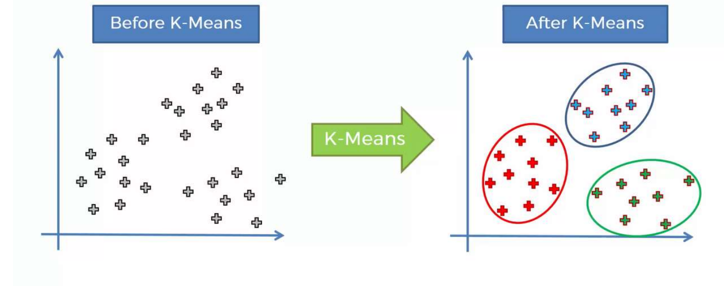
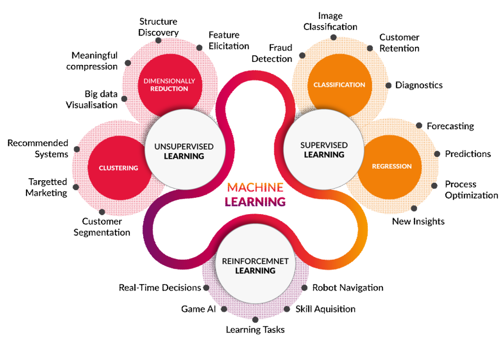
Clustering

Machine Learning

Dr. Adnan Abid

What K-Means does for you





How did it do that ?

STEP 1: Choose the number K of clusters

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)

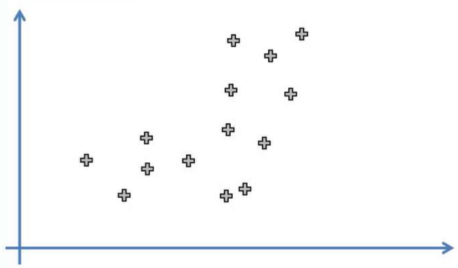
STEP 3: Assign each data point to the closest centroid ➡ That forms K clusters

STEP 4: Compute and place the new centroid of each cluster

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.

