Data Science

Unsupervised Learning

Agenda

01 Types of Unsupervised Learning

02 Clustering

03 K-Means Clustering

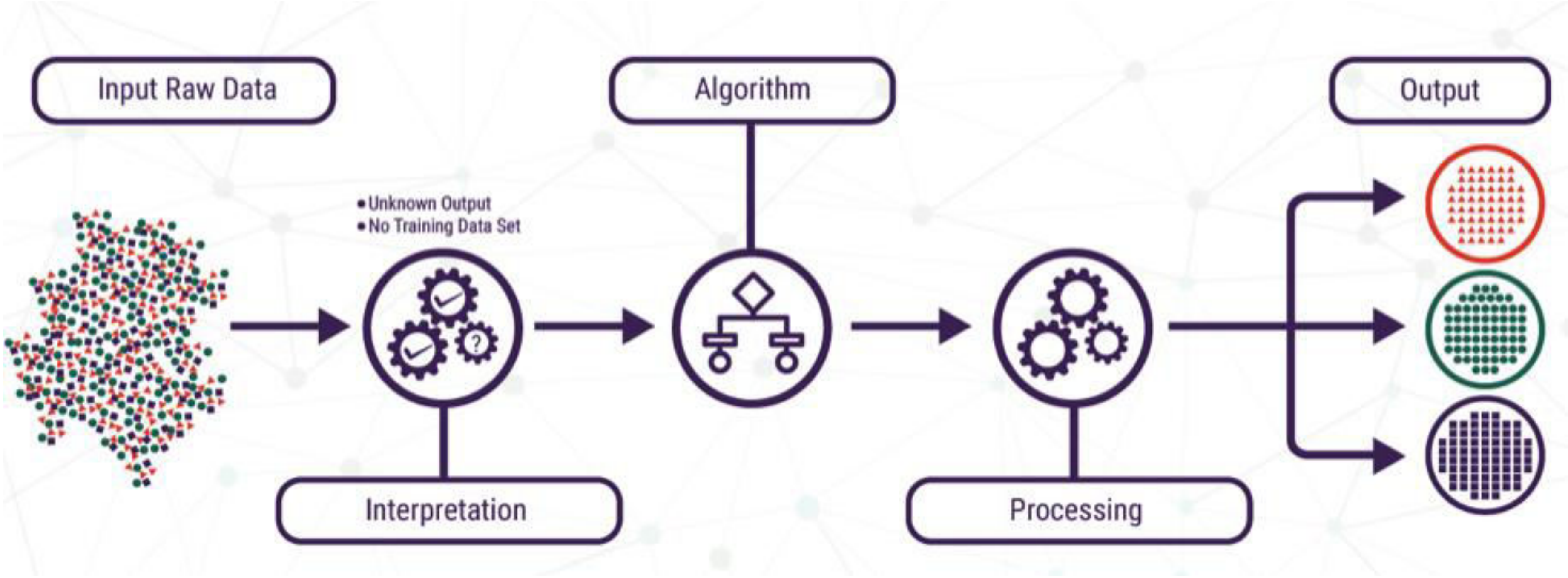
04 Hierarchical Clustering

Unsupervised Learning

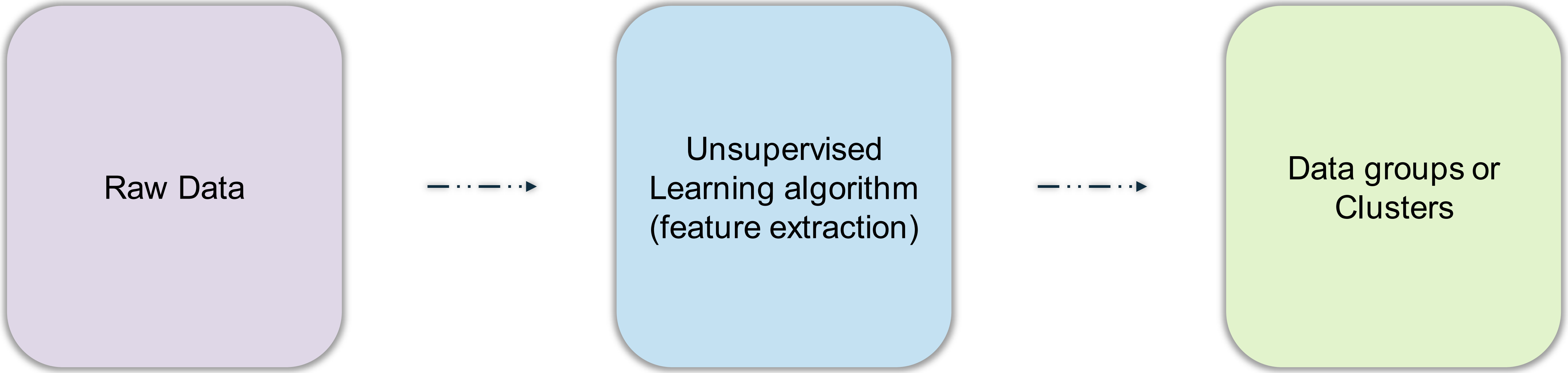
Unsupervised Learning

In **unsupervised learning**, an algorithm segregates the data in a data set in which the data is unlabeled based on some hidden features in the data

This function can be useful for discovering the hidden structure of data and for tasks like anomaly detection

