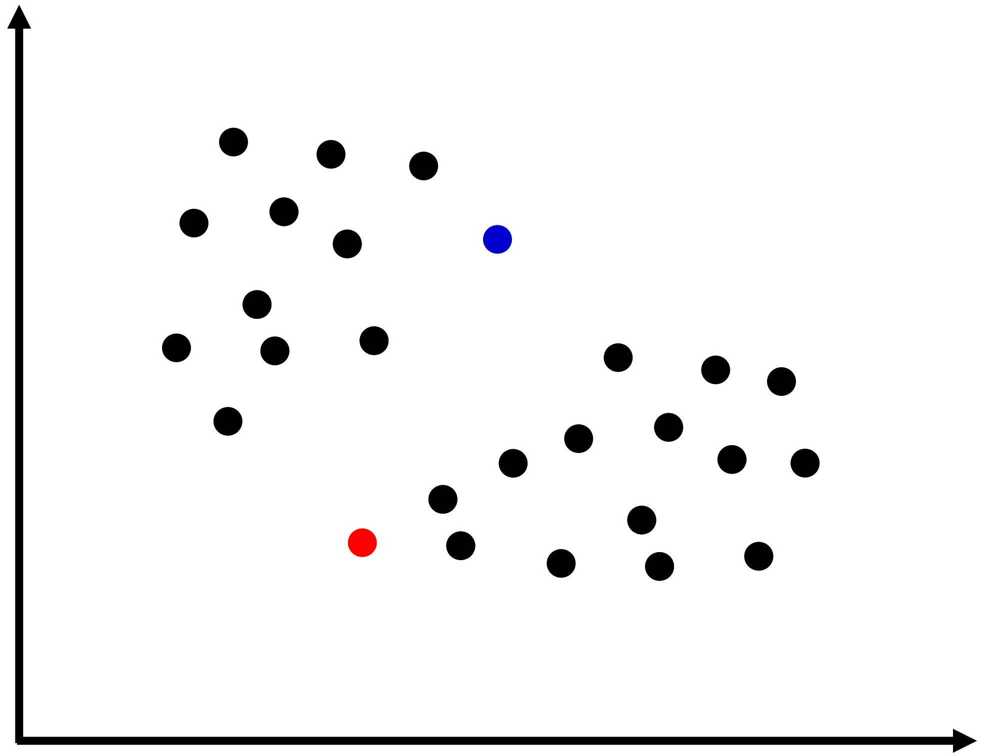
Introduction to Data Science
(Lecture 18)

Dr. Mohammad Pourhomayoun
Assistant Professor
Computer Science Department
California State University, Los Angeles

K-Means Clustering Algorithm
(Continue)

Random Initialization

- To make sure that the centroid points are around, we usually randomly pick "K" of our data samples as the **initial** centroid points.



Pseudo Code for K-Means

Notations:

i = the index of the data sample (1,2,…, m) .

$c^{(i)}$ = index of the cluster (1,2,…,K) to which example $\boldsymbol{x}^{(i)}$ is currently assigned.

$\boldsymbol{\mu}_k$ = cluster centroid k ($\boldsymbol{\mu}_k \in \mathbb{R}^n$).

E.g.: if $\boldsymbol{x}^{(1)}$ is in cluster 5, and $\boldsymbol{x}^{(2)}$ is in cluster 7, then

$c^{(1)} = 5$ and $c^{(2)} = 7$

Pseudo Code for K-Means

Randomly initialize K cluster centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$.

Repeat {

Data samples re-assign

for i = 1 to m :

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $\boldsymbol{x}^{(i)}$: k for

$$\min_k \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_k \right\|^2$$

Centroid re-assign

for k = 1 to K:

$\boldsymbol{\mu}_K$:= average (mean) of points assigned to cluster k

}

Random Initialization

- Notice that the final clustering results may depend on the choice of initial points!

