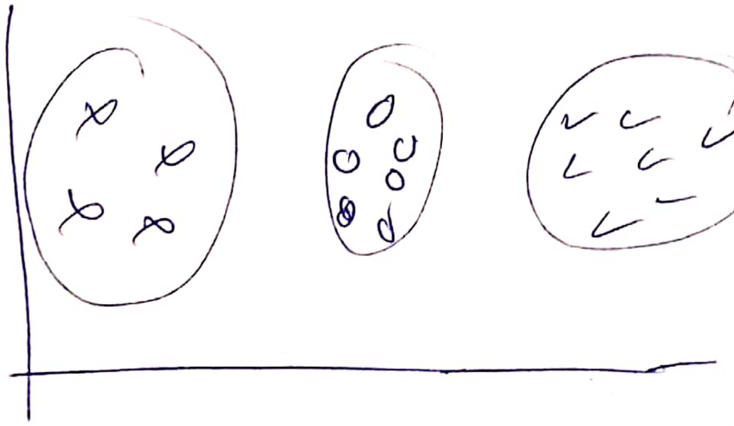
K-Means-Theory

What is clustering?

Unsupervised learning in which data is organized into distinct groups, based on similarity they exhibit.



After organizing data points into distinct clusters, we observe similarities within the clusters.

Why Clustering?

↳ Clustering helps organize Unknown observations into groups based on similarity.

↳ We do clustering when there is no ~~label~~ labeled Variable (dependant Column) in data set

It helps us to find patterns. Widely used for initial analysis of data.

Applications

* Segmenting customers based on purchasing behavior. ∴ Customer Segmentation
* Segmenting medicines based on constituents, reactions, function & results ∴ Medicine
* In psychology.

Why K-means clustering?

↳ K-means clustering performs better than Hierarchial clustering ~~(discussed in~~ in large data set.
↳ Because as data size increases computational time for ~~bi~~ Hierarchial clustering increases.

what is K-means Clustering?

↳ An unsupervised learning technique.
↳ In which data is organized into distinct groups having centroids (Mean values).
↳ K denotes number of clusters or groups.

Step 1:- Choose the Value of K - where K indicates number of clusters (we'll discuss later how to choose optimal K)

Step 2:- Initialize mean or Centroid of each cluster taking random values.

Step 3:- Calculate Euclidean distance of each attribute for each observation from each Centroid.

Step 4:- Based on nearest distance from Centroids, assign observation to clusters.

Step 5:- After each assignment recalculate mean of each attribute across all clusters.

Step 6:- Repeat Step 3, 4 & 5 until Convergence of Centroids i.e., centroids don't change significantly.

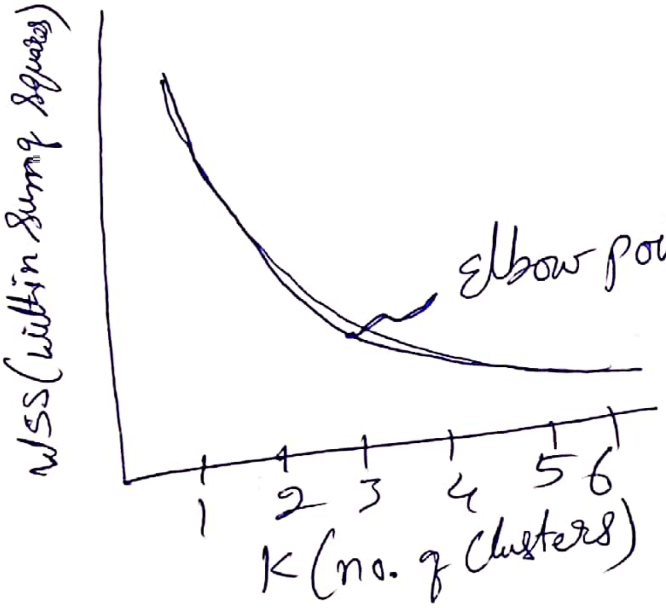
See the Example in the Pdf files

Choose optimal K

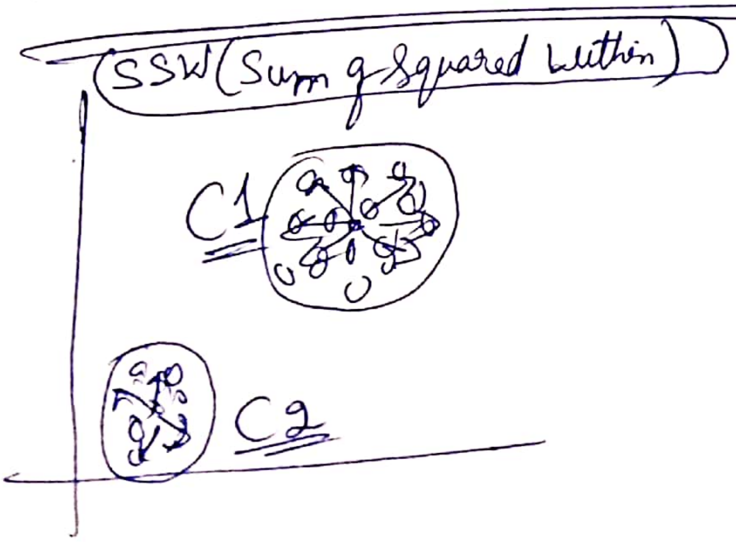
↳ Choose optimal K is the key of K-means clustering because