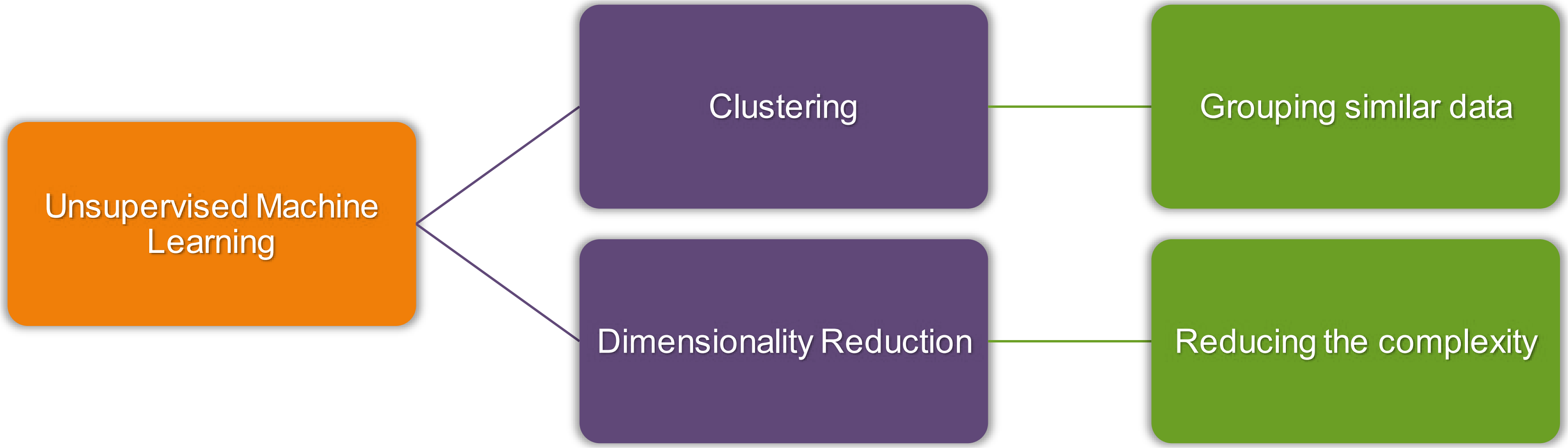


Unsupervised Learning



Types of Unsupervised Learning

Types of Unsupervised Learning



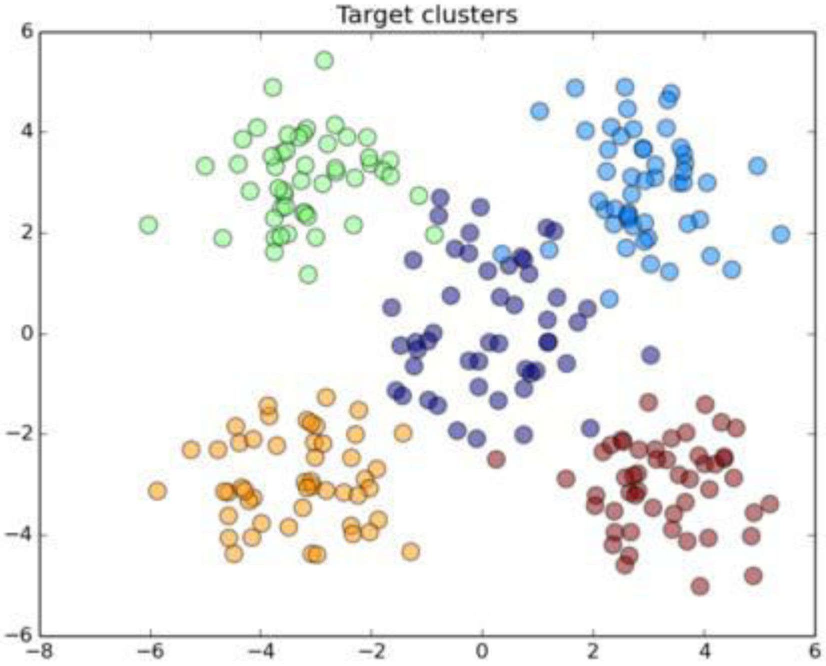
Clustering

Clustering

Process of dividing data sets into groups of similar data points

Dividing a dataset into data points where,

- Points in the same group are as similar as possible
- Points in the different group are dissimilar as possible



Types of Clustering

Types of Clustering

Clustering can be divided into two sub-groups:

Hard Clustering

In hard clustering, each data point either belongs to a cluster completely or not.

Soft Clustering

In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned.

Types of Clustering Algorithms

Types of Clustering Algorithms

Connectivity-based clustering

- Data points that are closer in the data space are more related (similar) than to data points farther away.
- The clusters are formed by connecting data points according to their distance.
- Examples - **Hierarchical Clustering** Algorithm

Centroid models

- These are iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters.
- **k-means** is a centroid based clustering

Types of Clustering Algorithms

Distribution-based clustering

- Clustering is based on the notion of how probable is it for a data point to belong to a certain distribution, such as the Gaussian distribution.
- Data points in a cluster belong to the same distribution. These models have a strong theoretical foundation, however they often suffer from overfitting.

Gaussian mixture models

- Using the expectation-maximization algorithm is a famous distribution based clustering method.

Types of Clustering Algorithms

Density-based methods

- Search the data space for areas of varied density of data points. Clusters are defined as areas of higher density within the data space compared to other regions.
- DBSCAN and OPTICS are some prominent density based clustering.

Intra-cluster cohesion (compactness)

- Cohesion measures how near the data points in a cluster are to the cluster centroid.

Inter-cluster separation (isolation)

- Separation means that different cluster centroids should be far away from each other.

Which Algorithm to Use?

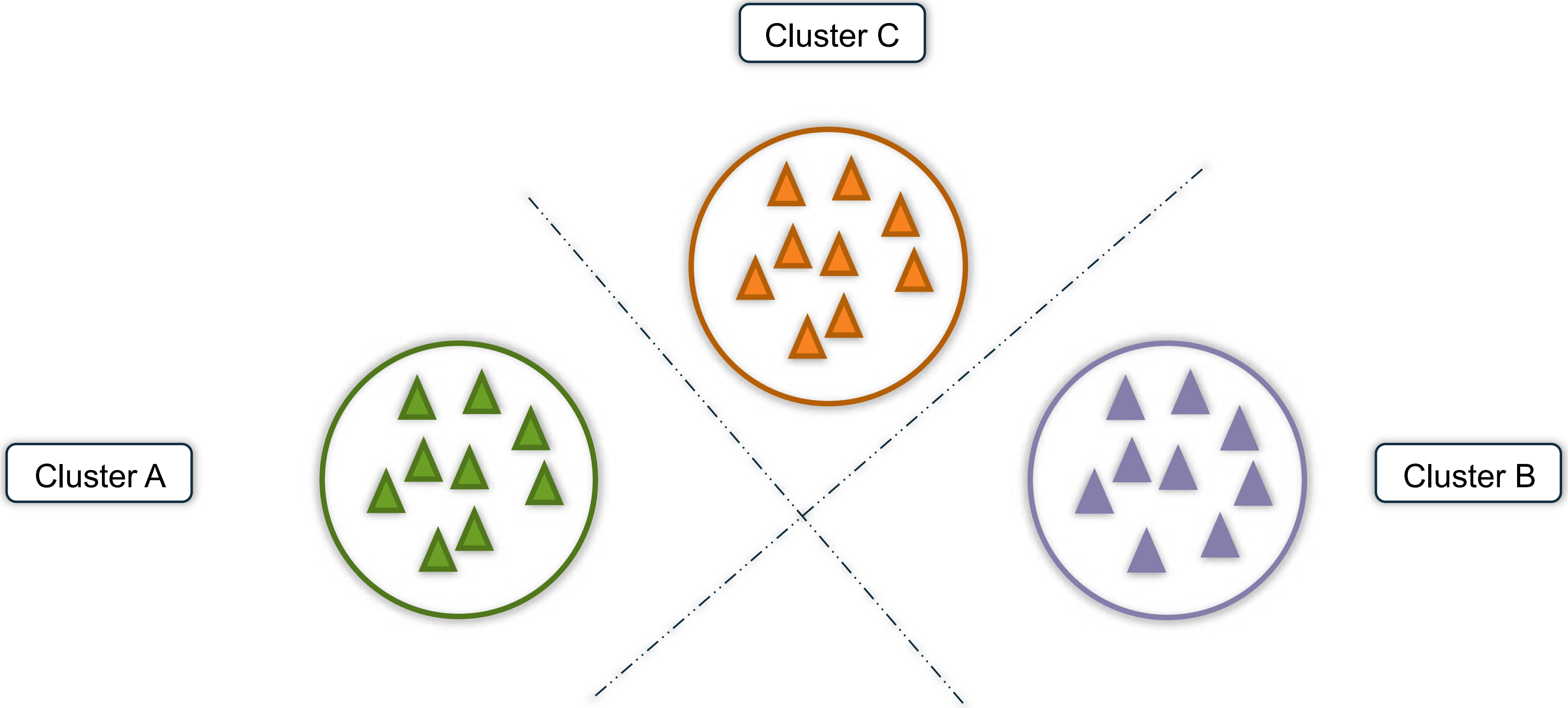
There is no ONE algorithm to rule them all!!

- Clustering is an subjective task and there can be more than one correct clustering algorithm.
- Every algorithm follows a different set of rules for defining the 'similarity' among data points.

K-means Clustering

K-means Clustering

K-means clustering is the most commonly used unsupervised machine learning algorithm for dividing a given dataset into k clusters. Here, ‘K’ represents the number of clusters provided by the user



K-means Clustering Algorithm

Step 1

Specify the desired number of clusters K":

Let us choose k=2 for these 5 data points.

Step 2

Randomly assign each data point to a cluster:

Let's assign three points in cluster 1 shown using red color and two points in cluster 2 shown using grey color.

Step 3

Compute cluster centroids:

The centroid of data points in the red cluster is shown using red cross and those in grey cluster using grey cross.

Step 4

Re-assign each point to the closest cluster centroid:

Only the data point at the bottom is assigned to the red cluster even though its closer to the centroid of grey cluster. Thus, we assign that data point into grey cluster

Step 5

Re-compute cluster centroids:

Re-computing the centroids for both the clusters.