Random Initialization

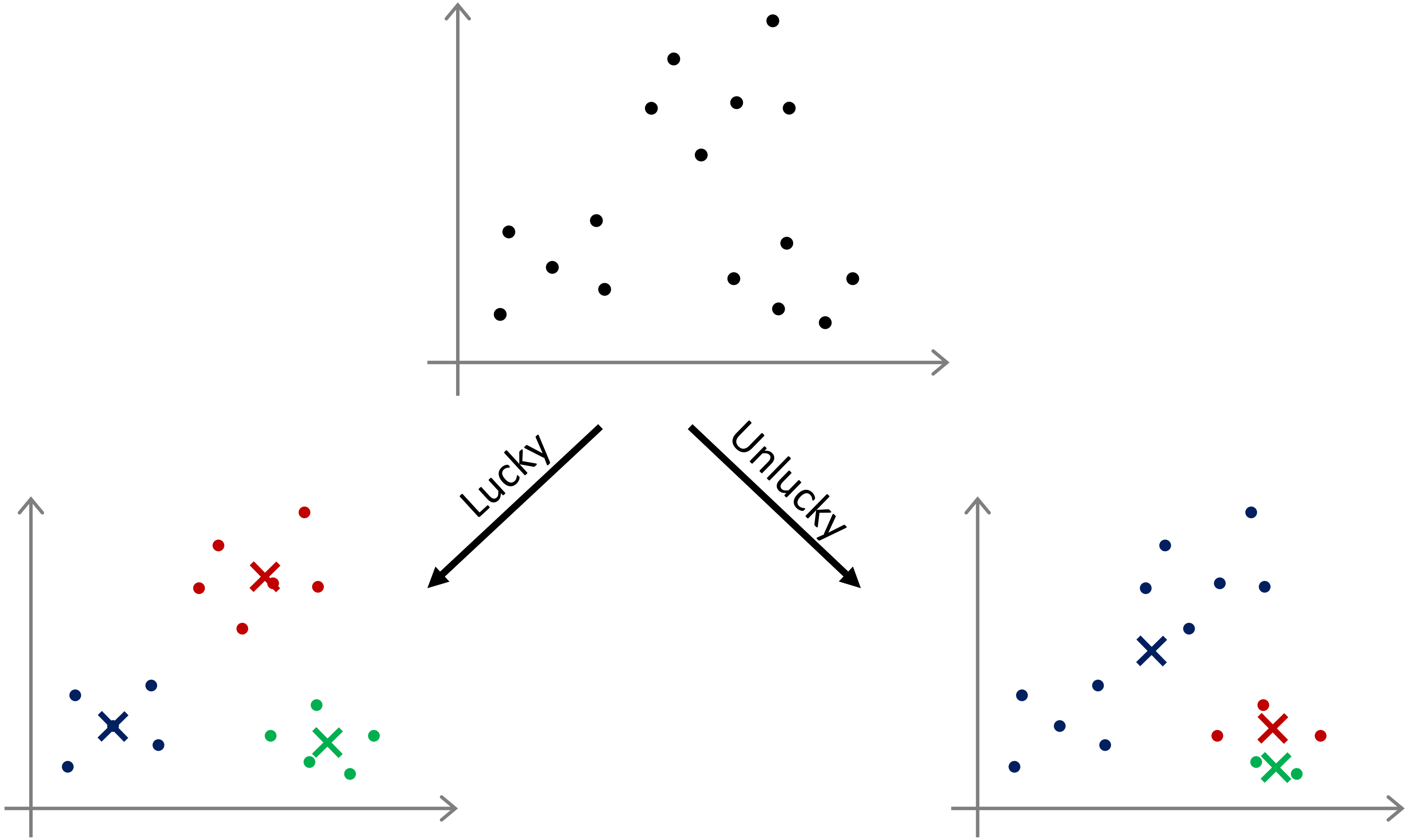


Figure Ref: Andrew Ng

Random Initialization

- Notice that the final clustering results may depend on the choice of initial points!

- Thus, The best approach is to **repeat random initialization multiple times** (rather than trusting on one single initialization), perform clustering several times, and finally select the the best clustering results.

Random Initialization

Notation:

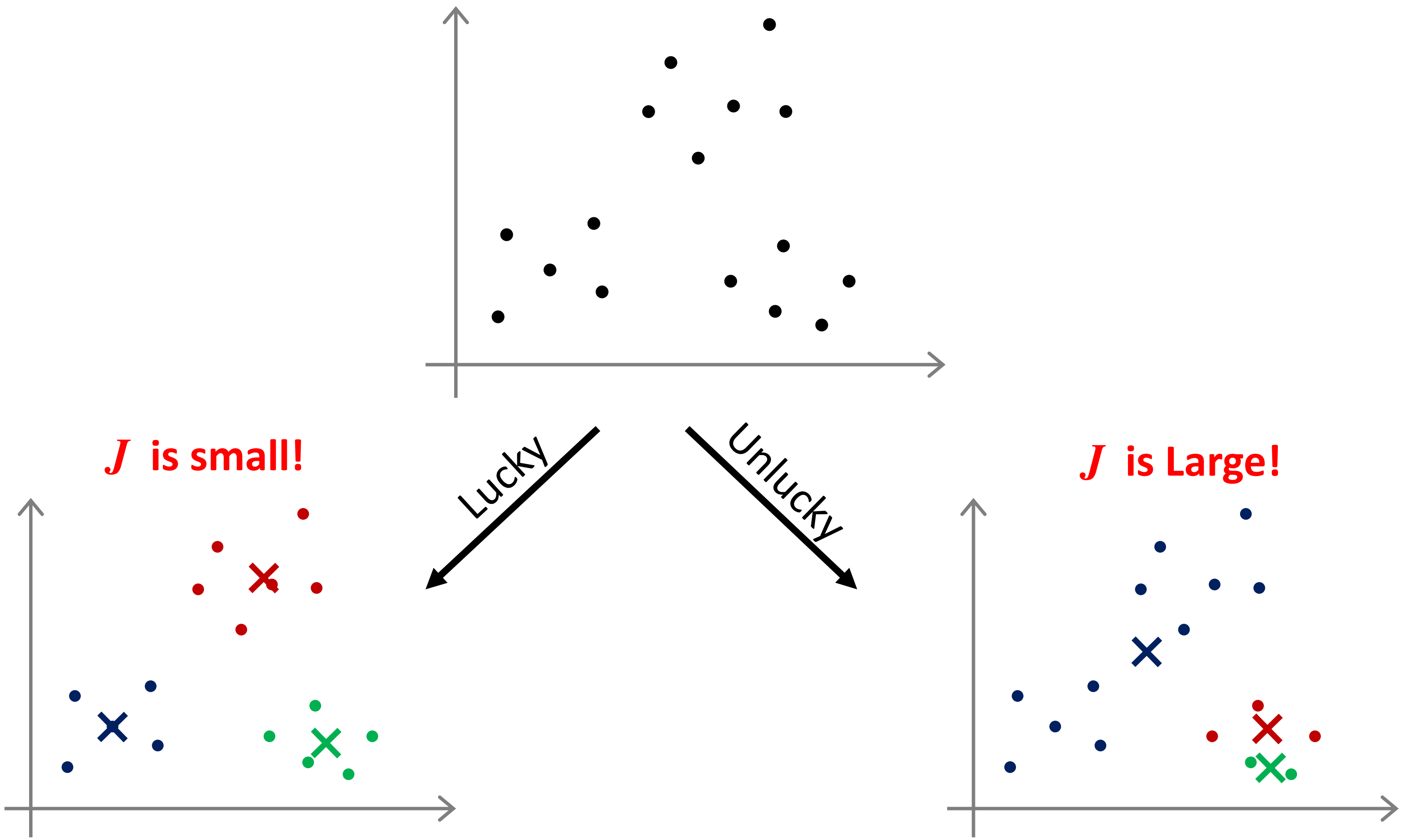
$c^{(i)}$ = index of cluster (1,2,…,K) to which example $\boldsymbol{x}^{(i)}$ is currently assigned.

