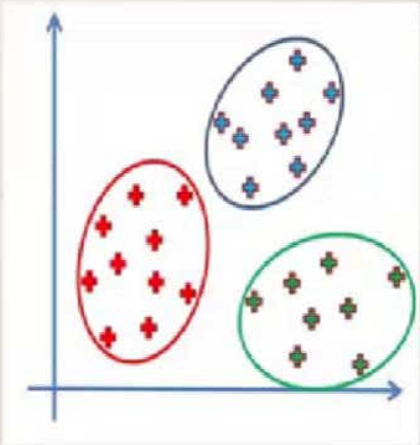
- Clustering is similar to classification, but the basis is different.
- In Clustering we don't know what we are looking for, and we are trying to identify some segments or clusters in our data.
- When we use clustering algorithms on your dataset, unexpected things can suddenly pop up like structures, clusters and groupings we would have never thought of otherwise.

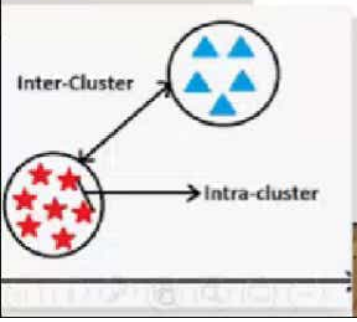


# METRICS FOR CLUSTERING

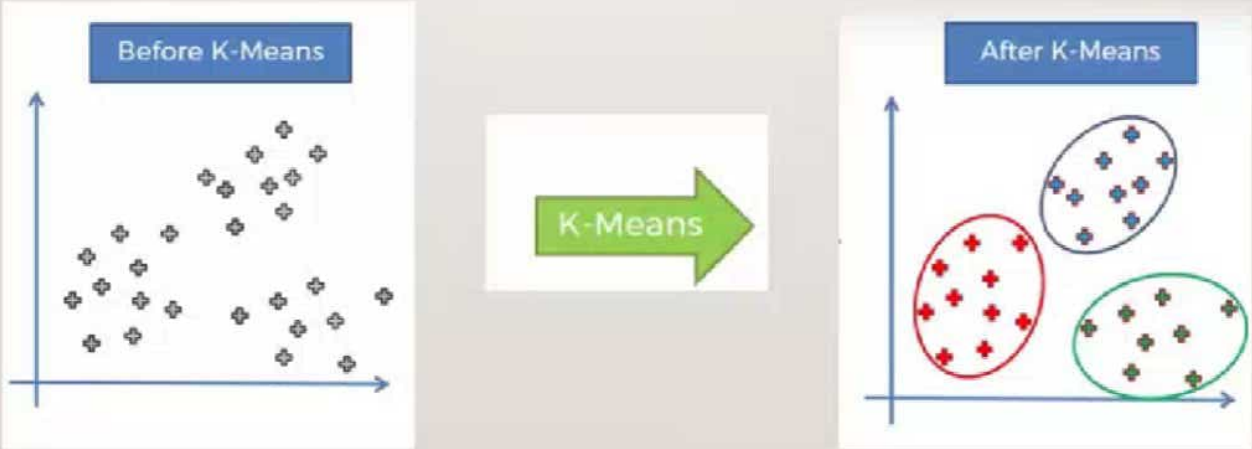
$$\text{Dunn Index} = \frac{\min(\text{Inter cluster distance})}{\max(\text{Intra cluster distance})}$$