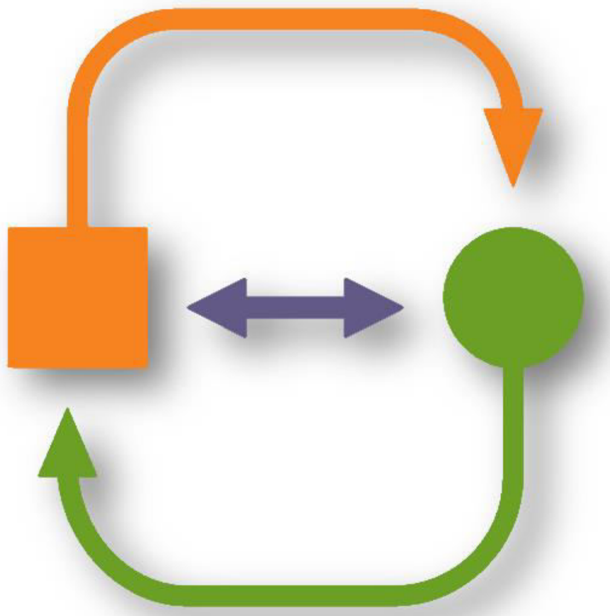
Step 6

Repeat steps 4 and 5 until no improvements are possible:

When there will be no further switching of data points between two clusters for two successive repeats. It will mark the termination of the algorithm.

K-means Clustering Algorithm

K-Means runs on distance calculations, which uses "**Euclidean Distance**"



$$\text{Euclidean Distance} = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$$

K-means Clustering

The basic restriction for K-Means algorithm is that your data should be continuous in nature

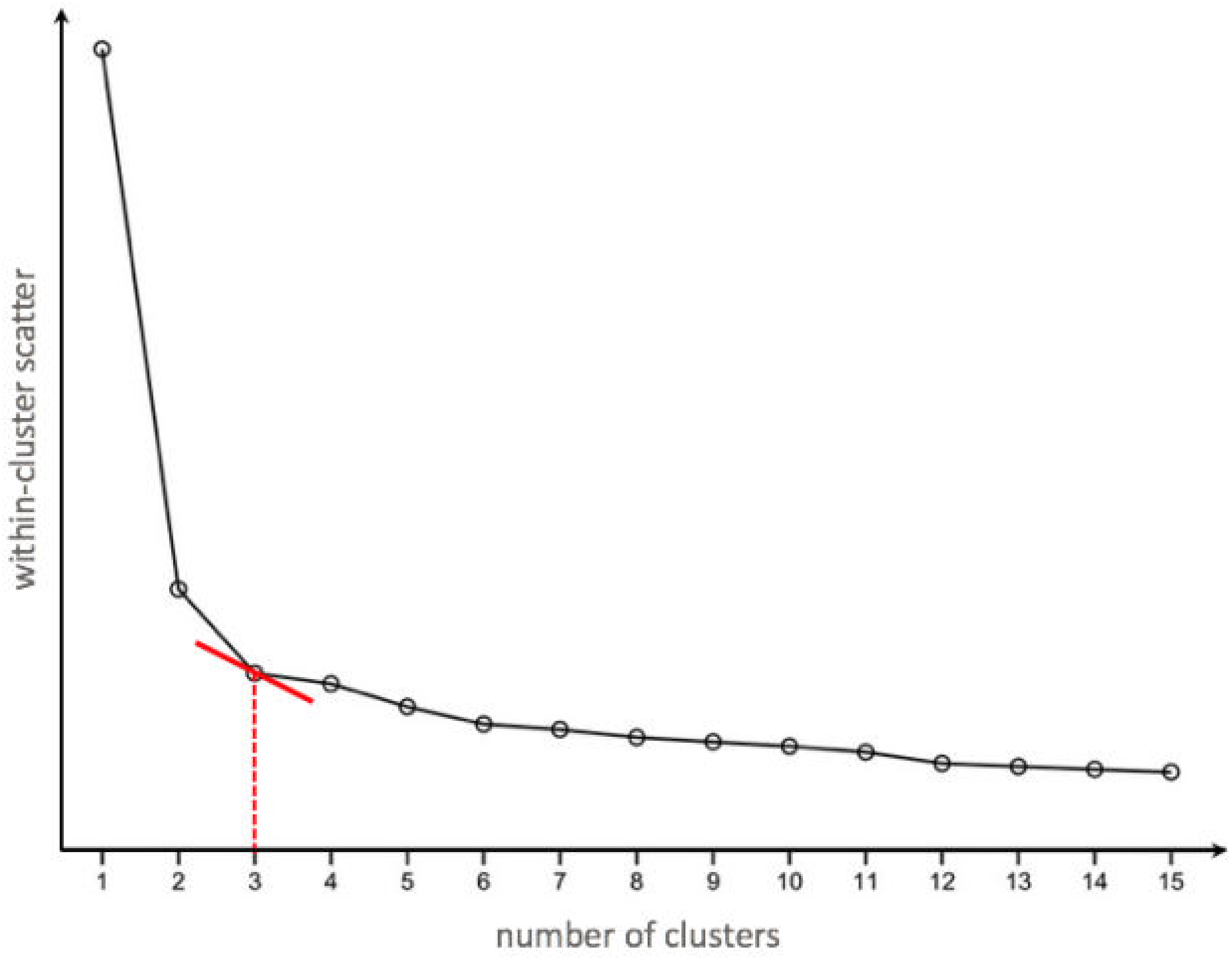
It won’t work if data is categorical in nature!

Finding the Optimal Number of Clusters

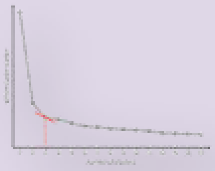
Optimal Number of Clusters

Run k-means multiple times to see how model quality changes as the number of clusters change

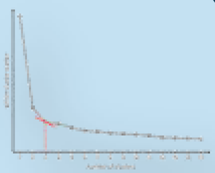
Plots displaying this information help to determine the number of clusters and are often referred to as *scree plots*



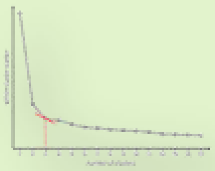
Optimal Number of Clusters



The ideal plot will have an elbow where the quality measure improves more slowly as the number of clusters increases



This indicates that the quality of the model is no longer improving substantially as the model complexity (i.e. number of clusters) increases



In other words, the elbow indicates the number of clusters inherent in the data

Optimal Number of Clusters

Compute k-means clustering for different values of k. For instance, by varying k from 1 to 15 clusters.

For each k, calculate the total within-cluster sum of square (wss).

Plot the curve of wss according to the number of clusters k.

The location of a bend (**knee**) in the plot is generally considered as an indicator of the appropriate number of clusters.

K-means output

kmeans() function in R

Kmeans() output generates -

cluster	centers	Withinss	tot.withinss	Size
a vector of integers (from 1:k) indicating the cluster to which each point is allocated.	a matrix of cluster centers.	vector of within-cluster sum of squares, one component per cluster.	total within-cluster sum of squares. That is, sum(withinss).	the number of points in each cluster.

K-means in R

Problem Statement

Building k-means algorithm on top of the customer_churn dataset

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
7590-VHVEG	Female	0	Yes	No	1	No	No phone service
5575-GNVDE	Male	0	No	No	34	Yes	No
3668-QPYBK	Male	0	No	No	2	Yes	No
7795-CFOCW	Male	0	No	No	45	No	No phone service
9237-HQITU	Female	0	No	No	2	Yes	No
9305-CDSKC	Female	0	No	No	8	Yes	Yes
1452-KIOVK	Male	0	No	Yes	22	Yes	Yes
6713-OKOMC	Female	0	No	No	10	No	No phone service
7892-POOKP	Female	0	Yes	No	28	Yes	Yes
6388-TABGU	Male	0	No	Yes	62	Yes	No