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $\boldsymbol{x}^{(i)}$ has been assigned.

J = "**clustering cost function**" defined as the **average distance of each sample to its cluster centroid**:

$$J = \frac{1}{m}\sum_{i=1}^{m}\left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{c^{(i)}} \right\|^2$$

Random Initialization



Random Initialization

Multiple Random Initialization:

For i = 1 to 50 {

Randomly initialize K-means.

Run K-means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$.

Compute cost function as following:

$$J = \frac{1}{m}\sum_{i=1}^{m}\left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{c^{(i)}} \right\|^2$$

}

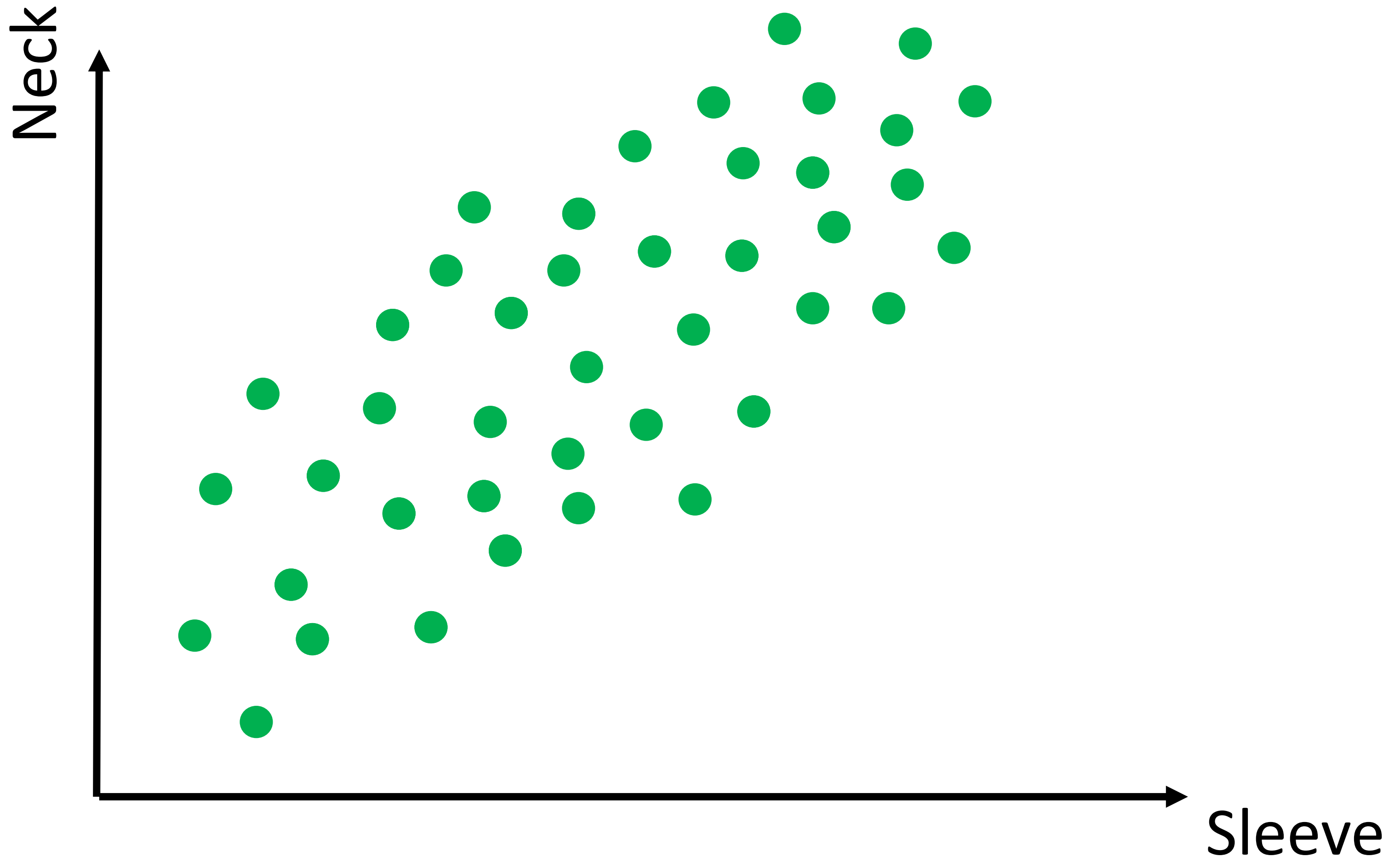
In this approach, we try kmeans clustering for 50 times. Then, we pick the one that gave the lowest cost J.