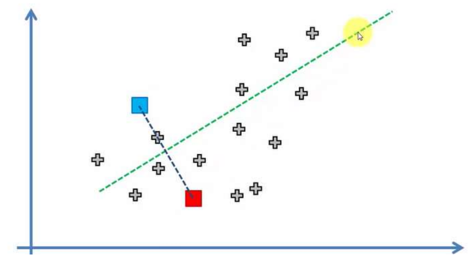
K-Means algorithm

STEP 1: Choose the number K of clusters: K = 2



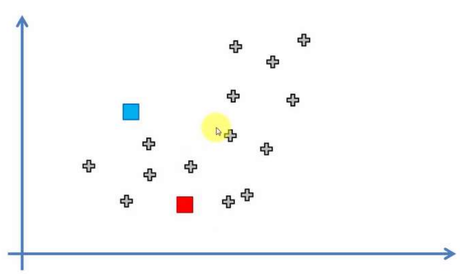
K-Means algorithm

STEP 3: Assign each data point to the closest centroid → That forms K cluster



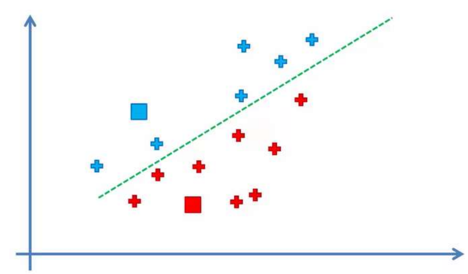
K-Means algorithm

STEP 3: Assign each data point to the closest centroid → That forms K clusters



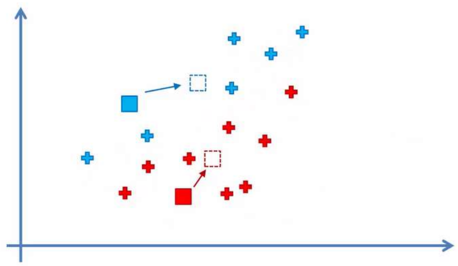
K-Means algorithm

STEP 3: Assign each data point to the closest centroid → That forms K clusters



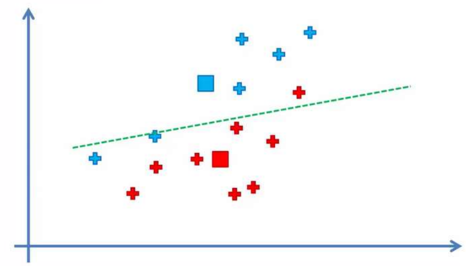
K-Means algorithm

STEP 4: Compute and place the new centroid of each cluster



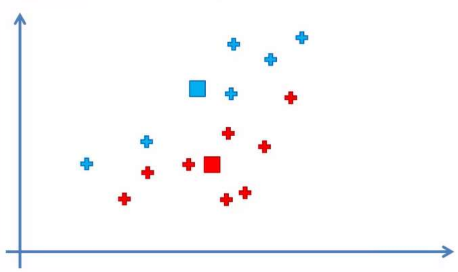
K-Means algorithm

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



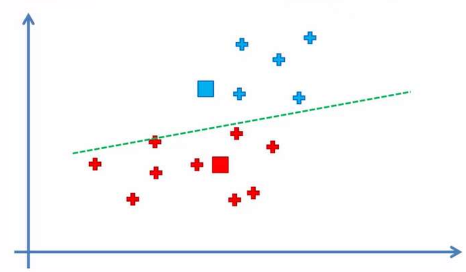
K-Means algorithm

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



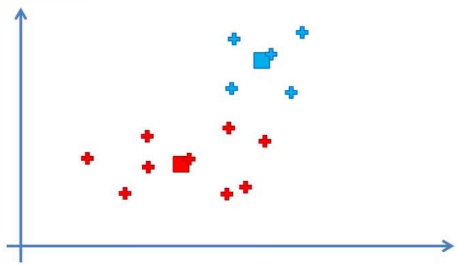
K-Means algorithm

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



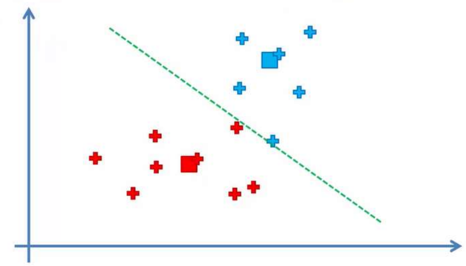
K-Means algorithm

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



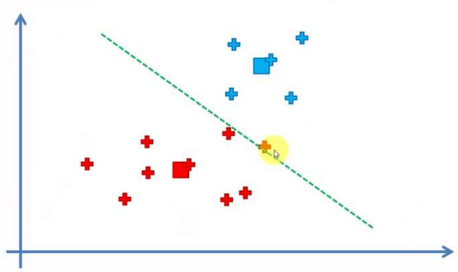
K-Means algorithm

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



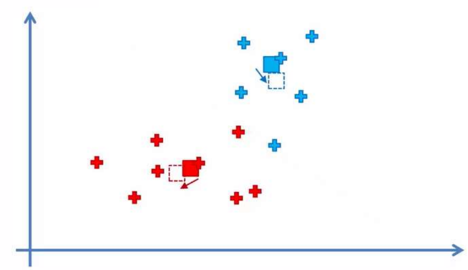
K-Means algorithm

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