Clusters are far apart

Clusters are compact



# CAN WE DIVIDE THE DATA INTO GROUPS??



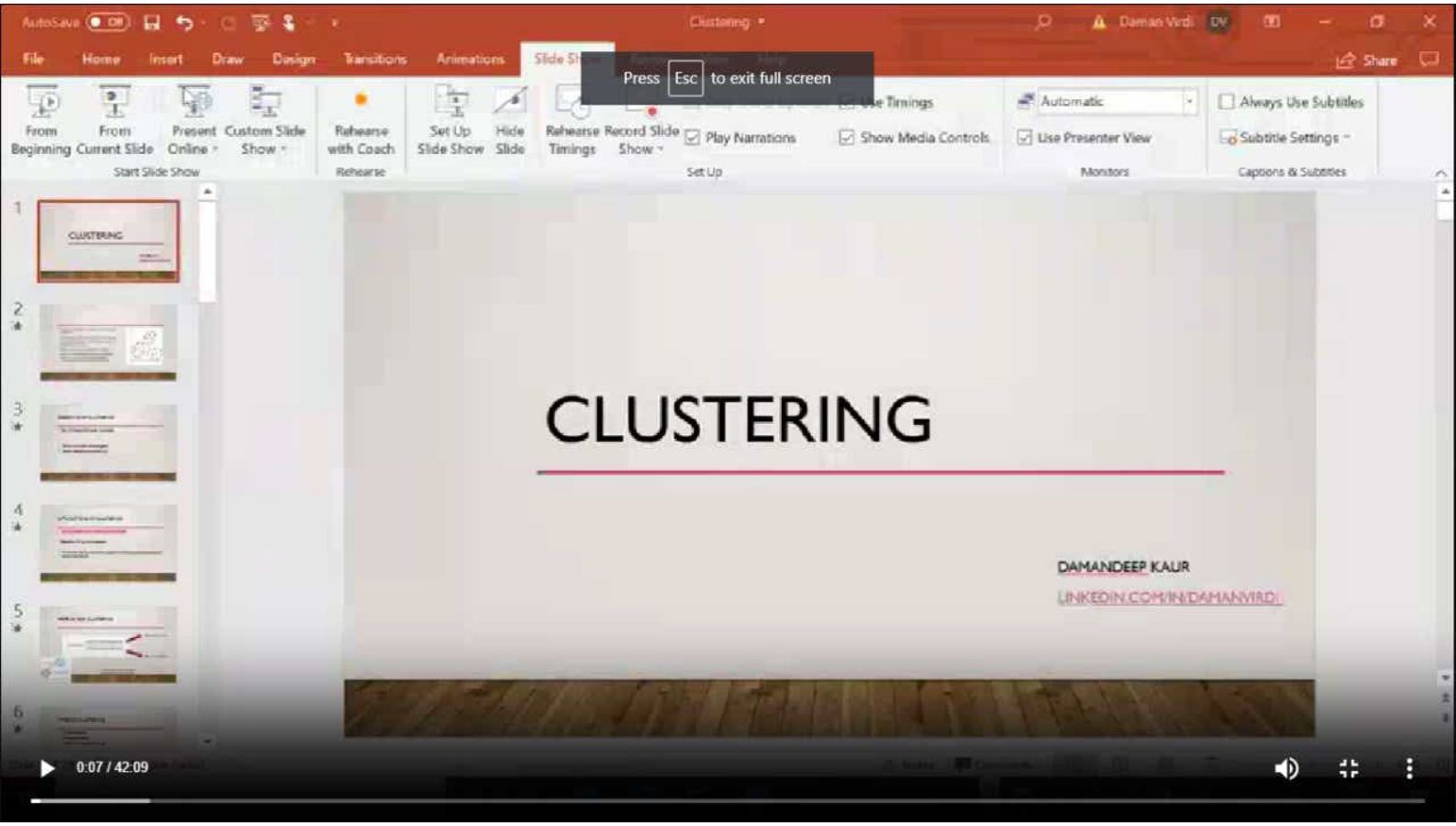
# TYPES OF CLUSTERING

1. K –Means clustering
2. Hierarchical clustering
3. DBSCAN (Density based clustering)

# OBJECTIVE OF CLUSTERING

- Task ----→ Group similar points in one cluster

1. Points in one cluster are close together
2. Points in different clusters are far away


Press Esc to exit full screen
CLUSTERING
DAMANDEEP KAUR
LINKEDIN.COM/IN/DAMANVIRDI
0:07 / 42:09