Tasks to be performed

1. Build the k-means algorithm on the ‘MonthlyCharges’ column & set the number of clusters to be 3

2. Build the k-means algorithm on the ‘tenure’ column & set the number of clusters to be 3

3. Build the k-means algorithm on the ‘TotalCharges’ column & set the number of clusters to be 3

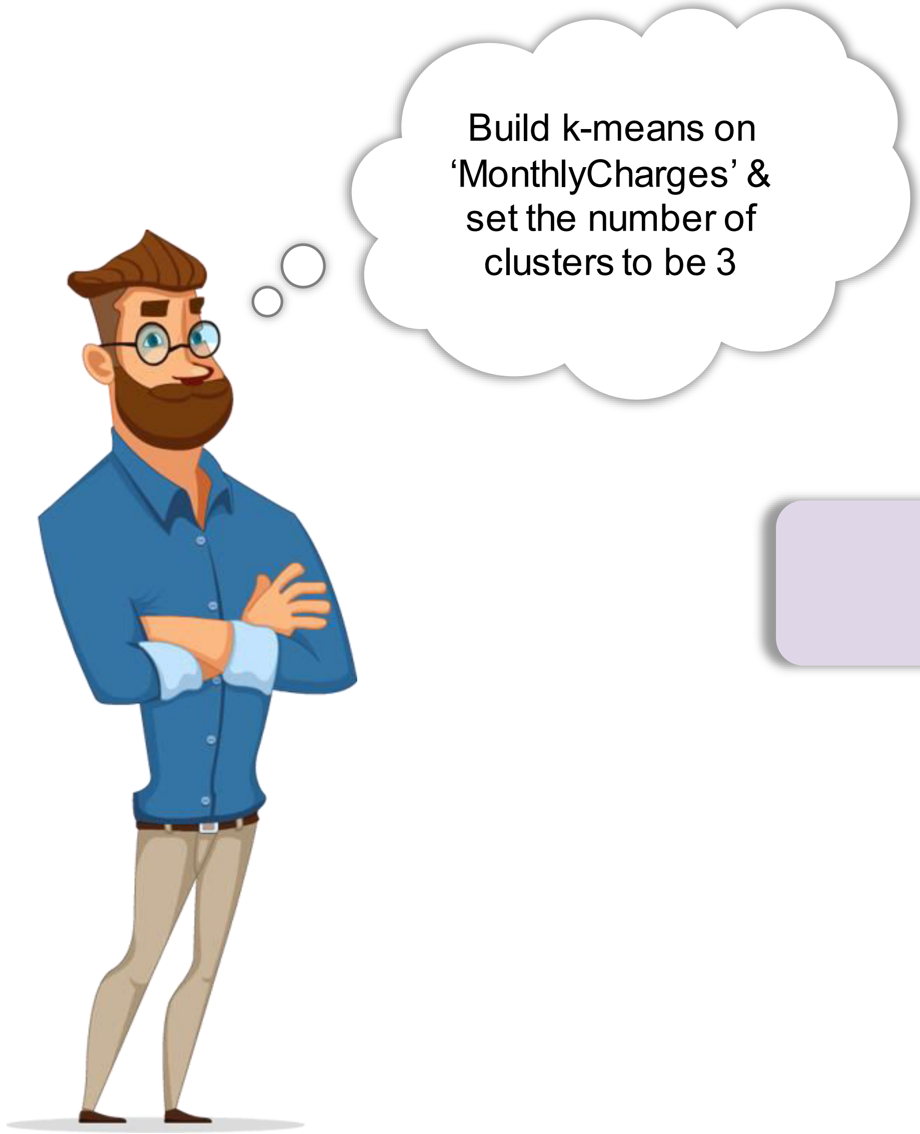
K-means in R

Read the customer_churn dataset and select only ‘MonthlyCharges’, ‘tenure’ & ‘TotalCharges’

```
customer_churn<-read.csv("C:/Users/INTELLIPAAT/Desktop/customer_churn.csv")
```

```
customer_churn %>% select("tenure","MonthlyCharges","TotalCharges")-> customer_features
```

K-means in R



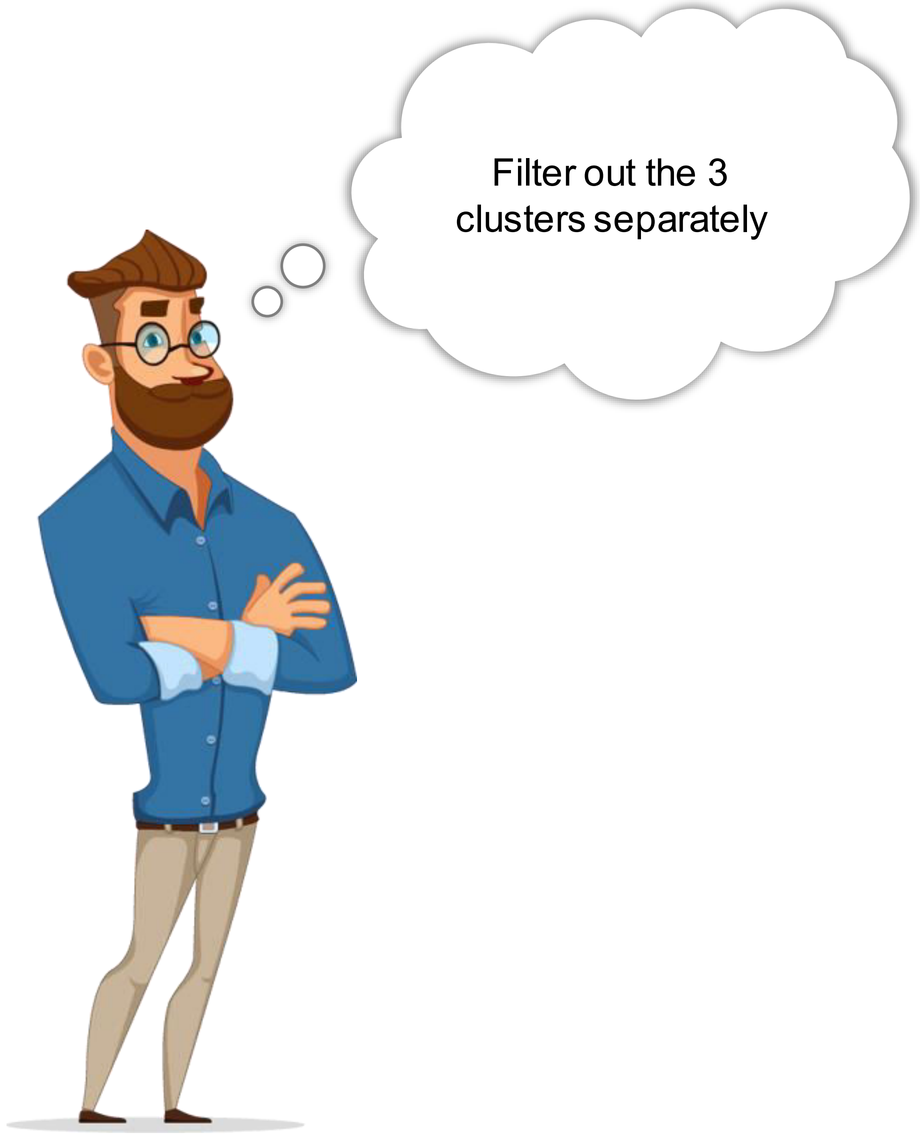

```
kmeans(customer_features$MonthlyCharges,3) -> k_month
```

K-means in R


```
cbind(Month=customer_features$MonthlyCharges, Clusters=k_month$cluster) -> month_group
```

```
head(month_group)
```

K-means in R




```
as.data.frame(month_group) ->
month_group
```

```
month_group %>% filter(Clusters==1)->
month_group_1
```

```
month_group %>% filter(Clusters==2)->
month_group_2
```

```
month_group %>% filter(Clusters==3)->
month_group_3
```

K-means in R

Build k-means on
'tenure' & set the
number of clusters
to be 3

```
kmeans(customer_features$tenures,3) -> tenure_group
```