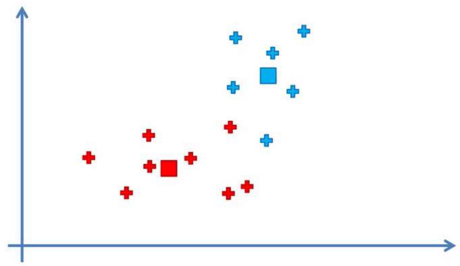
K-Means algorithm

STEP 4: Compute and place the new centroid of each cluster



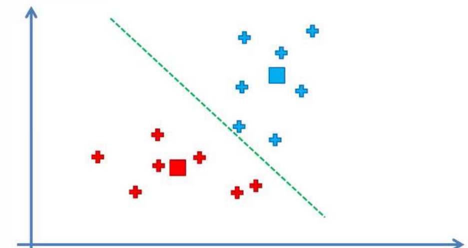
K-Means algorithm

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



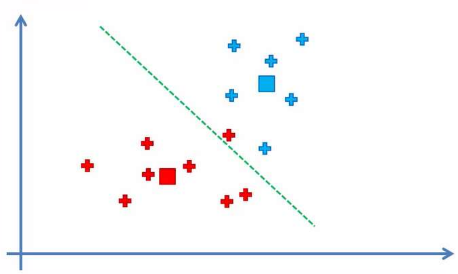
K-Means algorithm

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



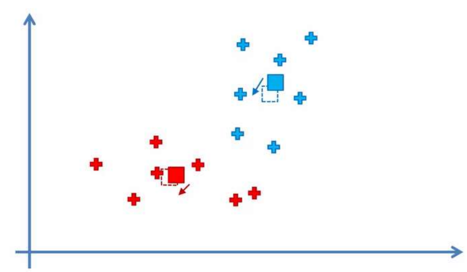
K-Means algorithm

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



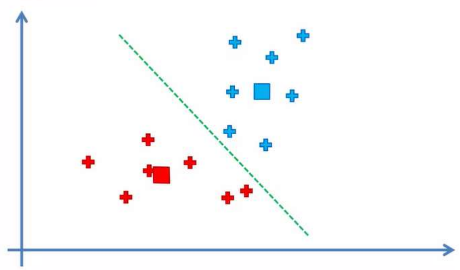
K-Means algorithm

STEP 4: Compute and place the new centroid of each cluster



K-Means algorithm

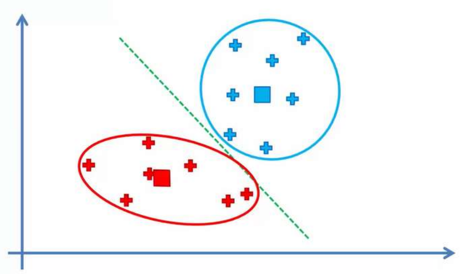
STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



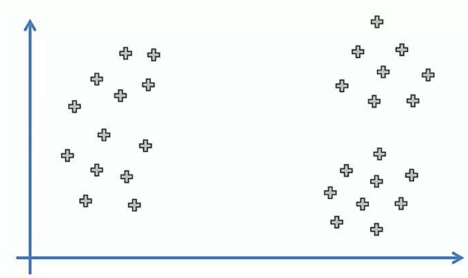
K-Means Intuition: Random Initialization Trap

K-Means algorithm

FIN: Your Model Is Ready

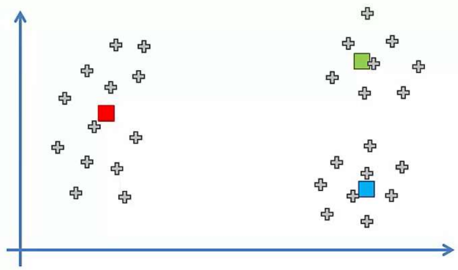


Random Initialization Trap



If we choose K = 3 clusters...

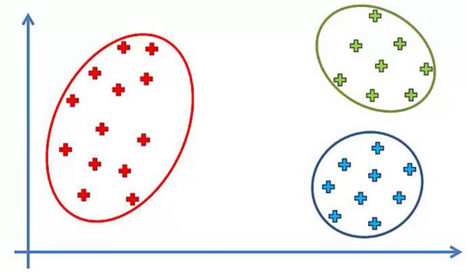
Random Initialization Trap



...this correct random initialisation would lead us to...

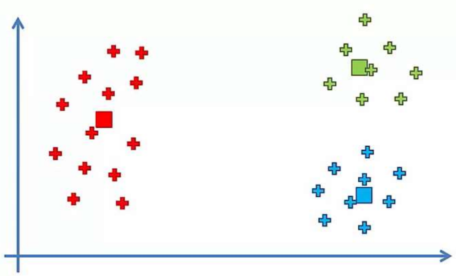
Activate Windc

Random Initialization Trap



...the following three clusters

Random Initialization Trap



...the following three clusters

Random Initialization Trap

But what would happen if we had a bad random initialisation ?

