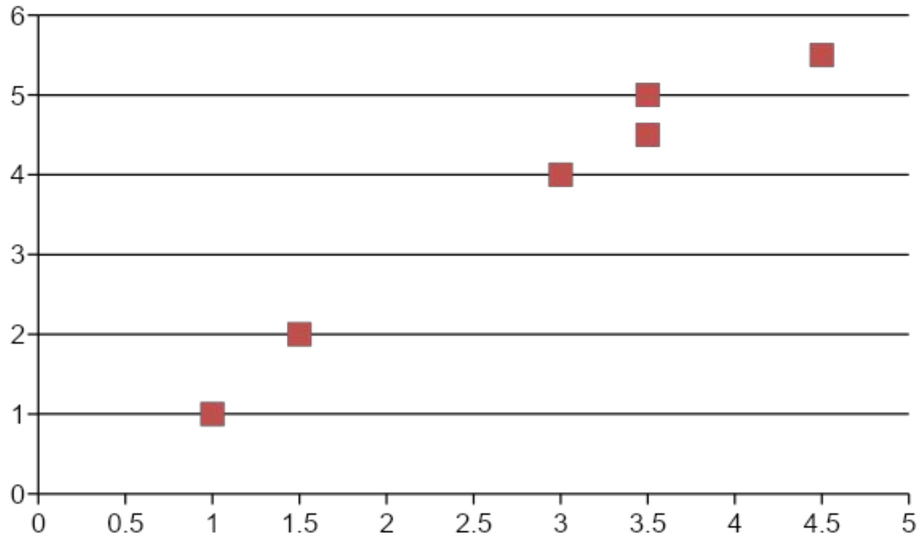
How to get optimal solution

Idea 1: careful about where we start

→ Choose first center at random
→ Choose second center that is far away from the first center
→ … Choose j^{th} center as far away as possible from the closest of centers 1 through (j-1)

Idea 2: Do many runs of K-means, each with different random starting point

I	X1	X2
A	1	1
B	1.5	2
C	3	4
D	4.5	5.5
E	3.5	5
F	3.5	4.5



Let’s take 2 points as cluster centers for the k=2

We can take point A and point B randomly

Now find distance of each point from these cluster centers

e.g. Distance of B from A = $((1.5-1)^2 + (2-1)^2)^{1/2}$ = 1.118

Distance of C from A = $((3-1)^2 + (4-1)^2)^{1/2}$ = 3.6 and so on…

i	Cluster1	Cluster2
A	0	1.18
B	1.18	0
C		
D		
E		
F		
G		

Example 2

Cluster the following eight points (with (x, y) representing locations) into three clusters:

A1(2, 10), A2(2, 5), A3(8, 4), A4(5, 8), A5(7, 5), A6(6, 4), A7(1, 2), A8(4, 9)

Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2).

The distance function between two points a = (x1, y1) and b = (x2, y2) is defined as-

$$P(a, b) = |x2 - x1| + |y2 - y1|$$

Use K-Means Algorithm to find the three cluster centers after the second iteration.

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (5, 8) of Cluster-02	Distance from center (1, 2) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	5	9	C1
A2(2, 5)	5	6	4	C3
A3(8, 4)	12	7	9	C2
A4(5, 8)	5	0	10	C2
A5(7, 5)	10	5	9	C2
A6(6, 4)	10	5	7	C2
A7(1, 2)	9	10	0	C3
A8(4, 9)	3	2	10	C2

From here, New clusters are-

Cluster-01:

First cluster contains points-

- A1(2, 10)

Cluster-02:

Second cluster contains points-

- A3(8, 4)
- A4(5, 8)
- A5(7, 5)
- A6(6, 4)
- A8(4, 9)

Cluster-03:

Third cluster contains points-

- A2(2, 5)
- A7(1, 2)

Now,

- We re-compute the new cluster clusters.
- The new cluster center is computed by taking mean of all the points contained in that cluster.

For Cluster-01:

- We have only one point A1(2, 10) in Cluster-01.
- So, cluster center remains the same.