# K-MEANS IS CENTROID BASED CLUSTERING

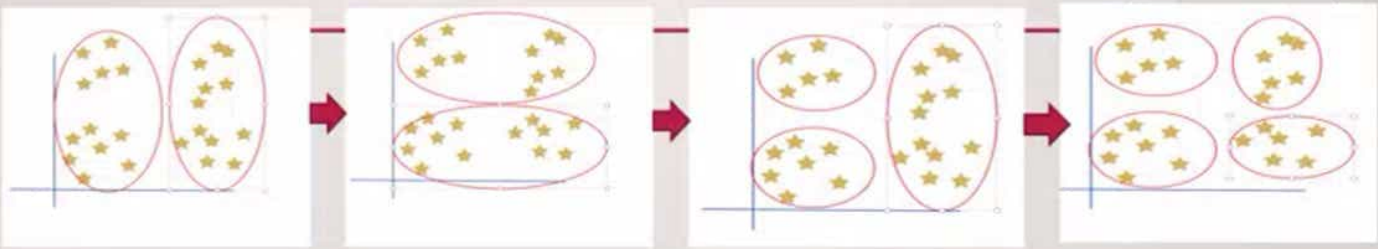
- K- no of parameters (hyper-parameter)
- HyperParameters are those which can't be deduced from the given data.

- Eg K= 3 ie. 3 Means Clustering
- No of clusters = 3
- For each cluster, No of centroids = 3 (C1 , C2 , C3)
- For each cluster , No of corresponding set of points = 3(S1, S2, S3)

- S1 U S2 U S3 = D — i.e Each point should belong to any cluster
- S1 n S2 = Ø , S1 n S3 = Ø , S2 n S3 = Ø — i.e No point belongs to more than one cluster

# APPLICATIONS OF CLUSTERING

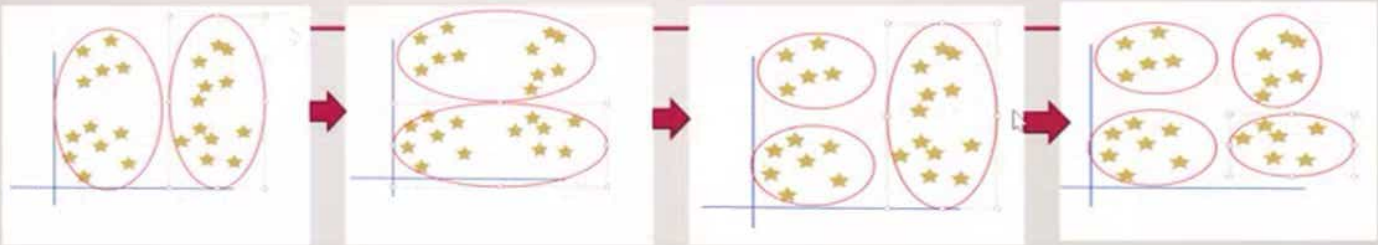
- https://en.wikipedia.org/wiki/Cluster_analysis#Applications
- Ecommerce -> To 'gp' similar customers
- Where manual labelling is time consuming, perform clustering as pre processing step and then do manual labelling



1. Points in one cluster are close together
2. Points in different clusters are far away

# K-MEANS ALGORITHM

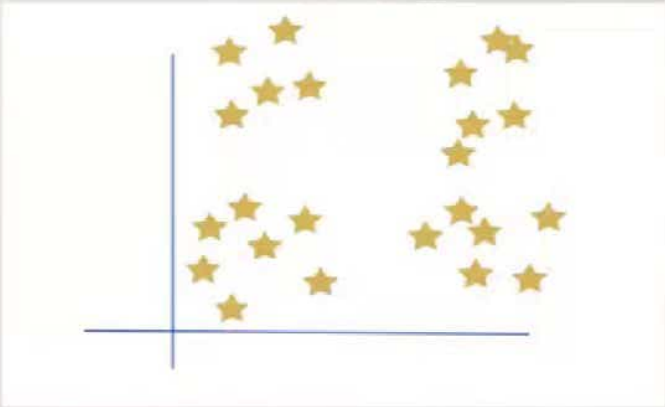
1. Choose the number K of clusters.
2. Select random K points, the CENTROIDS (not necessarily from the dataset)
3. Assign each data point to the closest centroid → It form K clusters
4. Compute and place new centroid of each cluster
5. Reassign each datapoint to the new closest centroid. If any reassignment happens , go to step4 otherwise finish.