Random Initialization Trap

STEP 1: Choose the number K of clusters

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)

STEP 3: Assign each data point to the closest centroid ➡ That forms K clusters

STEP 4: Compute and place the new centroid of each cluster

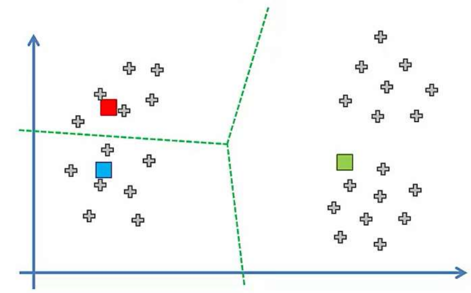
STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.

Your Model is Ready

Activate Window

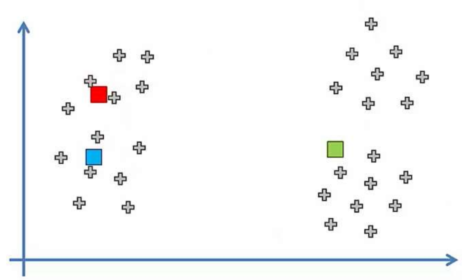
Random Initialization Trap

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)



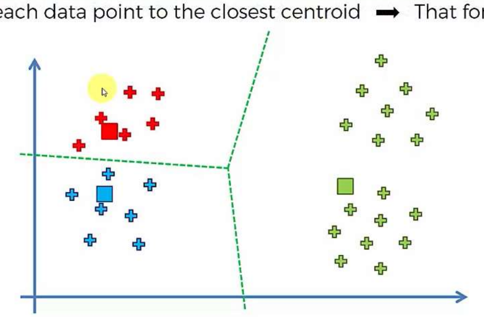
Random Initialization Trap

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)