For Cluster-02:

Center of Cluster-02

= ((8 + 5 + 7 + 6 + 4)/5, (4 + 8 + 5 + 4 + 9)/5)

= (6, 6)

For Cluster-03:

Center of Cluster-03

= ((2 + 1)/2, (5 + 2)/2)

= (1.5, 3.5)

This is completion of Iteration-01.

Iteration-02:

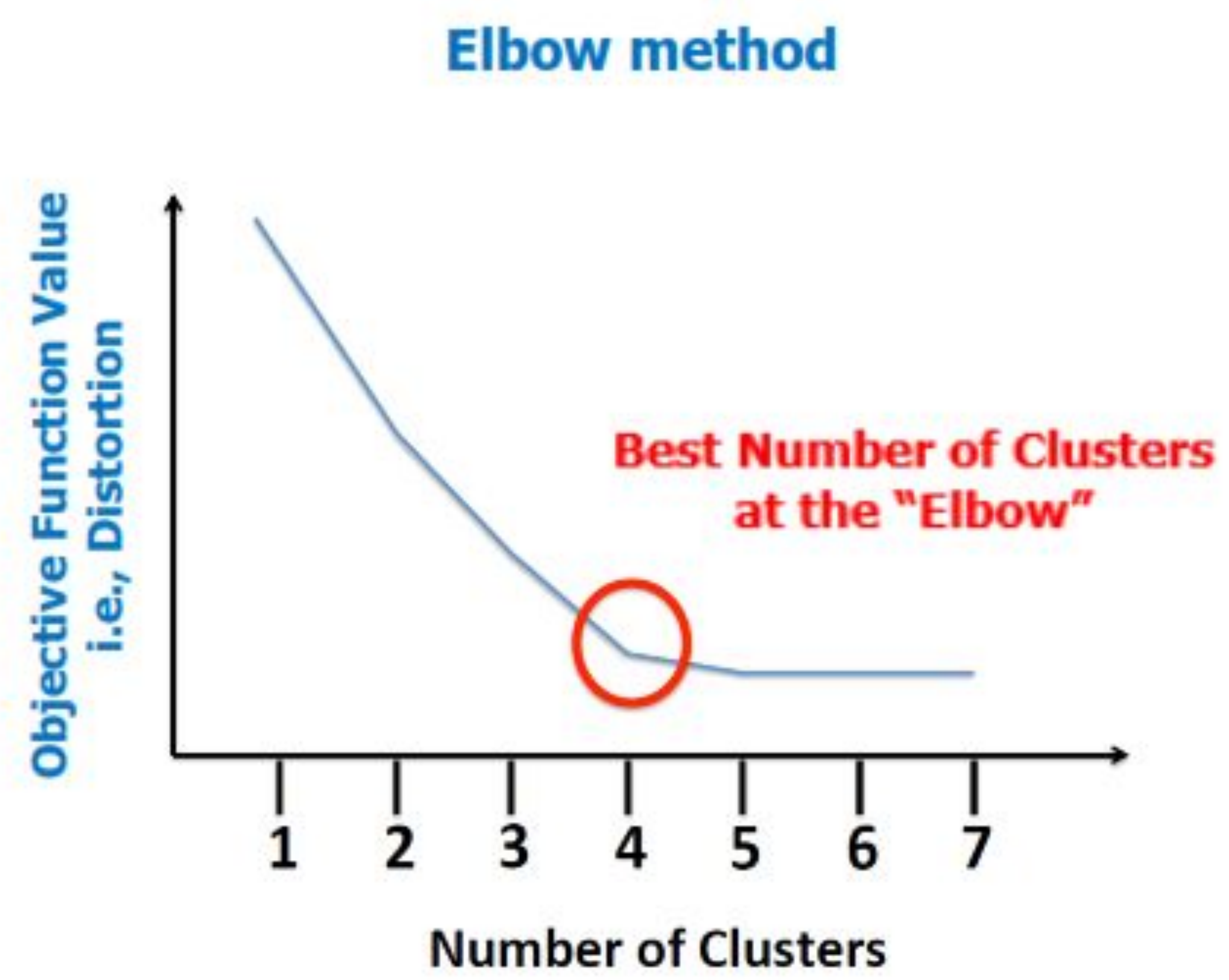
- We calculate the distance of each point from each of the center of the three clusters.
- The distance is calculated by using the given distance function.