# DETERMINING THE RIGHT K

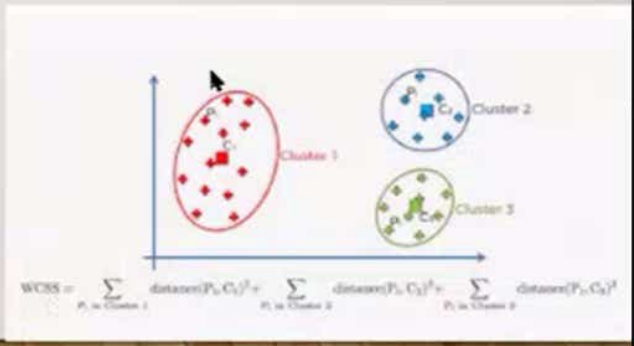
- Once we get the centroids, we can easily describe the datapoint. But first how many clusters should be there?

Q- Find no of clusters?



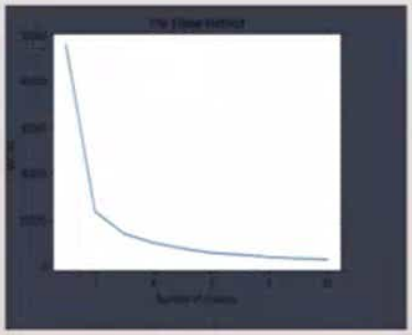
# DETERMINING THE RIGHT K

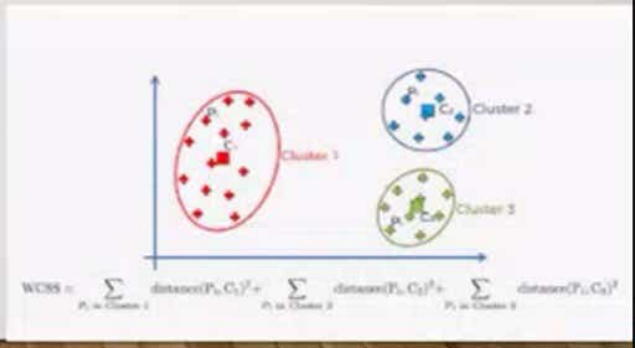
- K is Hyper Parameter i.e. It's value can't be estimated from the data
- 1. DOMAIN KNOWLEDGE e.g. –Movie Reviews.. Only three categories are possible +ve, neutral, -ve
- 2. Elbow Method :



# DETERMINING THE RIGHT K

- K is Hyper Parameter i.e. It's value can't be estimated from the data
- 1. DOMAIN KNOWLEDGE e.g. –Movie Reviews.. Only three categories are possible +ve, neutral, -ve
- 2. Elbow Method :



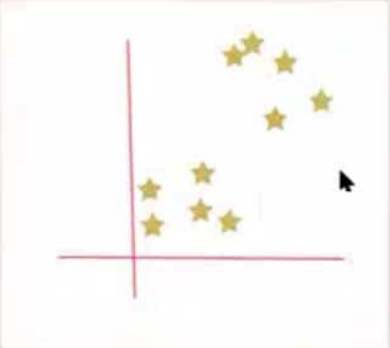


# K-MEANS ALGORITHM

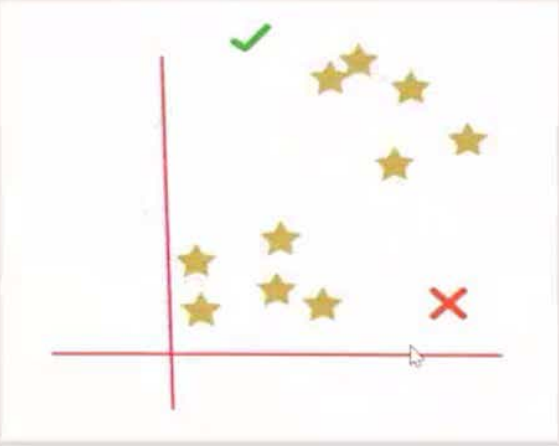
1. Choose the number K of clusters.
2. Select random K points, the CENTROIDS (not necessarily from the dataset)
3. Assign each data point to the closest centroid → It form K clusters
4. Compute and place new centroid of each cluster
5. Reassign each datapoint to the new closest centroid. If any reassignment happens , go to step4 otherwise finish.

