FAQ’s K Means Clustering

1. What is K-means clustering?

K-means is an Unsupervised Learning algorithm, it is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a data point to a cluster. In K-Means, each cluster is associated with a centroid. Here K defines number of pre-defined clusters that needs to be created in the process.

2. Why K-means clustering?

- It is very smooth in terms of interpretation and resolution.
- For a large number of variables present in the dataset, K-means operates quicker than hierarchical clustering.
- K-means reforms compact clusters.
- It can work on unlabeled numerical data.

3. What are some applications of clustering in real-world scenarios?

Common applications of clustering include

- Customer Segmentation
- Document Clustering
- Image Segmentation
- Recommendation Engines

4. What are the limitations of K-means clustering?

- Sometimes, it is quite tough to figure out the appropriate number of clusters, or the value of k.
- The output is highly influenced by the original input, for example, the number of clusters.
- It gets affected by the presence of outliers in the data set.
- In some cases, clusters show complex spatial views, then executing clustering is not a good choice.

5. What are some applications of clustering in real-world scenarios?

Common applications of clustering include

- Customer Segmentation
- Document Clustering
- Image Segmentation
- Recommendation Engines

6. Is there any metric to compare clustering results?

You can compare clustering results by checking silhouette scores and by doing cluster profiling. Besides this, you should also validate the clustering results by consulting with a domain expert to see if the cluster profiles make sense or not.

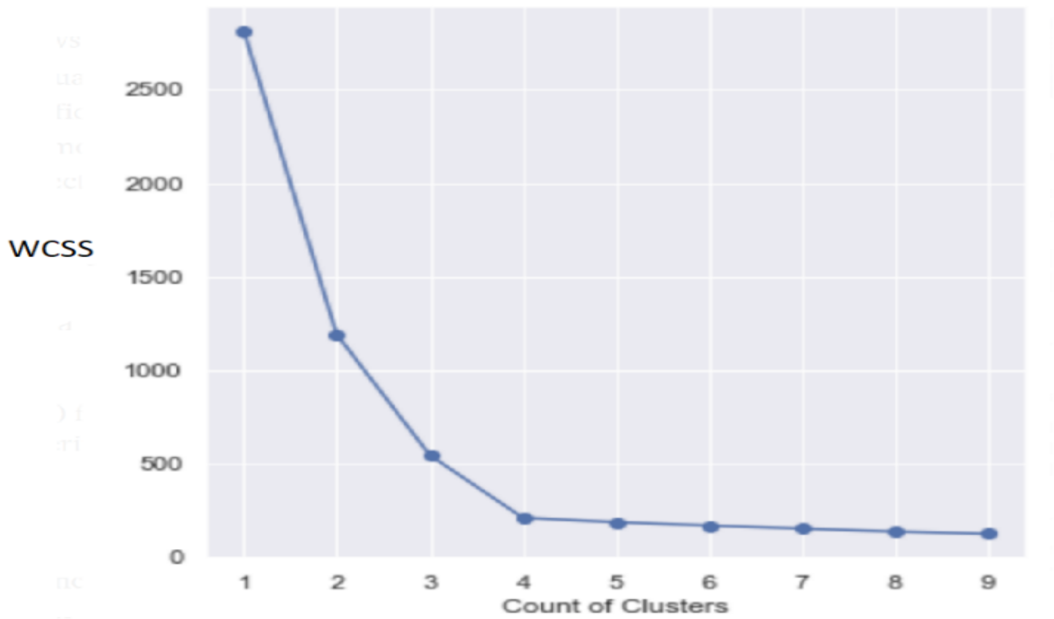
7. For K-means if there is a y-dependent variable, do we remove it before trying to group customers?

Yes, if you have a dependent variable in your dataset, you should remove that before applying clustering algorithms to your dataset.

8. How do we select the optimal number of clusters from the Elbow curve?

Choosing the optimal number of clusters is a fairly subjective matter, and the best method to identify the optimum number of clusters is to use a combination of metrics and domain expertise. The Elbow curve is one of the most common ways of finding the right number of clusters for K-Means clustering if we don't have domain expertise. The elbow curve is plotted between the number of clusters on the X-axis and WCSS (within the cluster sum of squares) on the Y-axis.

The elbow method uses the WCSS to choose an ideal value of k based on the distance between the data points and their assigned clusters. WCSS is the sum of the squared distance between each point and the centroid in a cluster. We would choose a value of k where the WCSS begins to flatten out, and we see an inflection point.



The graph above shows that k = 4 is an appropriate number of clusters to choose from, with an obvious elbow at that number. At K=4, the graph shows a significant fall in WCSS. As a result, 4 is the best K-value.

Codes to import K Means and related libraries used in K means clustering

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

** HAPPY LEARNING **