# User Segmentation using KMeans Clustering

**Name: Shivam Pandey**

**Roll No: 202401100300234**

**Problem Statement**:

To segment users into meaningful groups based on their preferences and demographic data using KMeans clustering and visualize the results using Principal Component Analysis (PCA).

## Introduction

In many real-world applications such as marketing, recommender systems, and personalization engines, grouping users with similar traits is essential. This project applies KMeans clustering, an unsupervised machine learning algorithm, to classify users based on their features. By clustering the data and reducing its dimensions with PCA, we can visualize and interpret group behavior more clearly.

## Methodology

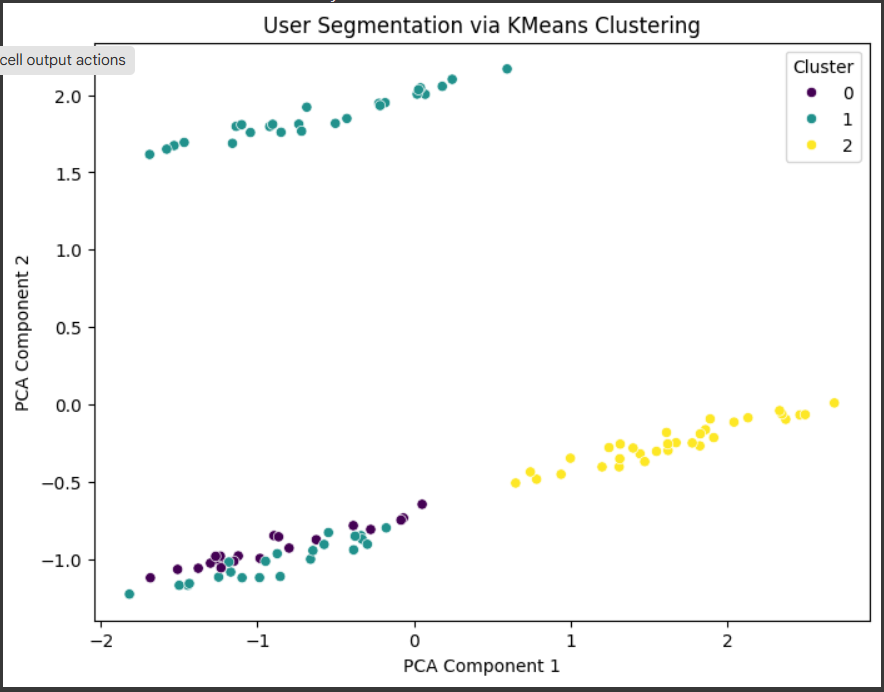
1. Data Upload & Loading:  
 - The dataset is uploaded via Google Colab and read using Pandas.  
  
2. Categorical Encoding:  
 - The column 'genre\_preference' is encoded using one-hot encoding.  
  
3. Feature Scaling:  
 - Features are standardized using StandardScaler to ensure fair distance calculations.  
  
4. Clustering:  
 - KMeans with 3 clusters is applied to segment users.  
  
5. PCA for Visualization:  
 - PCA reduces the feature dimensions to 2D for plotting.  
  
6. Visualization:  
 - Clusters are plotted using a scatter plot with color-coded labels.  
  
7. Export:  
 - Final results with cluster labels are exported to a new CSV file.

## Code

import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
from sklearn.preprocessing import StandardScaler  
from sklearn.cluster import KMeans  
from sklearn.decomposition import PCA  
  
# Upload the file  
from google.colab import files  
uploaded = files.upload()  
  
# Load dataset  
df = pd.read\_csv(next(iter(uploaded)))  
  
# Encode categorical column  
df\_encoded = pd.get\_dummies(df, columns=['genre\_preference'])  
  
# Standardize features  
scaler = StandardScaler()  
scaled\_data = scaler.fit\_transform(df\_encoded)  
  
# Apply KMeans  
kmeans = KMeans(n\_clusters=3, random\_state=42)  
clusters = kmeans.fit\_predict(scaled\_data)  
  
# Add cluster to DataFrame  
df['Cluster'] = clusters  
  
# Reduce dimensions  
pca = PCA(n\_components=2)  
reduced\_data = pca.fit\_transform(scaled\_data)  
  
# Plot clusters  
plt.figure(figsize=(8,6))  
sns.scatterplot(x=reduced\_data[:, 0], y=reduced\_data[:, 1], hue=clusters, palette='viridis')  
plt.title('User Segmentation via KMeans Clustering')  
plt.xlabel('PCA Component 1')  
plt.ylabel('PCA Component 2')  
plt.legend(title='Cluster')  
plt.show()  
  
# Save clustered data  
df.to\_csv("segmented\_users.csv", index=False)

## Output/Result

Below is the screenshot of the resulting cluster visualization:



## References/Credits

- scikit-learn documentation: https://scikit-learn.org/  
- Google Colab: https://colab.research.google.com/  
- Seaborn: https://seaborn.pydata.org/