# STEP 1: CHOOSE THE NUMBER K OF CLUSTERS

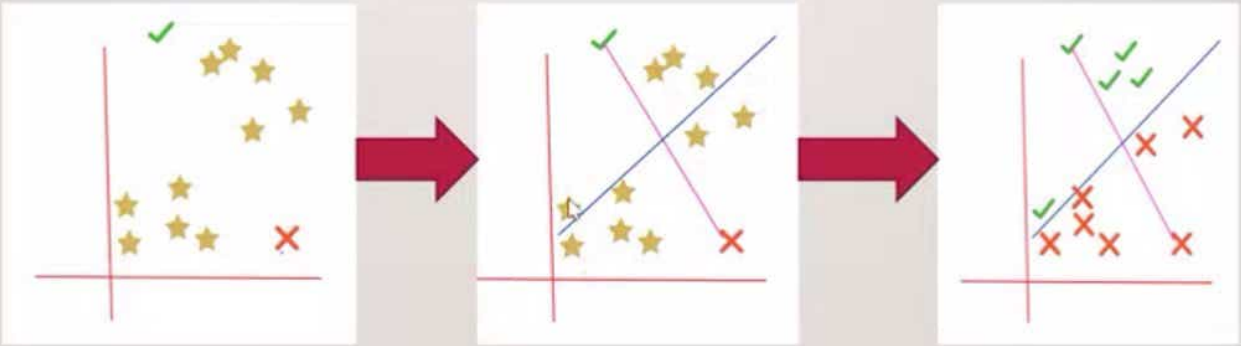
K = 2



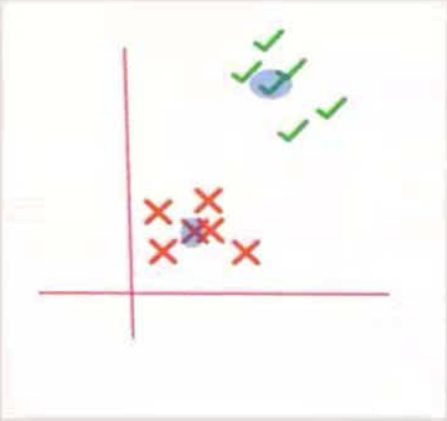
# STEP 2: SELECT RANDOM K POINTS, THE CENTROIDS



# STEP 3 - ASSIGN EACH DATA POINT TO THE CLOSEST CENTROID

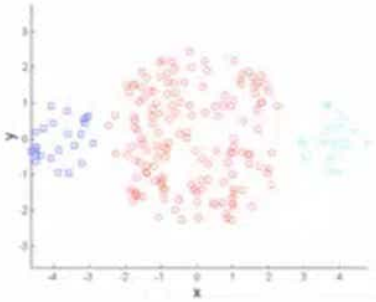


REPEAT STEP 5-REASSIGN EACH DATAPOINT TO THE NEW CLOSEST CENTROID. SINCE NO REASSIGNMENT HAPPENS ,FINISH.