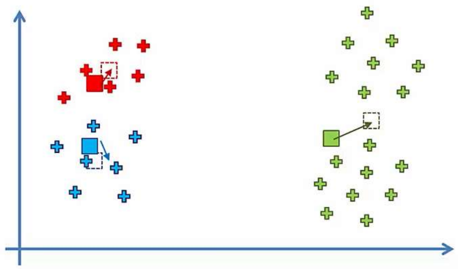
Random Initialization Trap

STEP 3: Assign each data point to the closest centroid ➡ That forms K cluste



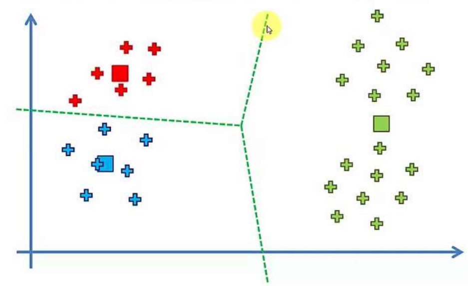
Random Initialization Trap

STEP 3: Assign each data point to the closest centroid ➡ That forms K clusters



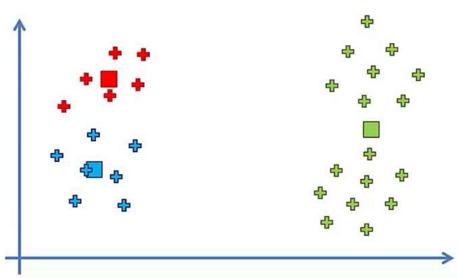
Random Initialization Trap

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



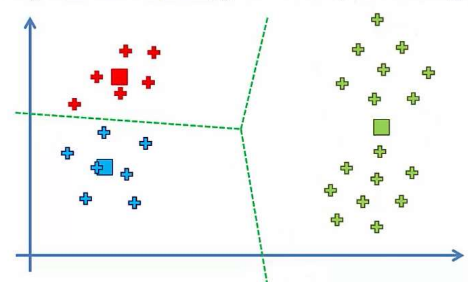
Random Initialization Trap

STEP 4: Compute and place the new centroid of each cluster



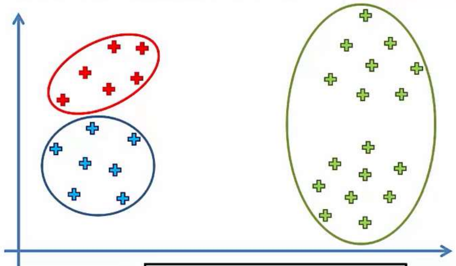
Random Initialization Trap

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



Random Initialization Trap

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



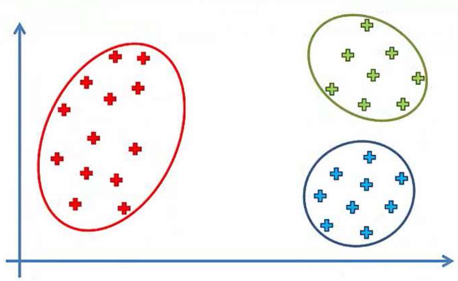
Your Model is Ready

Activate Windov

Random Initialization Trap

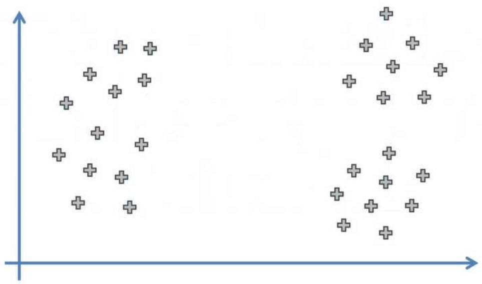
Solution → K-Means++

Random Initialization Trap



K-Means Intuition: Choosing the right number of clusters

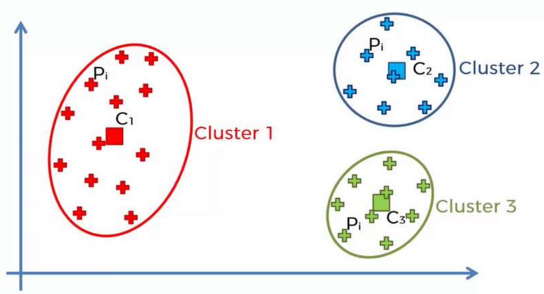
Choosing the right number of clusters



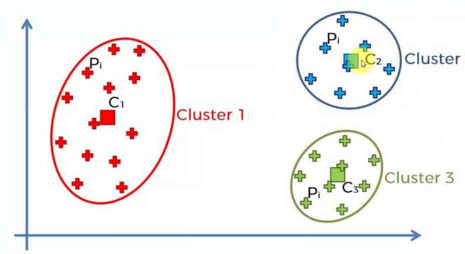
Choosing the right number of clusters

$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

Choosing the right number of clusters



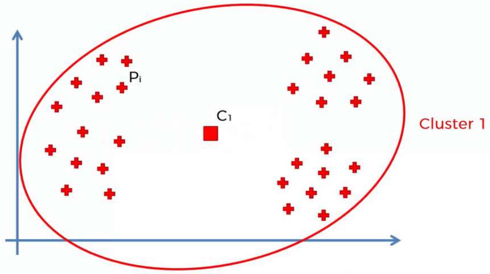
Choosing the right number of clusters



$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

Activate Windows
Go to Settings to activate Windows.

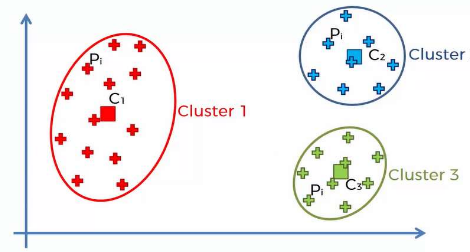
Choosing the right number of clusters



$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2$$

Activate Windows
Go to Settings to activate Windows.

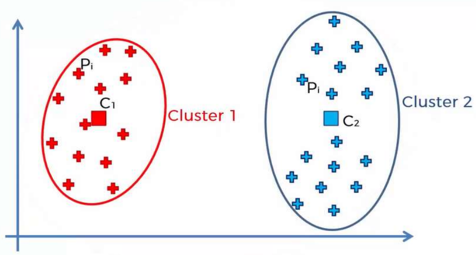
Choosing the right number of clusters



$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

Activate Windows
Go to Settings to activate Windows.