The following illustration shows the calculation of distance between point A1(2, 10) and each of the center of the three clusters-

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (6, 6) of Cluster-02	Distance from center (1.5, 3.5) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	8	7	C1
A2(2, 5)	5	5	2	C3
A3(8, 4)	12	4	7	C2
A4(5, 8)	5	3	8	C2
A5(7, 5)	10	2	7	C2
A6(6, 4)	10	2	5	C2
A7(1, 2)	9	9	2	C3
A8(4, 9)	3	5	8	C1

R CODE

Choosing number of clusters



Perform k-means clustering on a data matrix.

- kmeans(x, centers, iter.max = 10, nstart = 1, algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"), trace=FALSE) ## S3 method for class 'kmeans' fitted
- (object, method = c("centers", "classes"), ...)
- **Arguments**
- x: numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a data frame with all numeric columns).
- centers: either the number of clusters, say *k*, or a set of initial (distinct) cluster centres.
- iter.max : the maximum number of iterations allowed.
- nstart: if centers is a number, how many random sets should be chosen?
- algorithm character: may be abbreviated. Note that "Lloyd" and "Forgy" and alternative names for one algorithm.
- object: an **R** object of class "kmeans", typically the result ob of ob <- kmeans(..).
- Method: character: may be abbreviated. "centers" causes fitted to return cluster centers (one for each input point) and"classes" causes fitted to return a vector of class assignments.

Value

- kmeans returns an object of class "kmeans" which has a print and a fitted method. It is a list with at least the following components:
- Cluster: A vector of integers (from 1:k) indicating the cluster to which each point is allocated.
- Centers: A matrix of cluster centres.
- Totss: The total sum of squares.
- Withinss: Vector of within-cluster sum of squares, one component per cluster.
- tot.withinss: Total within-cluster sum of squares, i.e., sum(withinss).
- Betweenss: The between-cluster sum of squares, i.e. totss-tot.withinss.
- Size: The number of points in each cluster.
- Iter: The number of (outer) iterations.
- ifaultinteger: indicator of a possible algorithm problem – for experts.

R code

```
i<-iris
View(i)
i1<- i[,1]
View(i1)
i2<- i[,c(1,2)]
i3<- i[,c(1,2,3)]
i4<- i[,c(1,2,3,4)]
View(i2)
View(i3)
View(i4)

c1<-kmeans(i1,3)
c2<-kmeans(i2,3)
c3<-kmeans(i3,3)
c4<-kmeans(i4,3)
```

```
ic1<- data.frame(i1,i$Species,c1$cluster)
View(ic1)

ic2<- data.frame(i2,i$Species,c2$cluster)
View(ic2)

ic3<- data.frame(i3,i$Species,c3$cluster)
View(ic3)

ic4<-data.frame(i4,i$Species,c4$cluster)
View(ic4)
```

Example output: c2

- K-means clustering with 3 clusters of sizes 50, 47, 53 Cluster means:
- Sepal.Length Sepal.Width
- 1 5.006000 3.428000
- 2 6.812766 3.074468
- 3 5.773585 2.692453
- Clustering vector: [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 [47] 1 1 1 1 2 2 2 3 2 3 2 3 2 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 2 2 2 3 3 3 3 3 3 3 2 3 3 3 3 3 [93] 3 3 3 3 3 3 3 3 2 3 2 2 2 2 3 2 2 2 2 2 2 3 3 2 2 2 2 3 2 3 2 3 2 2 3 3 2 2 2 2 2 3 3 2 2 2 [139] 3 2 2 2 3 2 2 2 3 2 2 3
- Within cluster sum of squares by cluster:
- [1] 13.1290 12.6217 11.3000
- (between_SS / total_SS = 71.6 %)
- Available components: [1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss" [7] "size" "iter" "ifault”

Errors

- table(ic1[,2],ic1[,3])

	1	2	3
setosa	0	40	10
versicolor	14	5	31
virginica	37	1	12

table(ic3[,4],ic3[,5])

	1	2	3
setosa	0	50	0
versicolor	5	0	45
virginica	37	0	13

table(ic2[,3],ic2[,4])