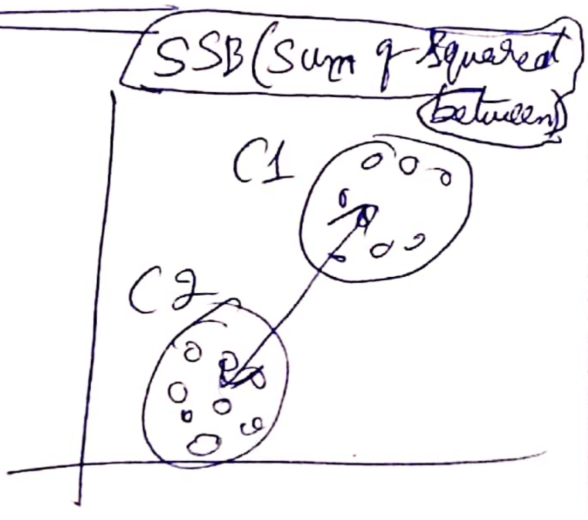
Improper selection of K may lead to erroneous assignment of observations.

There are a number of methods to choose optimal K such as Elbow method, GAP analysis, Average Silhoutte but the most common is Elbow method.



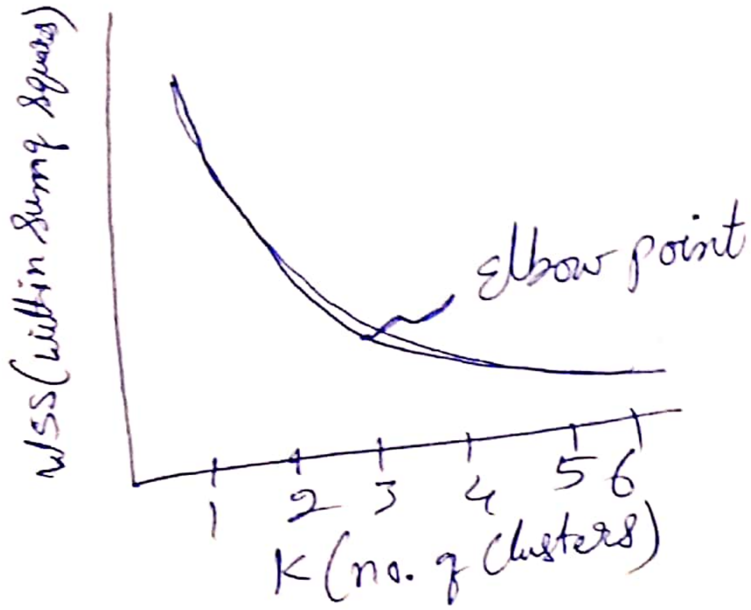
Let's know about SSW & SSB



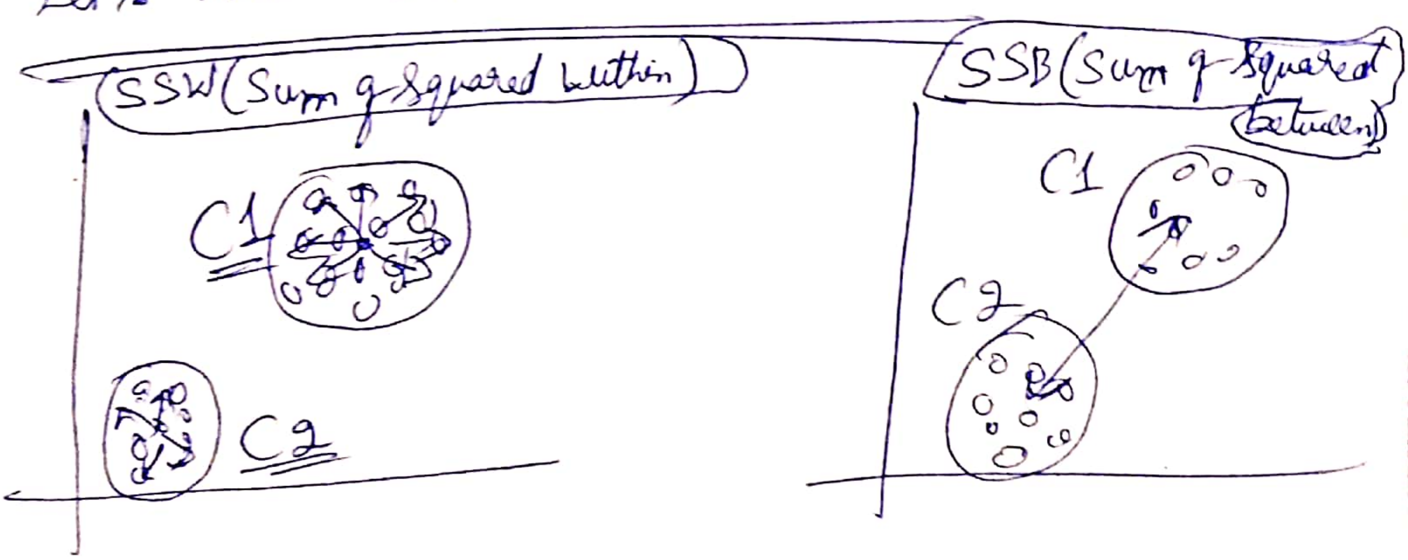


Improper selection of K may lead to erroneous assignment of observations.

There are a number of methods to choose optimal K such as Elbow method, GAP analysis, Average Silhoutte but the most common is Elbow method.



Let's know about SSW & SSB



For perfect seperation

→ We want SSW to be least.

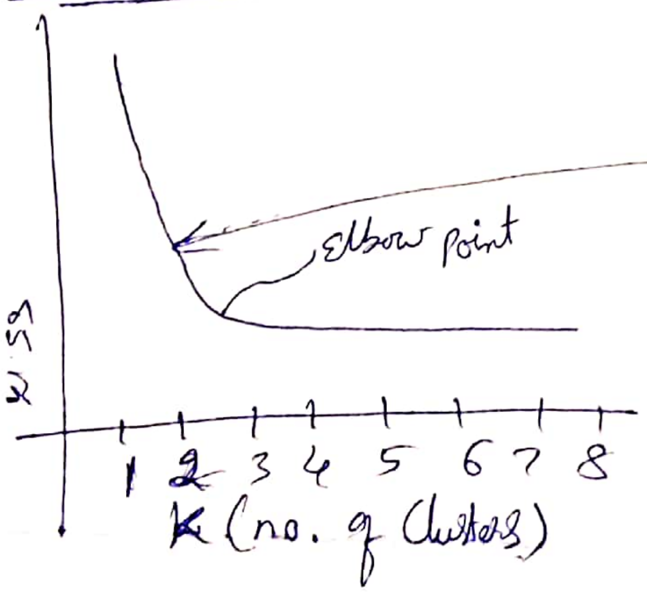
→ The smaller SSW the more compact data points are in clusters.

↳ So that there is no intersection between 2 clusters.

→ SSB should be maximum

→ Max SSB indicates centroids are distantly & perfectly positioned.

Choose optimal K



↳ Elbow method is very useful to find optimal K.

↳ In this method SSW (within sum of square) decrease as number of clusters increases.

↳ Elbow method helps us to find optimal K with increasing number of K.

↳ We should take that number of K where elbow is spotted

↳ Adding clusters beyond elbow point doesn't significantly reduce SSW.

K-Means

(1) Import packages (Load the data)
(3) Clean Data — [Nan/fill/drop ; outlier treatment] numpy, Pandas
(2) EDA → All visualizations should be done here ← matplotlib / Seaborn + statistics