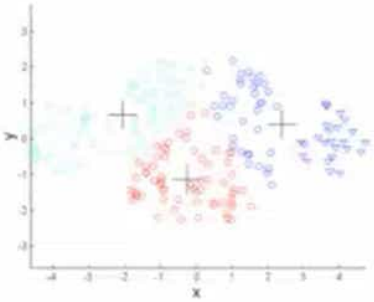


HTTPS://CS.WMICH.EDU/ALFUQAHA/SUMMER14/CS6530/LECTURES/CLUSTERINGANALYSIS.PDF

# Limitations of K-means: Differing Sizes

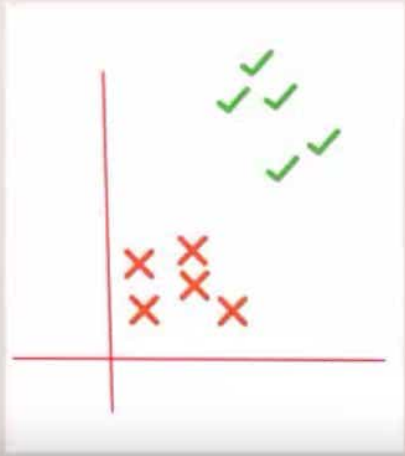



Original Points

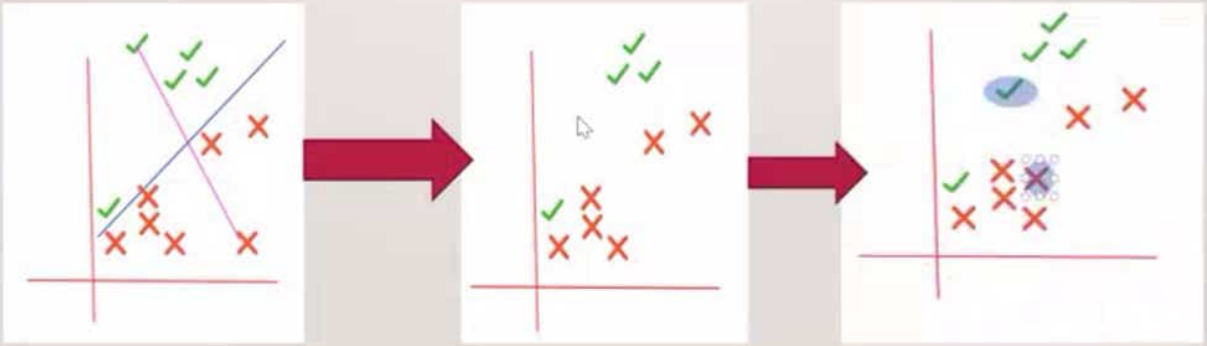



K-means (3 Clusters)

# FINAL CLUSTERS



# STEP -4 COMPUTE AND PLACE NEW CENTROID OF EACH CLUSTER



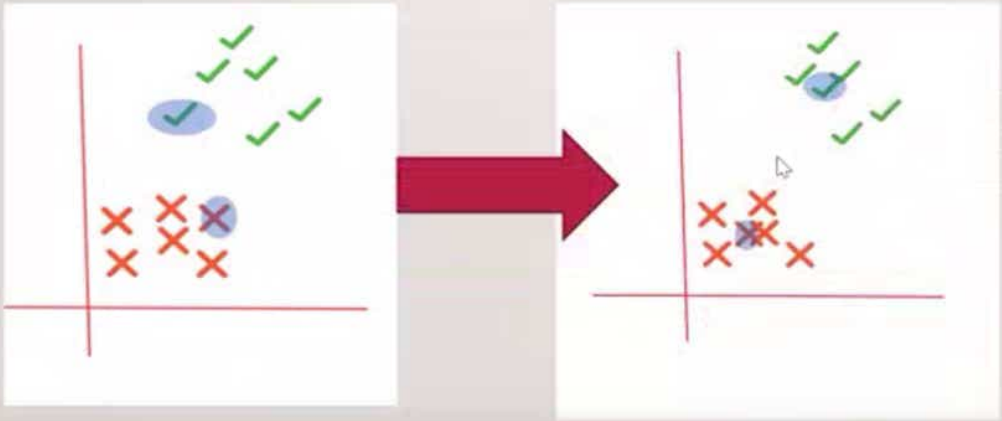
# LIMITATIONS OF K-MEANS

- K-Means have problem when clusters are of different

1. sizes.
2. Density
3. Non-globular shapes

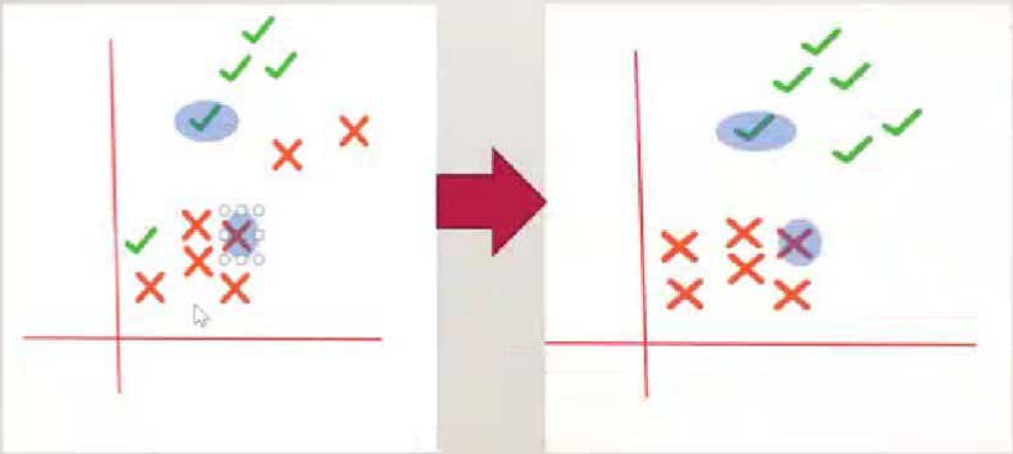
- K-Means have problem when data contains outliners