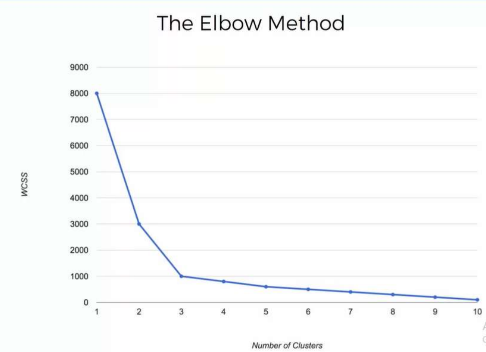
Choosing the right number of clusters



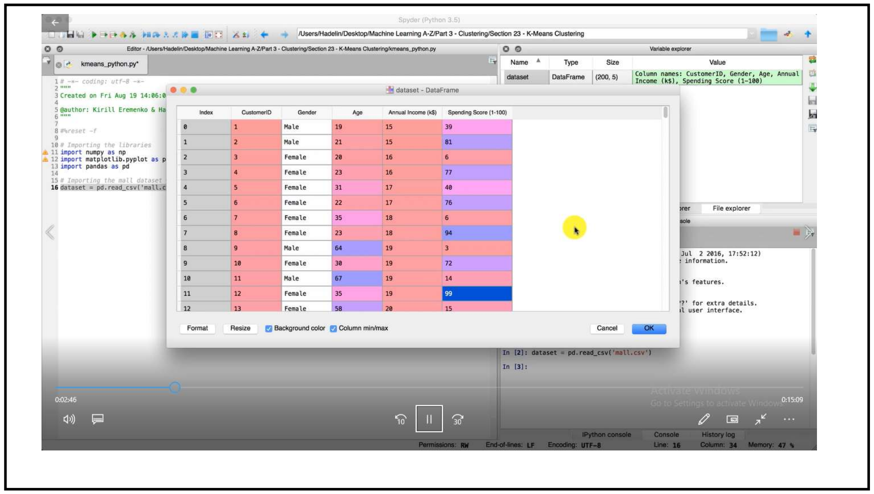
$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2$$

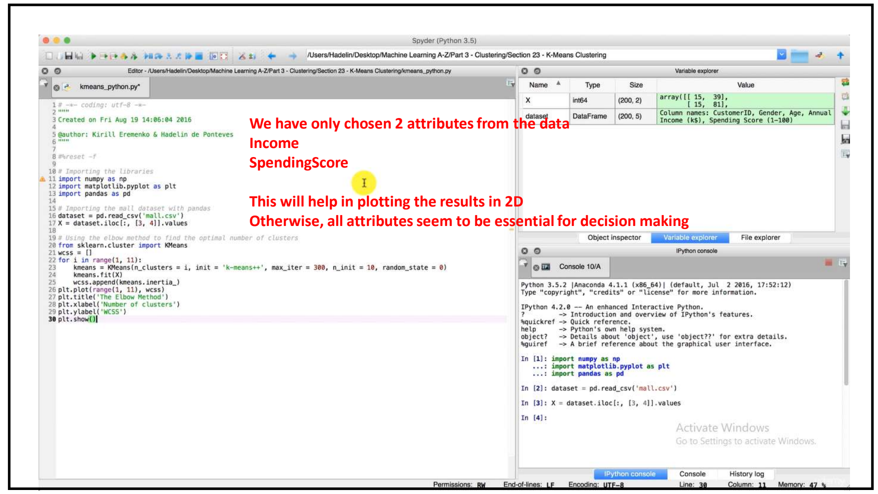
Activate Windows
Go to Settings to activate Windows.