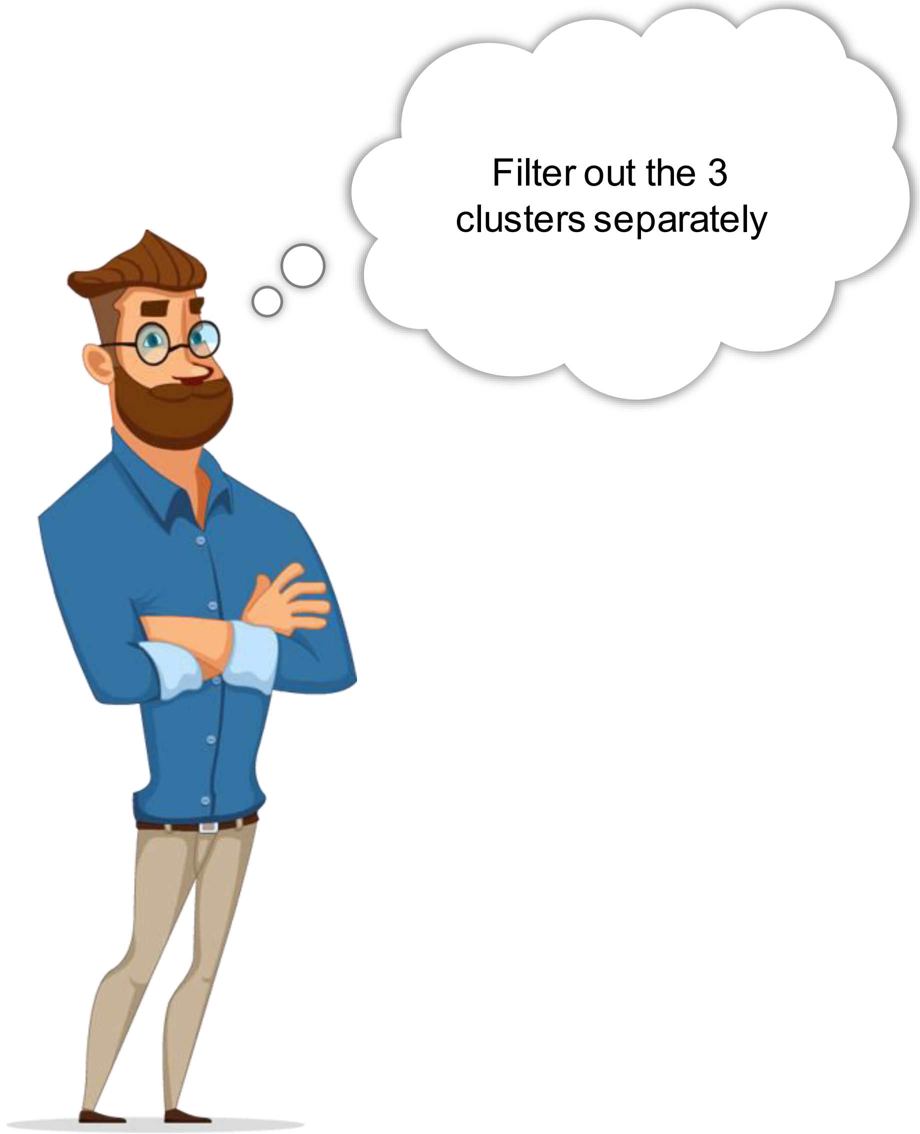
K-means in R

Bind the
'MonthlyCharges'
column & the
clustering vector
together

```
cbind(Month=customer_features$tenure, Clusters=tenure_group$cluster) ->
tenure_group_data
```

```
head(tenure_group_data)
```

K-means in R




```
as.data.frame(tenure_group_data) -> tenure_group_data
```

```
tenure_group_data %>% filter(Clusters==1)-> tenure_group_data1
```

```
tenure_group_data %>% filter(Clusters==2)-> tenure_group_data2
```

```
tenure_group_data %>% filter(Clusters==3)-> tenure_group_data3
```

K-means in R

Build k-means on ‘TotalCharges’ & set the number of clusters to be 3

```
kmeans(customer_features$TotalCharges,3) -> k_total
```

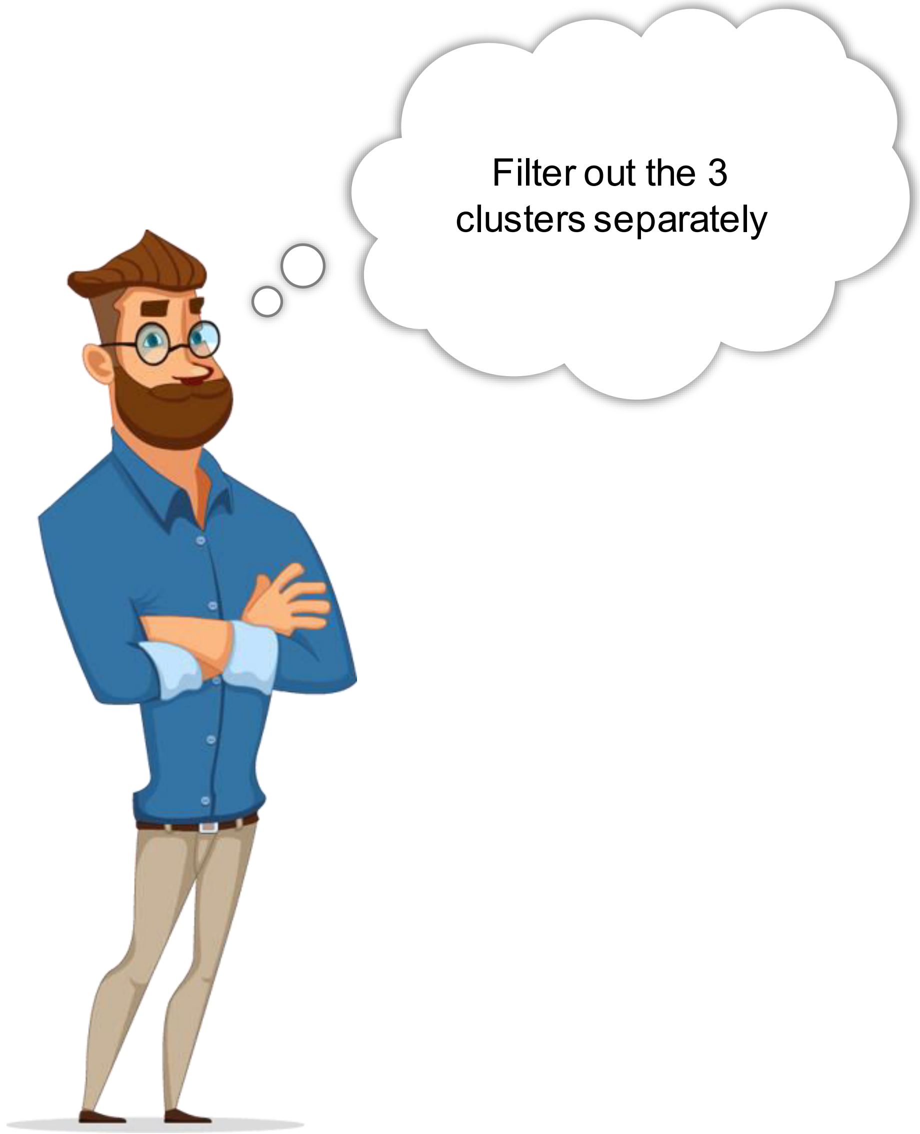
K-means in R

Bind the
‘MonthlyCharges’
column & the
clustering vector
together

```
cbind(Month=customer_features$TotalCharges, Clusters=k_total$cluster) -> total_group
```

```
head(total_group)
```

K-means in R




```
as.data.frame(total_group) -> total_group
```

```
total_group %>% filter(Clusters==1)-> total_group1
```

```
total_group %>% filter(Clusters==2)-> total_group2
```

```
total_group %>% filter(Clusters==3)-> total_group3
```

K-means on ‘iris’ Dataset

Problem Statement

Building k-means algorithm on top of the 'iris' dataset

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

K-means on 'iris'

Build k-means on
the numerical
columns of 'iris'
dataset

```
km <- kmeans(iris[,1:4], 3)
kmeans.ani(iris[,1:4],3)
```

K-means on ‘iris’

Glance at the model attributes

km$cluster
km$centers
km$totss
km$withinss
km$tot.withinss
km$betweenss
km$size

K-means on 'iris'

Build 'scree-plot' to get optimal number of clusters

```
mydata <- iris[,c(1:4)]
kmax=20
twss=rep(0,kmax)
ratio=rep(0,kmax)
```