	1	2	3
setosa	50	0	0
versicolor	0	12	38
virginica	0	35	15

table(ic4[,5],ic4[,6])

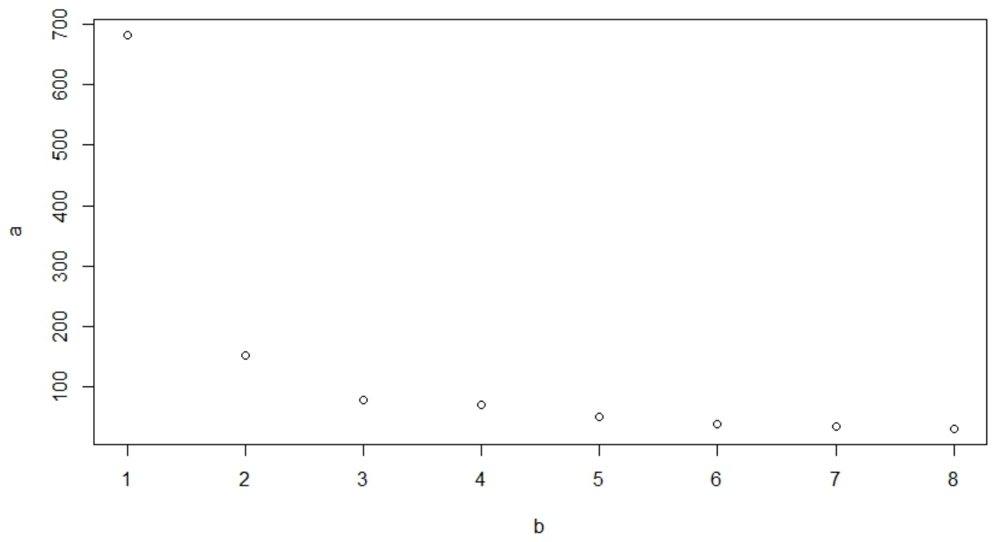
	1	2	3
setosa	0	33	17
versicolor	46	0	4
virginica	50	0	0

CODE for Elbow Rule

- ir<- iris
- irt<-ir[,-5]
- head(ir)
- head(irt)
- #Elbow Rule
- a<- numeric()
- for(i in 1:8){
-
-   irk1<- kmeans(irt,i)
-   a<- append(a,irk1$tot.withinss,i)
- }

- b<-1:8

- plot(b,a)



NAÏVE BAYE’S


Statistics and Probability are two related but separate academic disciplines. Statistical analysis often uses probability distributions, and the two topics are often studied together. Now we will discuss about probability.

Probability

- Probability is the measure of how likely something will occur.
- It is the ratio of desired outcomes to total outcomes.

(# desired) / (# total)

- Probabilities of all outcomes sums to 1.

Example:

- ✓ If I roll a dice, there are six total possibilities. (1,2,3,4,5,6)
- ✓ Each possibility only has one outcome, so each has a PROBABILITY of 1/6.
- ✓ For instance, the probability of getting a numeric 2 is 1/6, since there is only a single 2 on the dice.

Bayes Theorem

- Bayes' theorem (also known as Bayes' rule) is a useful tool for calculating conditional probabilities.

$$\text{Bayes' theorem:}$$

$$P(A \mid B) = \frac{P(A \text{ and } B)}{P(B)}$$

- For independent events P(A|B) = P(A), so by rearranging the formula we can see,

$$P(A \text{ and } B) = P(A) \times P(B)$$

Probability Distribution

- A probability distribution assigns a probability to each measurable subset of the possible outcomes of a random experiment

One Toss	Head	Tail
Probability	0.5	0.5

Two tosses	Head-Head	Tail-Tail	Head-Tail	Tail-Head
Probability	0.25	0.25	0.25	0.25

- Rules:

1. The outcomes listed must be disjoint
2. Each probability must be between 0 and 1
3. The probabilities must sum to 1

Consider a school with a total population of 100 persons. These 100 persons can be seen either as ‘Students’ and ‘Teachers’ or as a population of ‘Males’ and ‘Females’.

With below tabulation of the 100 people, what is the conditional probability that a certain member of the school is a ‘Teacher’ given that he is a ‘Man’?