# REPEAT STEP -4 COMPUTE AND PLACE NEW CENTROID OF EACH CLUSTER



21:22 / 42:09

# STEP -5 REASSIGN EACH DATAPOINT TO THE NEW CLOSEST CENTROID



Home Page - Select or create a | Edge detection - Wikipedia | Home Page - Select or create a | k_means_clustering - Jupyter No

localhost:8890/notebooks/k_means_clustering.ipynb

File Edit View Insert Cell Kernel Widgets Help Python 3

## Using the elbow method to find the optimal number of clusters

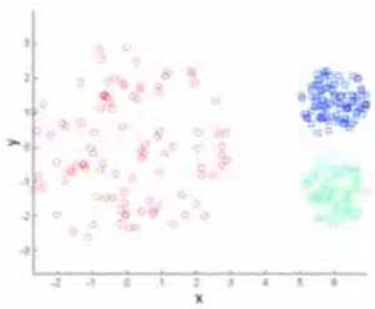
```
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

print(wcss)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
#print(wcss)
```

## Training the K-Means model on the dataset

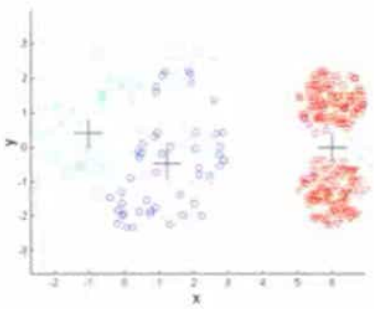
Type here to search

ENG 17:35 14-05-2021

# Limitations of K-means: Differing Density

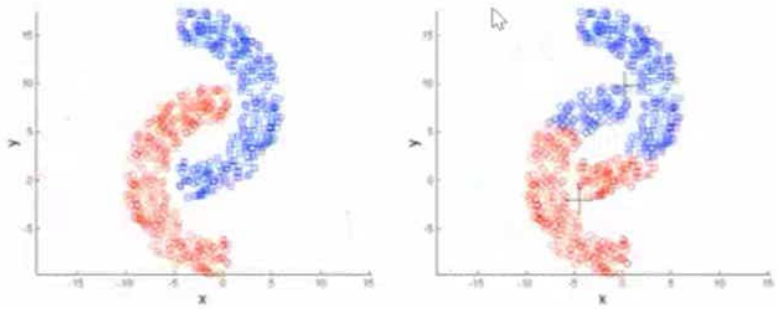



Original Points




K-means (3 Clusters)

# Limitations of K-means: Non-globular Shapes




Original Points

K-means (2 Clusters)

Home Page - Select or create a | Edge detection - Wikipedia | Home Page - Select or create a | k_means_clustering - Jupyter Ne

localhost:8890/notebooks/k_means_clustering.ipynb

File Edit View Insert Cell Kernel Widgets Help Python 3

# K-Means Clustering

## Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

## Importing the dataset

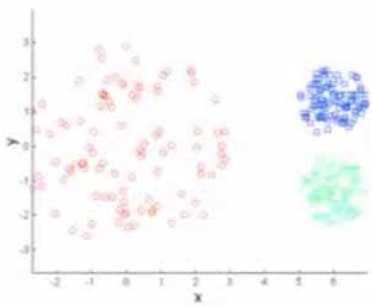
```
dataset = pd.read_csv('customer_gp.csv')
X = dataset.iloc[:, :].values
X
```