K-Means for Non-Separated Groups

- Sometimes, K-Means can be very helpful to cluster non-separated data.
- It is particularly very useful for “**product segmentation**” in marketing.
- **Example:** defining clothing size based on sleeve length, neck, chest, …
  – XS, S, M, L, XL, XXL, …

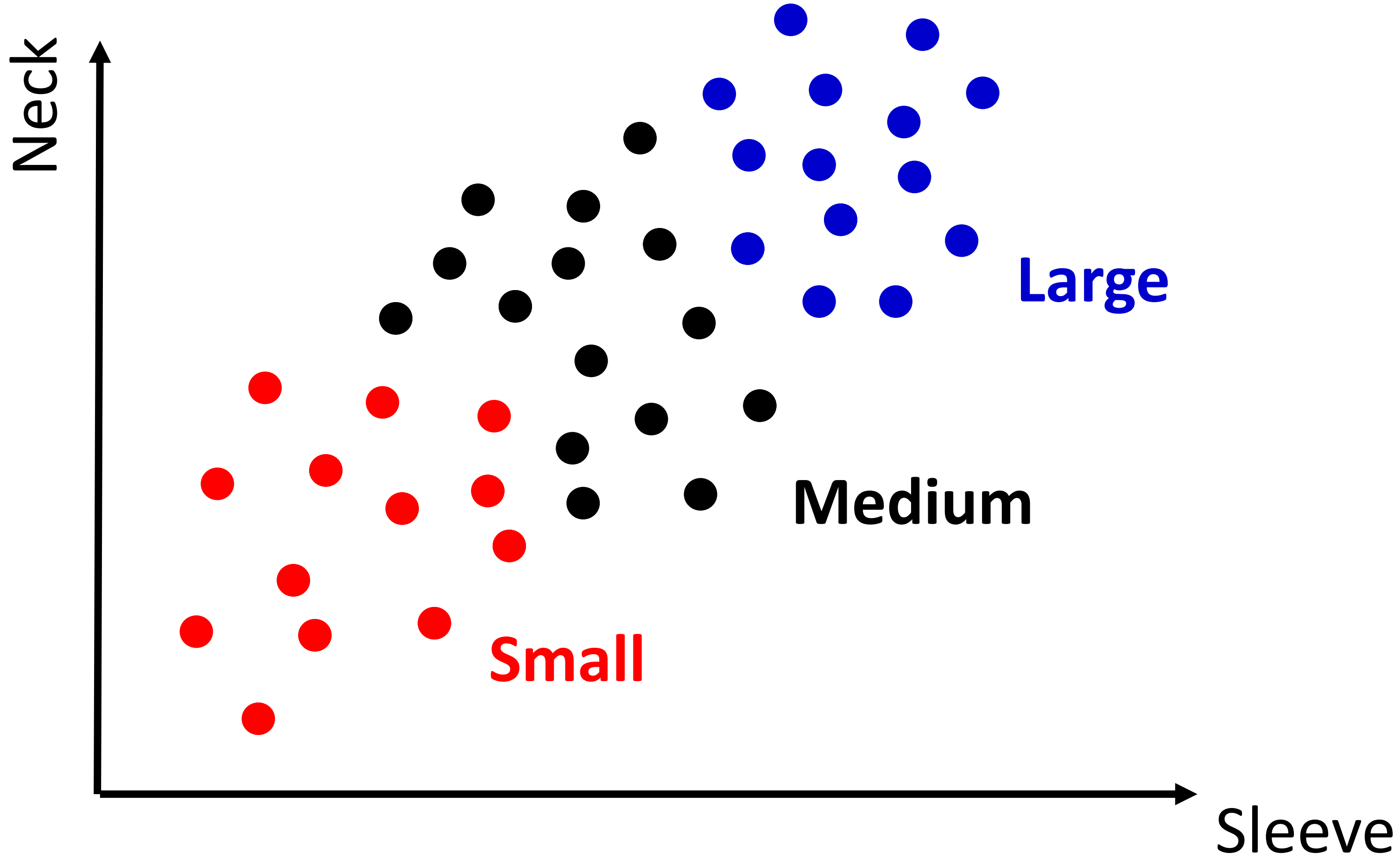
K-Means for Non-Separated Groups

- **Shirt Size:**

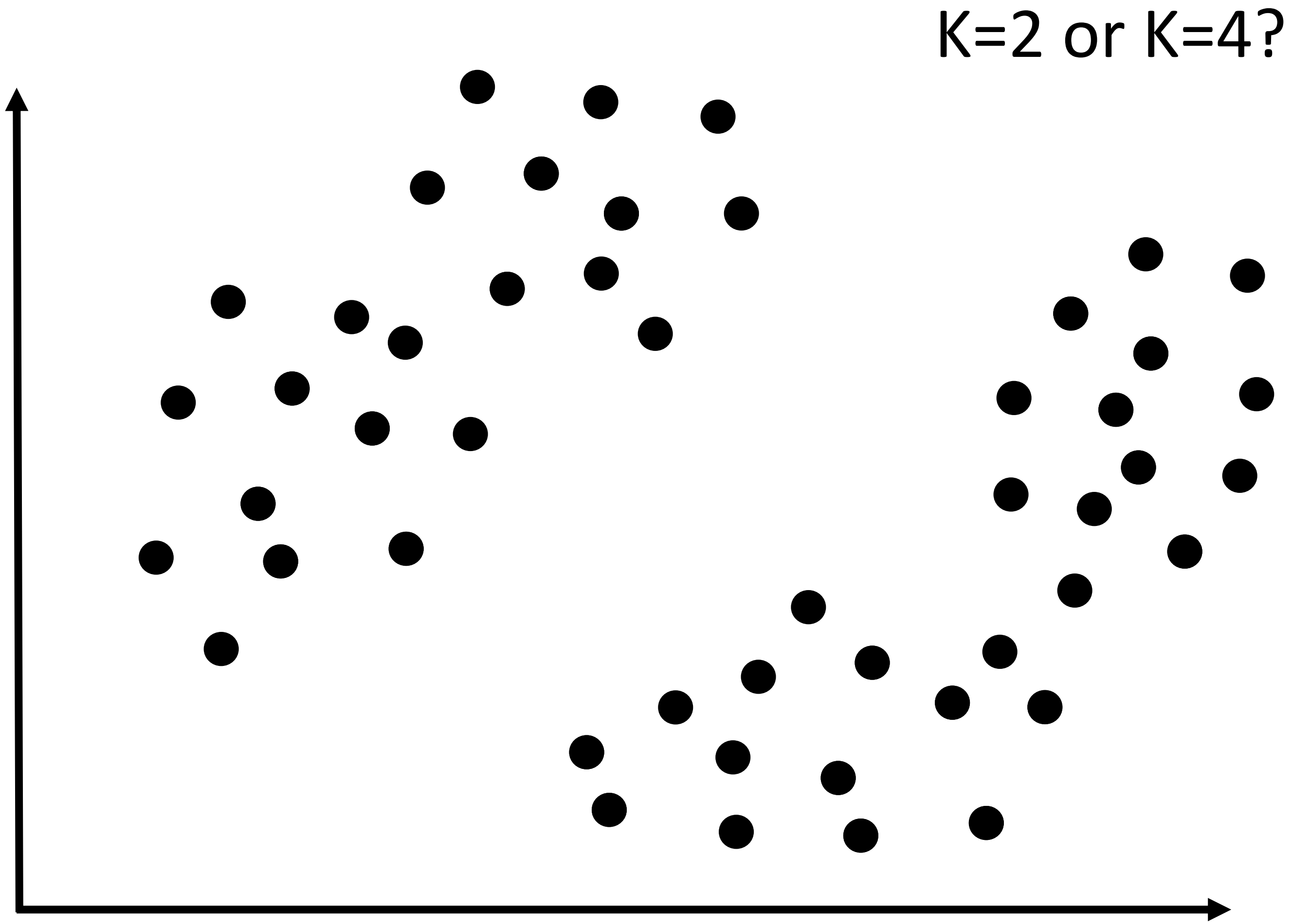


K-Means for Non-Separated Groups

- **Shirt Size:**

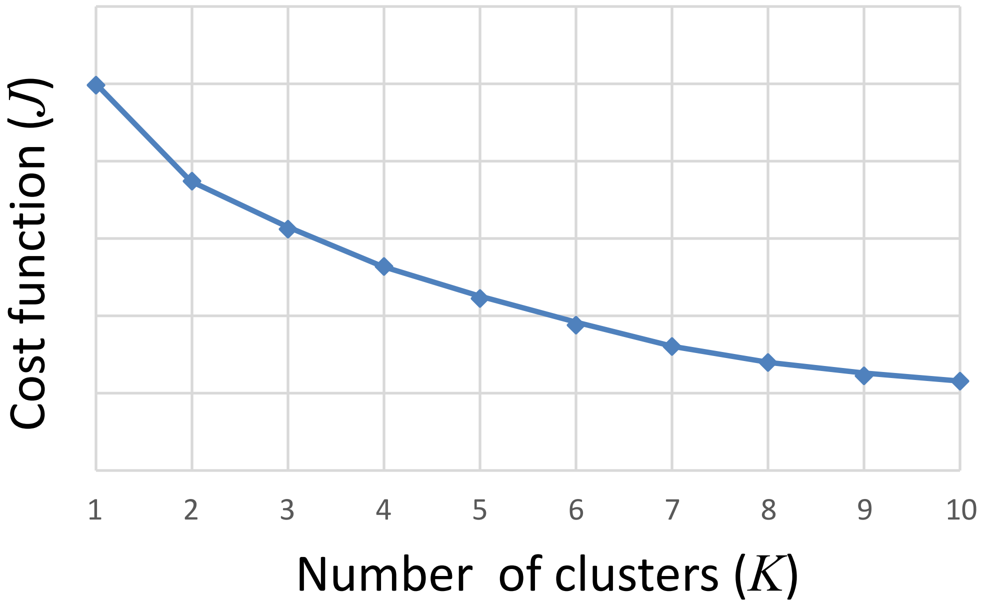


How to choose the Number of Clusters?



How to choose the Number of Clusters?