	Female	Male	Total
Teacher	8	12	20
Student	32	48	80
Total	40	60	100

$$P(Teacher \mid Male) = \frac{P(Teacher \bigcap Male)}{P(Male)} = 12/60 = 0.2$$

This can be represented as the intersection of Teacher (A) and Male (B) divided by Male (B). Likewise, the conditional probability of B given A can be computed. The Bayes Rule that we use for Naive Bayes, can be derived from these two notations.

$$P(A \mid B) = \frac{P(A \bigcap B)}{P(B)} \qquad (1)$$

$$P(B \mid A) = \frac{P(A \bigcap B)}{P(A)} \qquad (2)$$

Bayes Rule is a way to go from P (X | Y) to find P (Y | X)

$$P(X \mid Y) = \frac{P(X \cap Y)}{P(Y)} \quad \boxed{1}$$

Known

$$P(Y \mid X) = \frac{P(X \cap Y)}{P(X)} \quad \boxed{2}$$

UnKnown

P (Evidence | Outcome)
(Known from training data)

⇒

P (Outcome | Evidence)
(To be predicted for test data)

Bayes Rule

$$\boxed{P(Y \mid X) = \frac{P(X \mid Y) * P(Y)}{P(X)}}$$

When there are multiple X variables, we simplify it by *<u>assuming the X's are independent</u>*, so the **Bayes** rule

$$P(Y=k \mid X) = \frac{P(X \mid Y=k) * P(Y=k)}{P(X)}$$

where, k is a class of Y

becomes, Naive **Bayes**

$$P(Y=k \mid X1..Xn) = \frac{P(X1 \mid Y=k) * P(X2 \mid Y=k) \ldots * P(Xn \mid Y=k) * P(Y=k)}{P(X1) * P(X2) \ldots * P(Xn)}$$

$$P(Y=k \mid X1..Xn) = \frac{P(X1 \mid Y=k) * P(X2 \mid Y=k) \ldots * P(Xn \mid Y=k) * P(Y=k)}{P(X1) * P(X2) \ldots * P(Xn)}$$

can be understood as ..

$$\frac{\text{Probability of Outcome} \mid \text{Evidence}}{\text{(Posterior Probability)}} \;\; \text{— i.e.} \;\; \text{Probability of Outcome} \mid \text{Evidence (Posterior Probability)} = \frac{\text{Probability of Likelihood of evidence} * \text{Prior}}{\text{Probability of Evidence}}$$

Probability of Evidence is same for all classes of Y

Problem

- Say you have 1000 fruits which could be either ‘banana’, ‘orange’ or ‘other’. These are the 3 possible classes of the Y variable. We have data for the following X variables, all of which are binary (1 or 0).
  - Long
  - Sweet
  - Yellow

Type	Long	Not Long	Sweet	Not Sweet	Yellow	Not Yellow	**Total**
Banana	400	100	350	150	450	50	500
Orange	0	300	150	150	300	0	300
Other	100	100	150	50	50	150	200
Total	500	500	650	350	800	200	1000

So the objective of the classifier is to predict if a given fruit is a ‘Banana’ or ‘Orange’ or ‘Other’ when only the 3 features (long, sweet and yellow) are known.

- Let’s say you are given a fruit that is: Long, Sweet and Yellow, can you predict what fruit it is?

- This is the same of predicting the Y when only the X variables in testing data are known.

- Let’s solve it by hand using Naive Bayes. The idea is to compute the 3 probabilities, that is the probability of the fruit being a banana, orange or other. Whichever fruit type gets the highest probability wins.

Step 1: Compute the 'Prior' probabilities for each of the class of fruits.

That is, the proportion of each fruit class out of all the fruits from the population.

You can provide the 'Priors' from prior information about the population. Otherwise, it can be computed from the training data. For this case, let's compute from the training data. Out of 1000 records in training data, you have 500 Bananas, 300 Oranges and 200 Others.

So the respective priors are 0.5, 0.3 and 0.2.
P(Y=Banana) = 500 / 1000 = 0.50
P(Y=Orange) = 300 / 1000 = 0.30

<u>Step 2: Compute the probability of evidence that goes in the denominator.</u>

This is nothing but the product of P of Xs for all X.
This is an optional step because the denominator is the same for all the classes and so will not affect the probabilities.