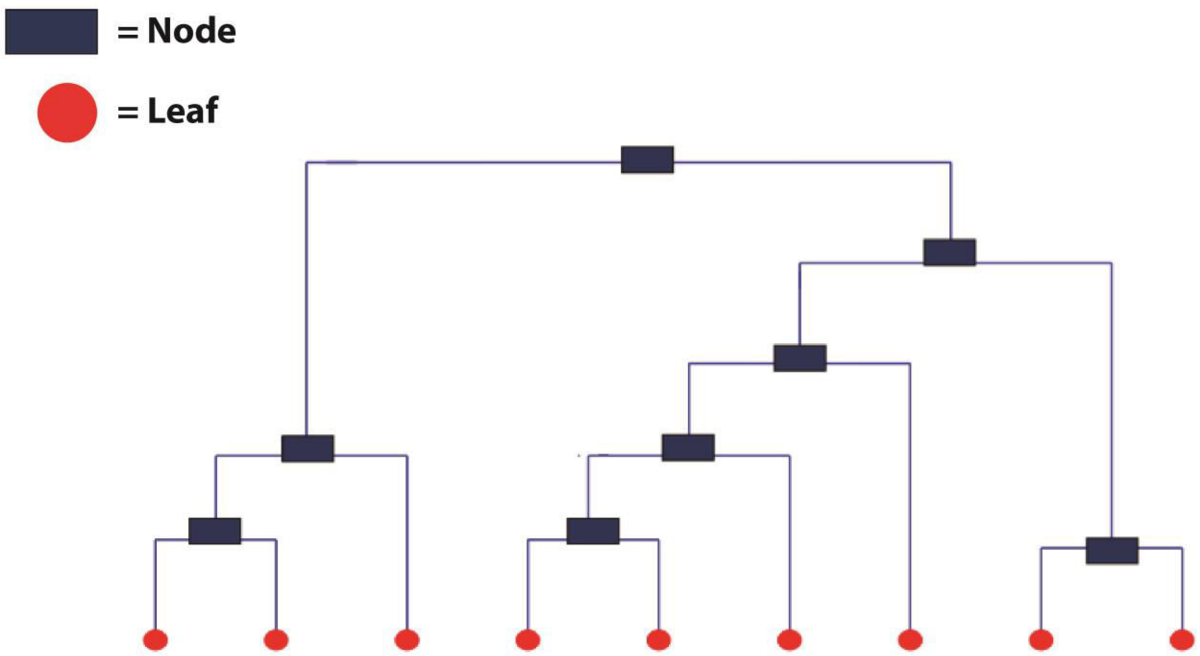
```
for (i in 1:kmax) {
set.seed(1234)
km=kmeans(mydata,centers = i,nstart = 10,iter.max = 1000000)
twss[i]<-km$tot.withinss
ratio[i]<-km$betweenss/km$totss
}
```

```
plot(1:kmax,twss,type='b',xlab="Number of clusters",ylab="TWSS",col="blue")
```

Hierarchical Clustering

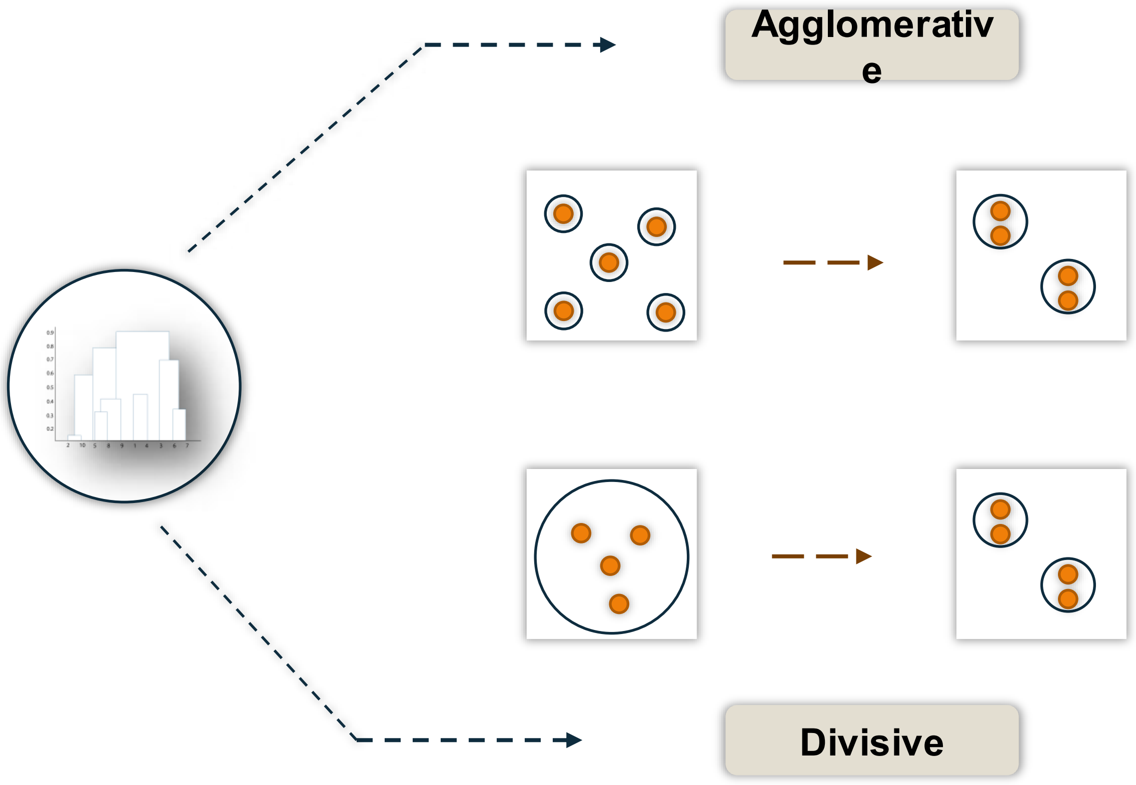
Hierarchical Clustering

Hierarchical Clustering is a method for creating a ***hierarchy of clusters***



Hierarchical Clustering

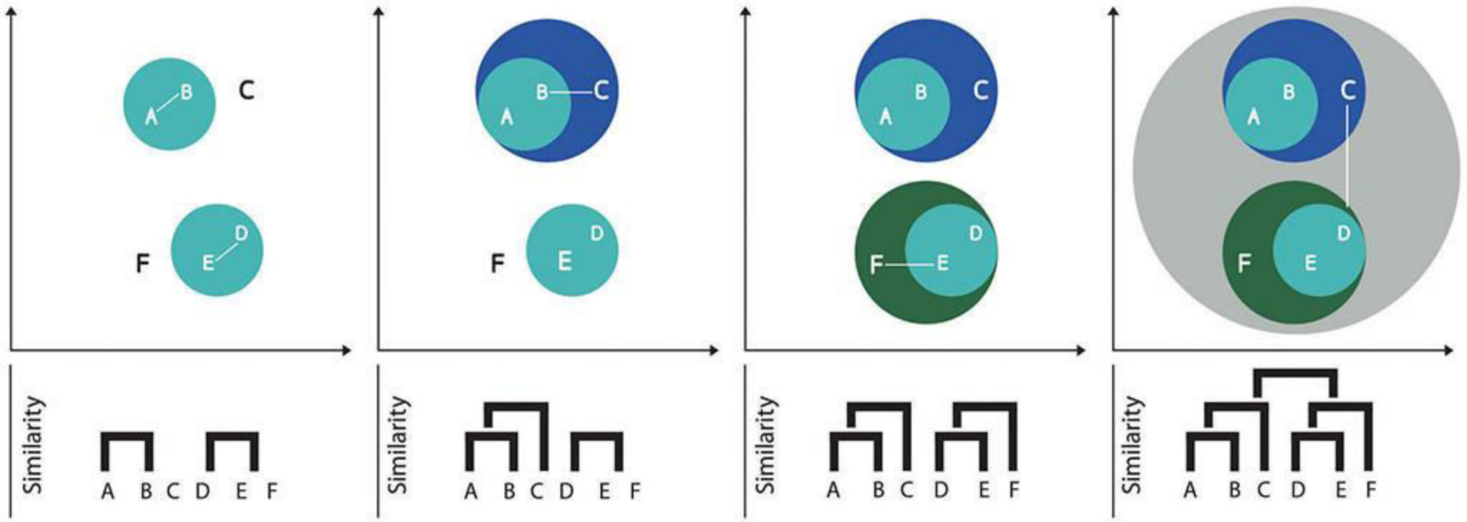
Hierarchical Clustering can be either bottom-up or top-down