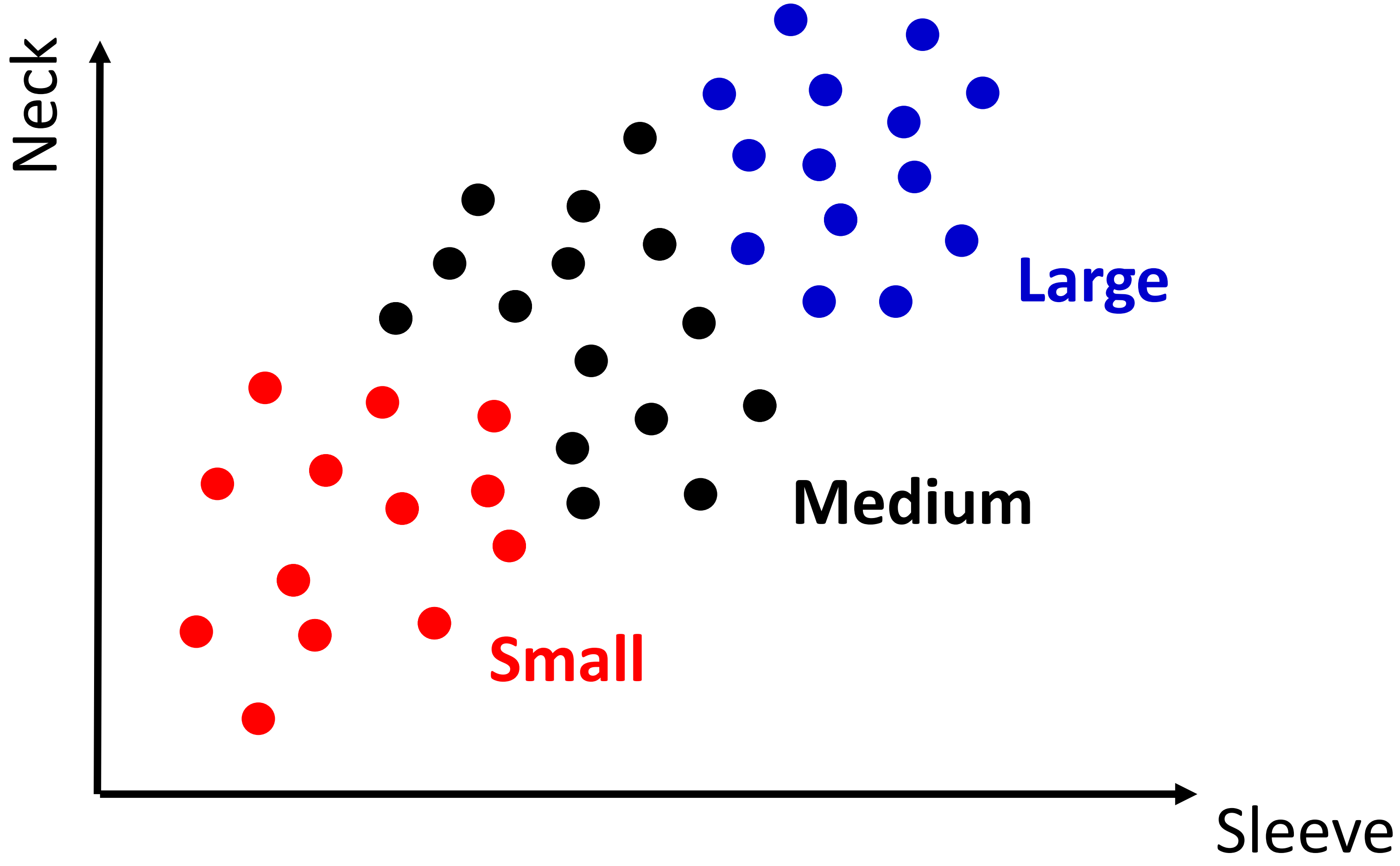
- Trade-off between "Cost Function (J)" and "Number of Clusters(K)":



- Sometimes, it really depends on the application (next example).

