**K means Algorithm**

After determining the optimal number of clusters using the elbow method, I proceed to train the K-Means algorithm on my dataset. First, I ensure that I have all the necessary libraries such as Scikit-learn for the K-Means algorithm, and I prepare my dataset, converting it to a suitable format like a NumPy array or a Pandas DataFrame. I then define the optimal number of clusters, which I identified from the elbow plot.

Next, I train the K-Means model using the KMeans class from Scikit-learn, specifying the optimal number of clusters. The fit method is used to train the model on my dataset. After training, I retrieve the cluster centroids, which represent the center of each cluster, and the labels assigned to each data point, indicating which cluster each data point belongs to. I add these labels to my original DataFrame for further analysis.

**Resulting clusters are visualised and labelled for interpretation and actionable insights.**

After training the K-Means algorithm on my dataset and determining the clusters, I visualize and label the resulting clusters to gain actionable insights. This process involves creating 2D or 3D plots, depending on the dimensionality of the data. For high-dimensional data, I use dimensionality reduction techniques like PCA to project the data into a 2D or 3D space, making it easier to visualize.

I use Matplotlib to create scatter plots, where each point represents a data point colored by its assigned cluster label, and the centroids of the clusters are marked to show their positions. This visual representation helps in understanding the spread and separation of the clusters.

After visualizing the clusters, I add the cluster labels to my original dataset for further analysis. This involves examining the characteristics of each cluster by analyzing the centroids and the distribution of features within each cluster. For instance, by grouping the data by cluster labels and calculating summary statistics, I can identify the typical attributes of each cluster.

This detailed analysis allows me to interpret the clusters and derive actionable insights. For example, if the clusters represent different customer segments, I can identify high-value customers, occasional buyers, and customers at risk of churn. For high-value customers, I might develop loyalty programs and personalized offers to enhance retention. For occasional buyers, offering discounts and promotions could encourage more frequent purchases. For customers at risk of churn, I can implement engagement campaigns to re-engage them and understand their needs better.