Agglomerative Hierarchical Clustering

Agglomerative Hierarchical Clustering

This algorithm starts with all the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left



- Combine A & B based on similarity
- Combine D & E based on similarity

1

- Combination of A & B is combined with C

2

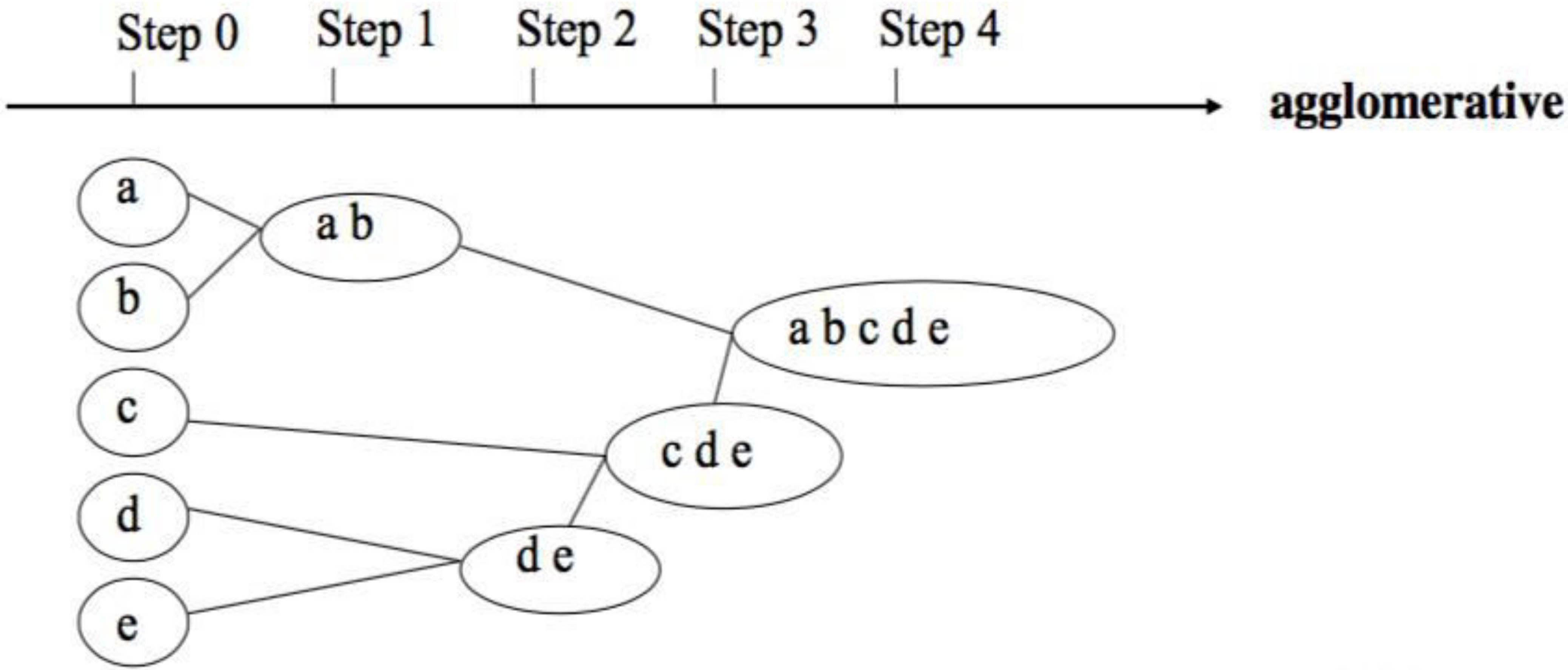
- Combination of D & E is combined with F

3

- Final tree contains all clusters combined into a single cluster

4

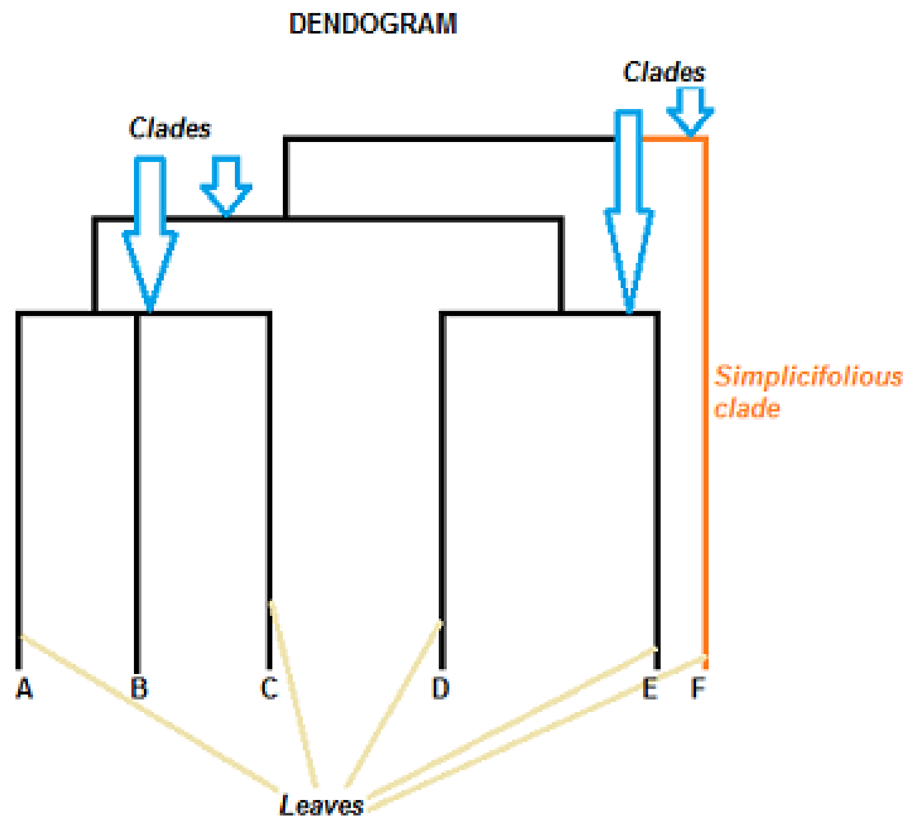
Agglomerative Hierarchical Clustering



Dendogram

Dendogram

A *dendrogram* is a tree-like structure which shows the hierarchical relationship between objects

