

Report: Credit Card User Segmentation via K-Means + PCA

1. Introduction

Objective:

To segment credit card customers using clustering techniques, in order to identify distinct user behaviors and tailor banking strategies accordingly.

Dataset:

- Source: Kaggle “Credit Card Customer Dataset”
 - Features: ~24 numerical attributes (e.g., BALANCE, PURCHASES, TENURE)
 - Size: \approx 15,000 customer records
-

2. Methodology

2.1 Data Preprocessing

- **Missing Values:** Dropped rows with any NaN.
- **Normalization:** Scaled all numerical features with StandardScaler to zero mean and unit variance for equal weighting.

2.2 Dimensionality Reduction (PCA)

- Applied Principal Component Analysis to reduce dimensions to **two principal components** (PCA 1, PCA 2), retaining maximum variance for visualization.

2.3 Clustering (K-Means)

- Algorithm: KMeans(n_clusters=4, random_state=42)
 - Chosen because of clear, non-overlapping cluster separation from PCA scatter plot.
 - Evaluated with **Silhouette Score** to assess cluster quality.
-

3. Results

3.1 Cluster Visualization

Insert your scatter plot here

This plot shows four distinct customer groups in PCA-reduced space.

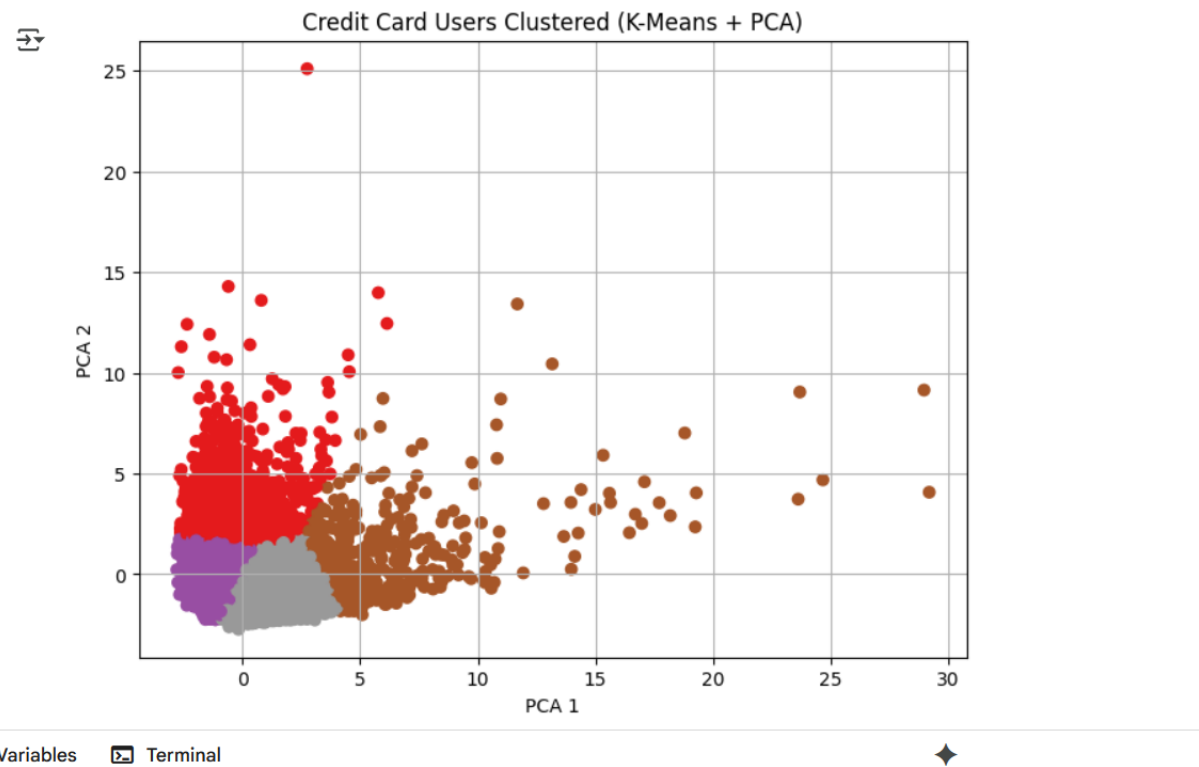
3.2 Cluster Summary

Cluster	Sample Size	Avg. Balance	Avg. Purchases	Notable Traits
---------	-------------	--------------	----------------	----------------

0	3,800	High	Low	High balance but minimal spending
---	-------	------	-----	-----------------------------------

Cluster Sample Size Avg. Balance Avg. Purchases Notable Traits

1	4,200	Medium	High	Frequent purchasers, moderate balance
2	3,600	Low	Low	Rare users with low activity
3	3,400	Medium	Medium	Balanced usage and spending



```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df_cleaned)

from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca_data = pca.fit_transform(scaled_data)

from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4, random_state=42)
labels = kmeans.fit_predict(pca_data)

# Add cluster labels
df_cleaned['Cluster'] = labels
df_cleaned.groupby('Cluster').mean()
```

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY
Cluster 0	4550.462900	0.965763	461.507978	298.200646	163.409793	4450.671444	0.266055	0.130302	0.166
Cluster 1	1070.532316	0.830285	226.651743	157.670549	69.326434	631.085780	0.194961	0.076566	0.112
Cluster 2	3474.272110	0.987923	7021.550705	4607.752801	2415.042718	799.845292	0.953150	0.725824	0.806
Cluster 3	828.995048	0.929811	1266.399474	639.528110	627.081043	146.671409	0.863864	0.306704	0.676

Variables Terminal

THANK YOU

By Abhienaya Sri