→ Boxplot
→ Correlation
→ Histogram
→ pairplot

Since it is an unsupervised learning; we do not have to (1) Create features / labels & (2) split data => Not applicable

$\boxed{\text{Step 1}}$

Finding clusters with Elbow method

```
SSW = []
cluster_range = range(1,10)
for i in cluster_range:
    model = KMeans(n_clusters = i, init = 'K-means++',
                   max_iter = 300, random_state = 0)
    model.fit(X)
    SSW.append(model.inertia_)
```

ent

→ Print the clusters with SSW (use dictionary)

```
SSW_df = pd.DataFrame({..... : ....., ......})
Print(SSW_df
```

Plot the clusters

```
plt.figure(figsize=(10,17))
plt.plot(cluster_range, SSW, marker=o, color=" ")
plt.xlabel(" ..... ")
plt.ylabel(" ____ ")
plt.title(" ...... ___ ...... ")
plt.show()
```

Step 2

Build a K-means model

```
Kmeans = KMeans(n_clusters=4 , init = "Kmeans++",
                n_init = 10, random_state =42)

Y_kmeans = Kmeans.fit_predict(x)
```

→ Fitting the model

Scanned by CamScanner

Step - 3

Visualizing the Clusters

plt. Scatter (X[Y-means == 0, 0], X[Y-means == 0, 1], S = 100, C = purple', label = Cluster 1

- - - - - == 1, 0], [- - - - = 1, 1,] - - - Cluster 2

2, 0] [- - - 2, 1] — Cluster 3

3, 0) [- - - 3, 1] — Cluster 4

plt. Scatter (Kmeans. Cluster Centers_[:, 0], Kmeans Clusters_[:, 1]
S = 300, marker = "s", C = 'red', label = 'Centroids')

plt. title (' - - — —')

plt. xlabel (— — - -)

plt. ylabel (— — -)

plt. Legend ()

plt. Show.