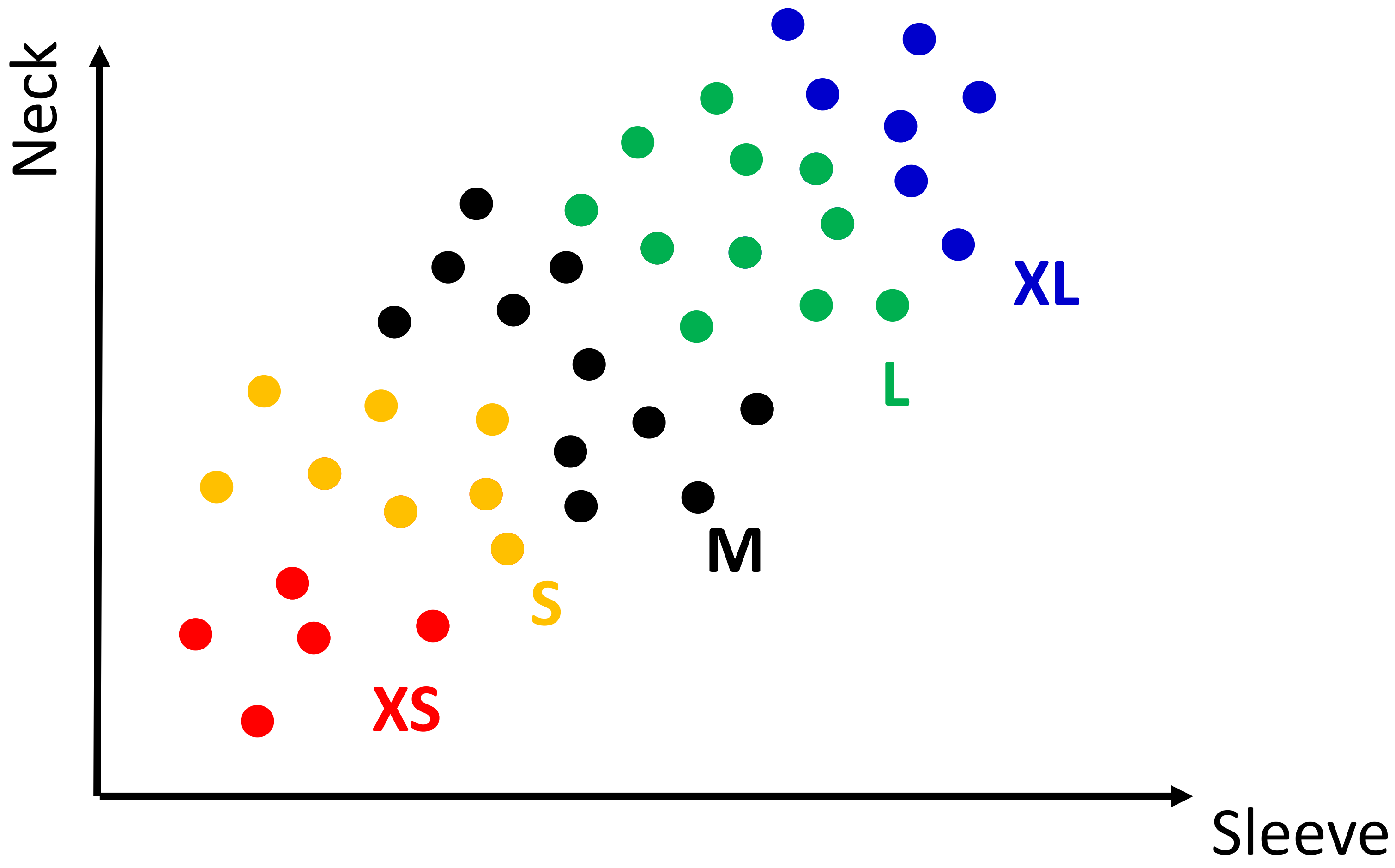
How to choose the Number of Clusters?

- **Shirt Size:** S, M, L



How to choose the Number of Clusters?