## Using the elbow method to find the optimal number of clusters

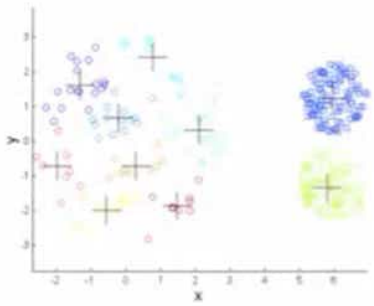
Type here to search

ENG 17:34 14-05-2021

# Overcoming K-means Limitations

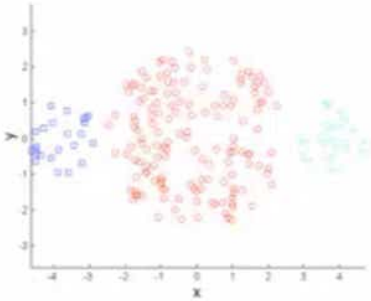



Original Points

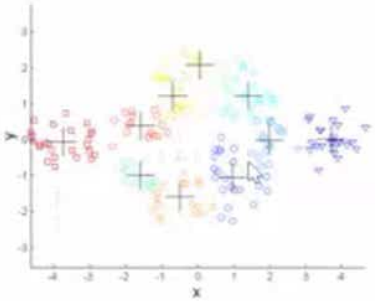



K-means Clusters

# Overcoming K-means Limitations




Original Points



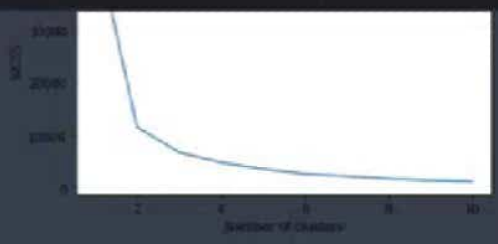

K-means Clusters

One solution is to use many clusters.
Find parts of clusters, but need to put together.

Home Page - Select or create a | Edge detection - Wikipedia | Home Page - Select or create a | k_means_clustering - Jupyter No

localhost:8890/notebooks/k_means_clustering.ipynb

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3



## Training the K-Means model on the dataset

```
kmeans = KMeans(n_clusters = 2, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(X)
print(y_kmeans)
```

## Visualising the clusters

```
X
```

```
y_kmeans
```

Type here to search ENG 17:39 14-05-2021

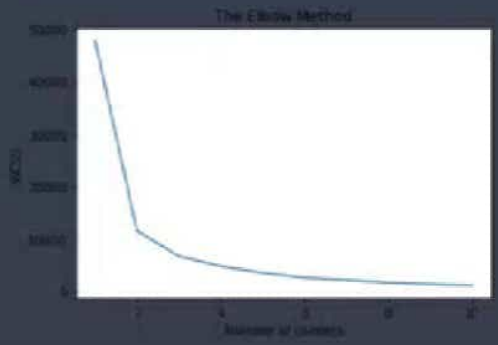
Home Page - Select or create a n | Edge detection - Wikipedia | Home Page - Select or create a n | k_means_clustering - Jupyter Not

localhost:8890/notebooks/k_means_clustering.ipynb

File Edit View Insert Cell Kernel Widgets Help Python 3

```
plt.ylabel('WCSS')
plt.show()
#print(wcss)
```

```
[47860.02, 11617.521794871793, 6804.08[illegible]115942, 4860.863005238096, 3608.5029137529136, 2658.418650793651, 2230.379365079365, 1744.9
040062040066, 1406.0984126984129, 1229.0021800952381]
```




Training the K-Means model on the dataset

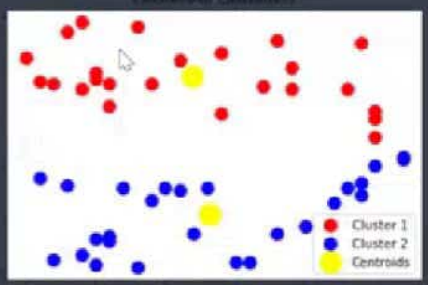
Type here to search ENG 17:38 14-05-2021

Home Page - Select or create a | Edge detection - Wikipedia | Home Page - Select or create a | k_means_clustering - Jupyter No

localhost:8890/notebooks/k_means_clustering.ipynb

File Edit View Insert Cell Kernel Widgets Help Python 3

```
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')

plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', l
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```




Type here to search

ENG 17:40 14-05-2021

Home Page - Select or create a | Edge detection - Wikipedia | Home Page - Select or create a | k_means_clustering - Jupyter No

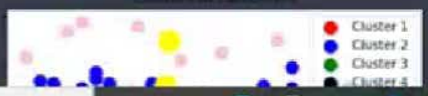
localhost:8890/notebooks/k_means_clustering.ipynb

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
x
```

```
y_kmeans
```

```
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'black', label = 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'pink', label = 'Cluster 5')

plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', l
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```




Type here to search ENG 17:46 14-05-2021