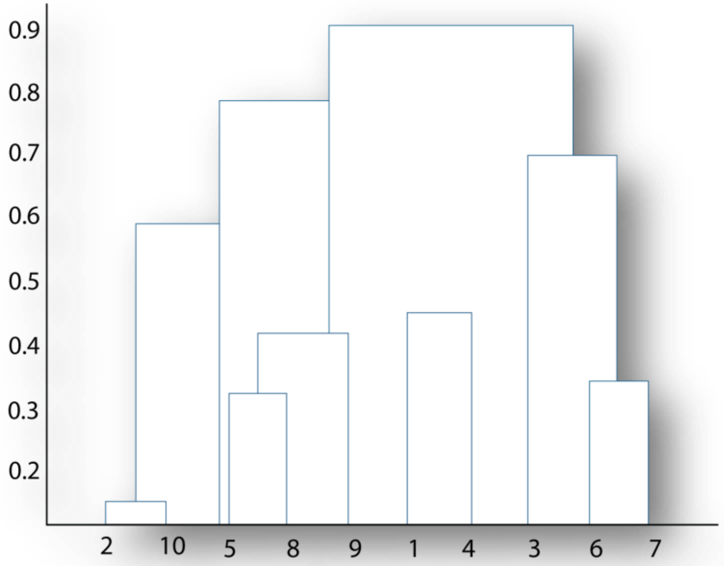


Dendogram

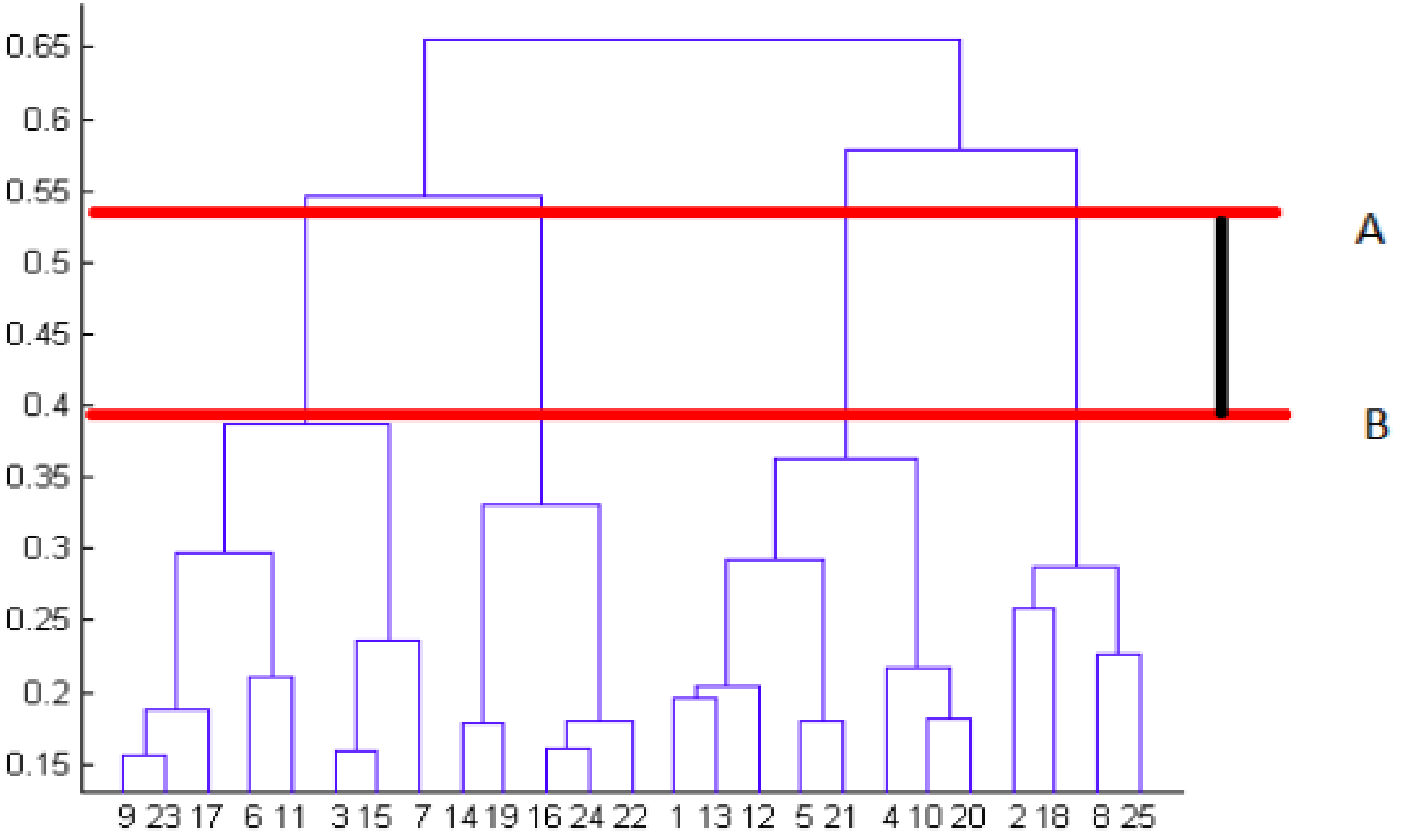
1. The height in the dendrogram at which two clusters are merged represents the distance between two clusters in the data space

2. The decision of the no. of clusters that can best depict different groups can be chosen by observing the dendrogram

3. The best choice of the no. of clusters is the no. of vertical lines in the dendrogram cut by a horizontal line that can transverse the maximum distance vertically without intersecting a cluster



Dendogram



0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
9 23 17 6 11 3 15 7 14 19 16 24 22 1 13 12 5 21 4 10 20 2 18 8 25
A
B

Hierarchical Clustering

The decision of merging two clusters is taken on the basis of closeness of these clusters

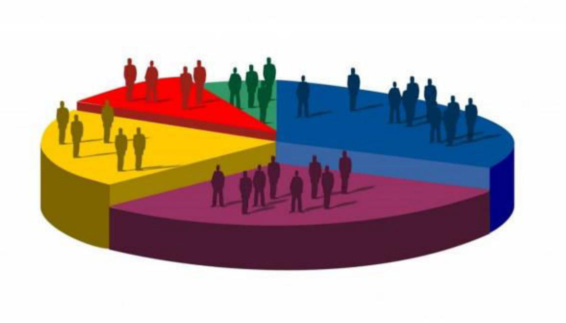
There are multiple metrics for deciding the closeness of two clusters:

- Euclidean distance
- Squared Euclidean distance
- Manhattan distance
- Maximum distance
- Mahalanobis distance

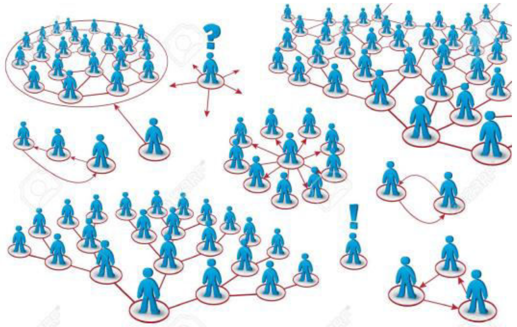
Clustering Examples

Clustering Examples

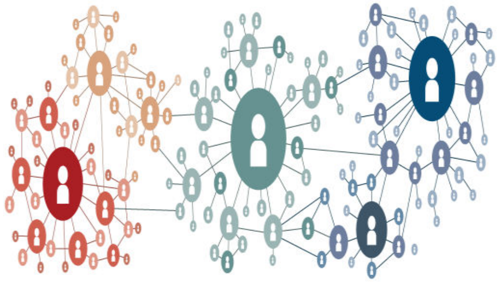
Recommendation Engines



Market Segmentation



Social Network Analysis



Search Result Grouping

Medical Imaging

Anomaly Detection

Quiz

Quiz

Q 1. How can Clustering (Unsupervised Learning) be used to improve the accuracy of Linear Regression model (Supervised Learning):

1. Creating different models for different cluster groups.
2. Creating an input feature for cluster ids as an ordinal variable.
3. Creating an input feature for cluster centroids as a continuous variable.
4. Creating an input feature for cluster size as a continuous variable.

A. 1 only

B. 1 and 2

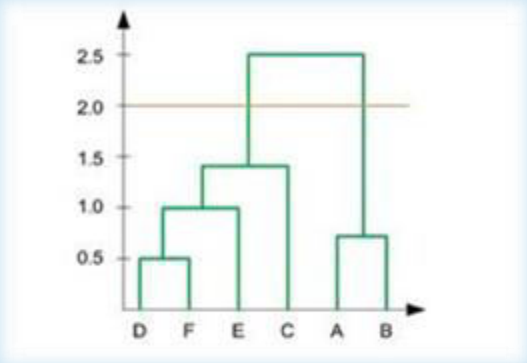
C. 3 only

D. 2 and 4

E. All of the above

Quiz

Q 2. In the figure below, if you draw a horizontal line on y-axis for y=2. What will be the number of clusters formed?



A. 1

B. 2

C. 3

D. 4

Quiz

Q 3. In which of the following cases will K-Means clustering fail to give good results?

1. **Data points with outliers**
2. **Data points with different densities**
3. **Data points with round shapes**
4. **Data points with non-convex shapes**

A. 1 and 2

B. 2 and 3

C. 2 and 4

D. 1, 2 and 4

Thank You

India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

sales@intellipaat.com

24X7 Chat with our Course Advisor