P(x1=Long) = 500 / 1000 = 0.50
P(x2=Sweet) = 650 / 1000 = 0.65
P(x3=Yellow) = 800 / 1000 = 0.80

Step 3: Compute the probability of likelihood of evidences that goes in the numerator.

It is the product of conditional probabilities of the 3 features.

If you refer back to the formula, it says P(X1 |Y=k).
Here X1 is ‘Long’ and k is ‘Banana’.
That means the probability the fruit is ‘Long’ given that it is a Banana.

In the above table, you have 500 Bananas. Out of that 400 is long.
So, P(Long | Banana) = 400/500 = 0.8.
Here, I have done it for Banana alone.

Probability of Likelihood for Banana

P(x1=Long | Y=Banana) = 400 / 500 = 0.80
P(x2=Sweet | Y=Banana) = 350 / 500 = 0.70
P(x3=Yellow | Y=Banana) = 450 / 500 = 0.90.

So, the overall probability of Likelihood of evidence for Banana = 0.8 * 0.7 * 0.9 = 0.504

Step 4: If a fruit is *'Long'*, *'Sweet' and 'Yellow'*, what fruit is it?

$$P(\text{Banana} \mid \textit{Long, Sweet and Yellow}) = \frac{P(\text{Long} \mid \text{Banana}) * P(\text{Sweet} \mid \text{Banana}) * P(\text{Yellow} \mid \text{Banana}) \times P(\text{banana})}{P(\text{Long}) * P(\text{Sweet}) * P(\text{Yellow})}$$

$$= \frac{0.8 * 0.7 * 0.9 * 0.5}{P(\text{Evidence})} = 0.252/P(\text{Evidence})$$

$$P(\text{Orange} \mid \textit{Long, Sweet and Yellow}) = 0\text{, because } P(\text{Long} \mid \text{Orange}) = 0$$

$$P(\text{Other Fruit} \mid \textit{Long, Sweet and Yellow}) = 0.01875 \,/\, P(\text{Evidence})$$

Answer: Banana - Since it has highest probability amongst the 3 classes

Naïve Bayes Classifier Steps

First we will create a frequency table using each attribute of the dataset,

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

Frequency Table		Play	
		Yes	No
Outlook	Sunny	2	3
	Overcast	4	0
	Rainy	3	2

Frequency Table		Play	
		Yes	No
Humidity	High	3	4
	Normal	6	1

Frequency Table		Play	
		Yes	No
Wind	Strong	6	2
	Weak	3	3

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

Outlook

	Yes	No	P(yes)	P(no)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Wind

	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

Play		P(Yes)/P(No)
Yes	9	9/14
No	5	5/14
Total	14	100%

For example, probability of playing golf given that the temperature is cool,

i.e P(temp. = cool | play golf = Yes) = 3/9.

Also, we need to find class probabilities (P(y)) which has been calculated in the table 5.

For example, P(play golf = Yes) = 9/14.

today = (Sunny, Hot, Normal, False)

$$P(Yes|today) = \frac{P(SunnyOutlook|Yes)P(HotTemperature|Yes)P(NormalHumidity|Yes)P(NoWind|Yes)P(Yes)}{P(today)}$$

$$P(No|today) = \frac{P(SunnyOutlook|No)P(HotTemperature|No)P(NormalHumidity|No)P(NoWind|No)P(No)}{P(today)}$$

$$P(Yes|today) \propto \frac{2}{9} \cdot \frac{2}{9} \cdot \frac{6}{9} \cdot \frac{6}{9} \cdot \frac{9}{14} \approx 0.0141$$

$$P(No|today) \propto \frac{3}{5} \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{2}{5} \cdot \frac{5}{14} \approx 0.0068$$

These numbers can be converted into a probability by making the sum equal to 1 (normalization):

$$P(Yes|today) = \frac{0.0141}{0.0141+0.0068} = 0.67$$

and

$$P(No|today) = \frac{0.0068}{0.0141+0.0068} = 0.33$$

Since

$$P(Yes|today) > P(No|today)$$

So, prediction that golf would be played is 'Yes'.