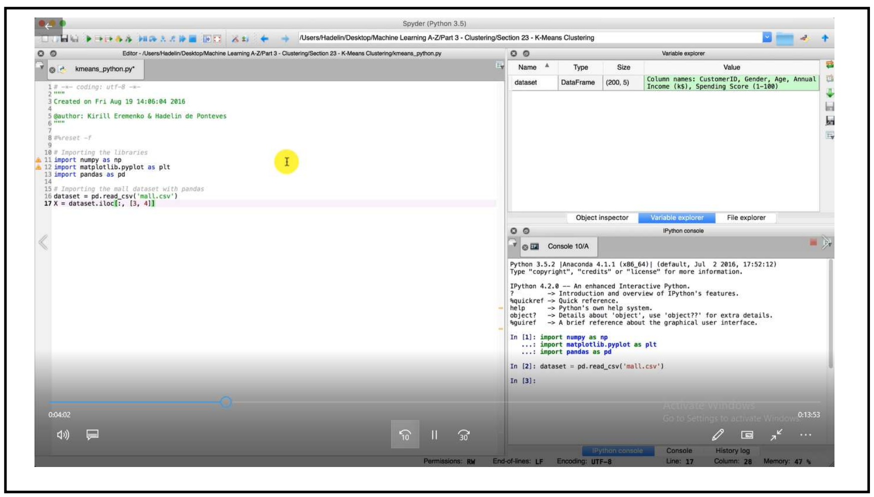
Choosing the right number of clusters

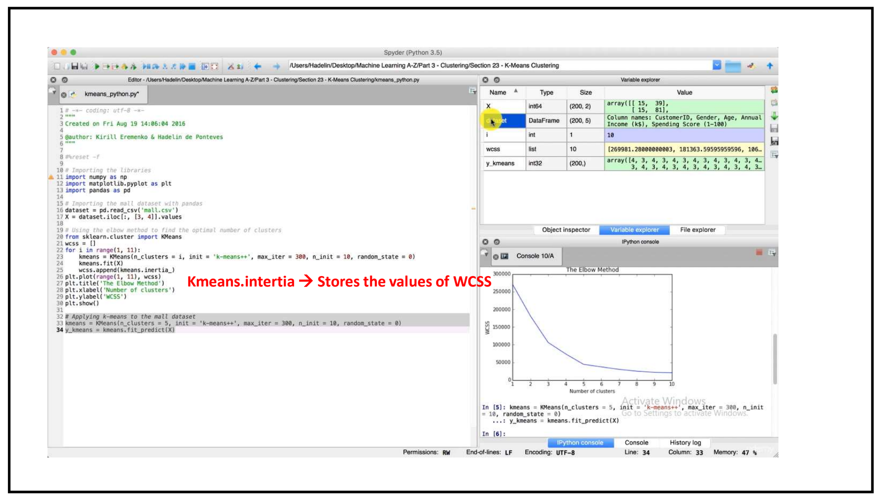


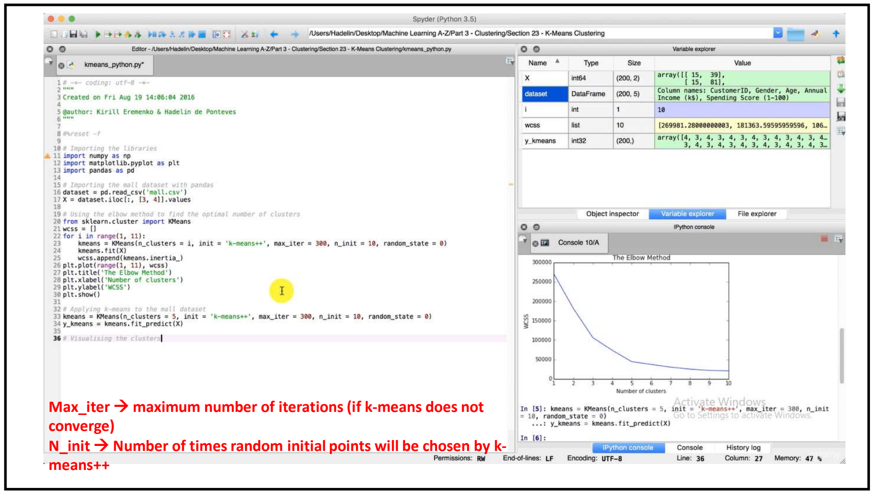
Activate Windows
Go to Settings to activate Windows.




We have only chosen 2 attributes from the data
Income
SpendingScore
This will help in plotting the results in 2D
Otherwise, all attributes seem to be essential for decision making




Kmeans.intertia → Stores the values of WCSS


Max_iter → maximum number of iterations (if k-means does not converge)
N_init → Number of times random initial points will be chosen by k-means++