Model Training Documentation

In this project, we trained a K-Means clustering model to segment customers based on their features.

# What is K-Means Clustering?

# K-Means clustering is an unsupervised machine learning algorithm used to partition a dataset into K distinct, non-overlapping subsets (clusters). The algorithm aims to minimize the within- cluster sum of squares (inertia), which measures the variance within each cluster.

# Steps Involved in K-Means Clustering:

1. Initialisation: From the dataset, choose K initial centroids at random.   
2. Assignment: Create K clusters by allocating each data point to the closest centroid.  
3. Update: Take the mean of all the data points in each cluster to recalculate the cluster centroids.   
  
4. Repeat: Continue the assignment and update procedures until the maximum number of iterations is reached or the centroids stop changing.   
The objective is to maximise the distance between clusters and minimise the distance between data points inside the same cluster.

# Model Training Steps:

1. Data Loading: The previously processed dataset was brought into use.   
2. Feature Selection: The target variable 'Churn\_Yes' was ignored when choosing features for clustering.   
3. Feature Scaling: To standardise the data, the StandardScaler was used to scale the features.   
4. Model Training: The Elbow approach was utilised to estimate the ideal number of clusters (3) for the model, which was then trained using the K-Means algorithm.   
5. Model Saving: Joblib was used to save the K-Means model from training.   
6. Saving Results: A CSV file was created using the clustering results. Understanding client segments and creating focused strategies are made easier with the help of the trained model, which can be used to forecast cluster labels for new data points.

# Code Snippet for Training the Model:

import joblib

import pandas as pd

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

# Load the dataset

url = 'https://drive.google.com/uc?id=1qBkAiPPǪ9bTiaY6PcDq7Tmp8DǪsCyyrZ' df = pd.read\_csv(url)

# Convert categorical variables to numeric df\_encoded = pd.get\_dummies(df)

target\_column = 'Churn\_No'

# Separate features and target variable

X = df\_encoded.drop([target\_column, 'Churn\_Yes'], axis=1) # Drop the target variables

# Standardize the features scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Train the clustering model (using 3 clusters as an example) kmeans = KMeans(n\_clusters=3, random\_state=42) kmeans.fit(X\_scaled)

# Save the trained model joblib.dump(kmeans, 'kmeans\_model.pkl')

print("Clustering model trained and saved as 'kmeans\_model.pkl'")