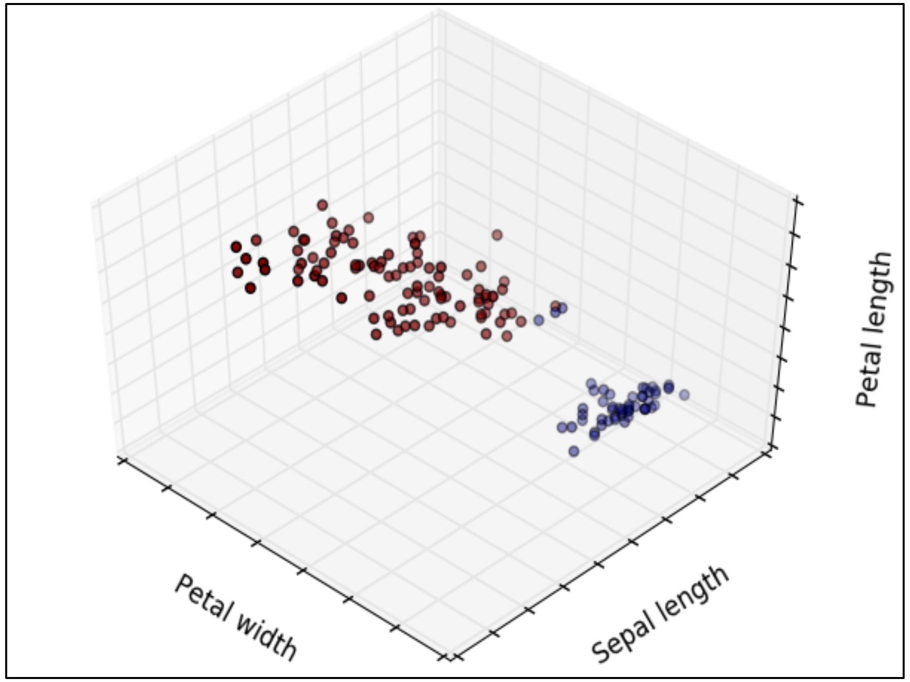
- **Shirt Size:** XS, S, M, L, XL



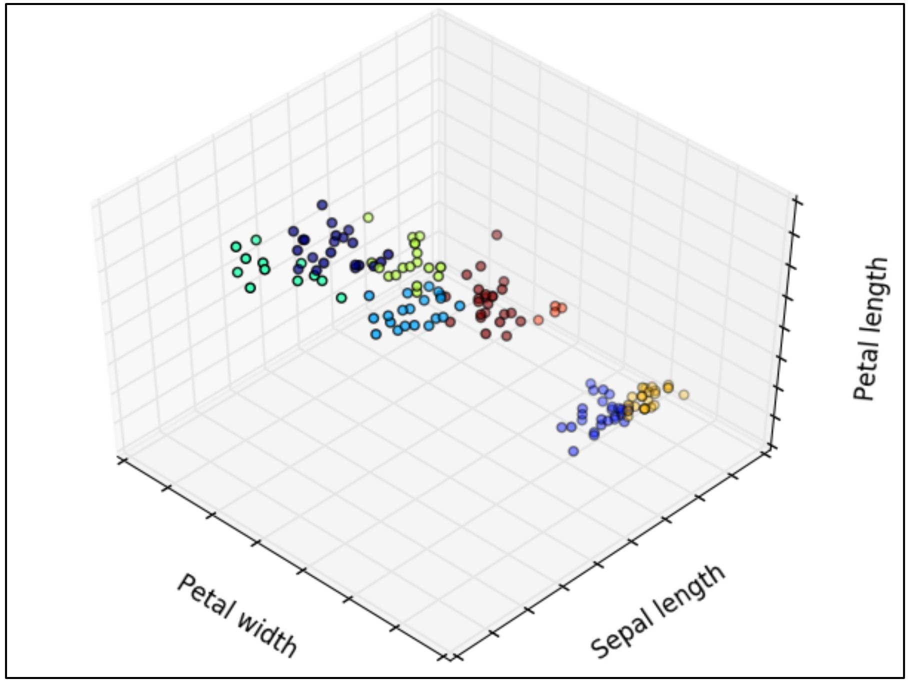
K-Means in Python

```
from sklearn.cluster import KMeans

my_Kmeans = KMeans(n_clusters=3)

my_Kmeans.fit(iris_data)
label_clustered = my_Kmeans.labels_
print(label_clustered)

my_Kmeans.predict(new_iris_data)
```

K-Means for iris dataset

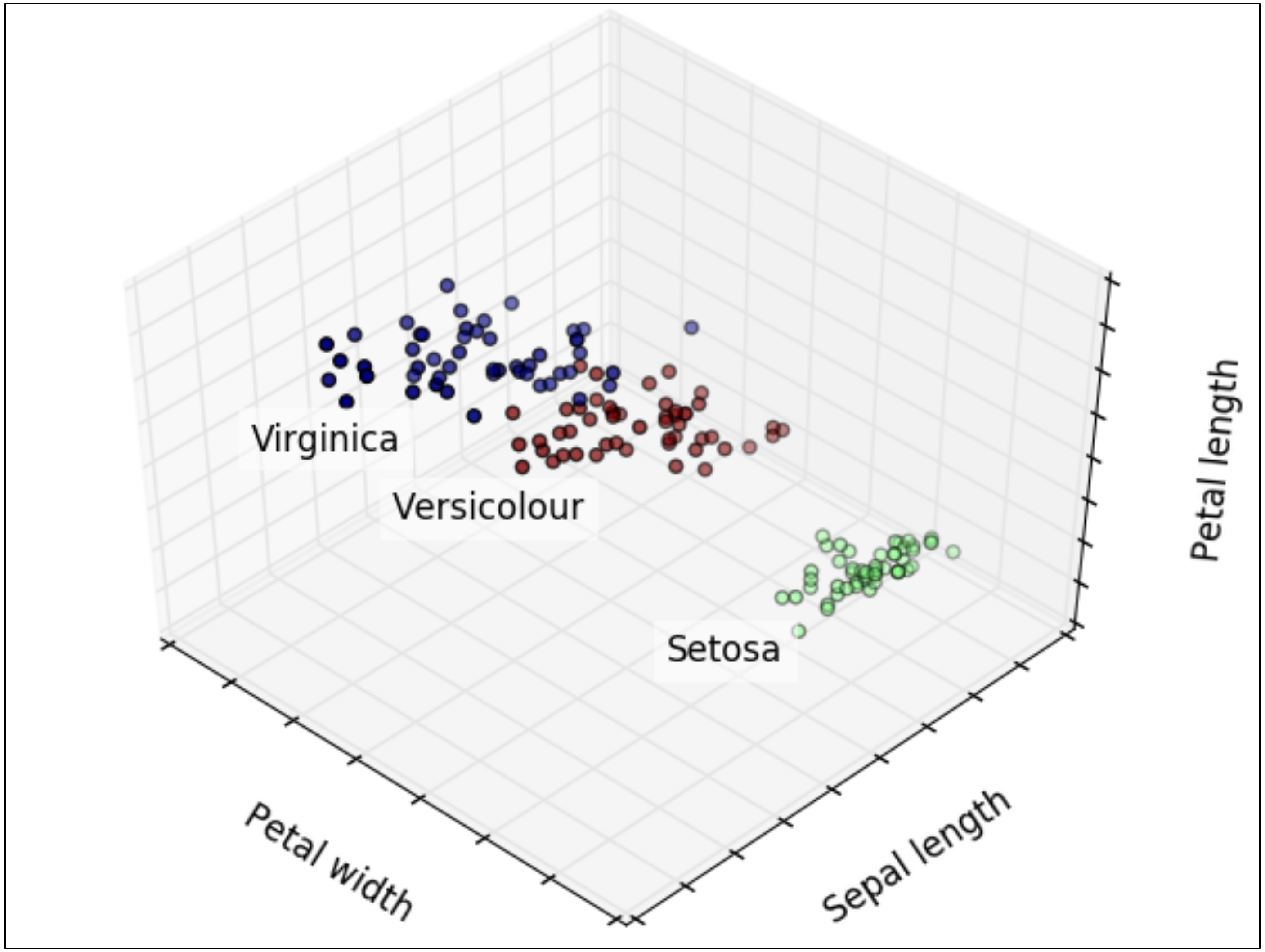


K = 2



K = 8

K-Means for iris dataset



K = 3

Thank You!

Questions?