Analyzing Credit Card Holder Behavior: Insights from Clustering Analysis

Mirko Bicchierai

University of Florence
Statistical Learning

June 25, 2024

Introduction

The dataset utilized captures the usage patterns of credit card holders over the past six months. The objective is to categorize these customers into distinct groups based on their behavior.

This study seeks to derive actionable marketing insights from the analysis by:

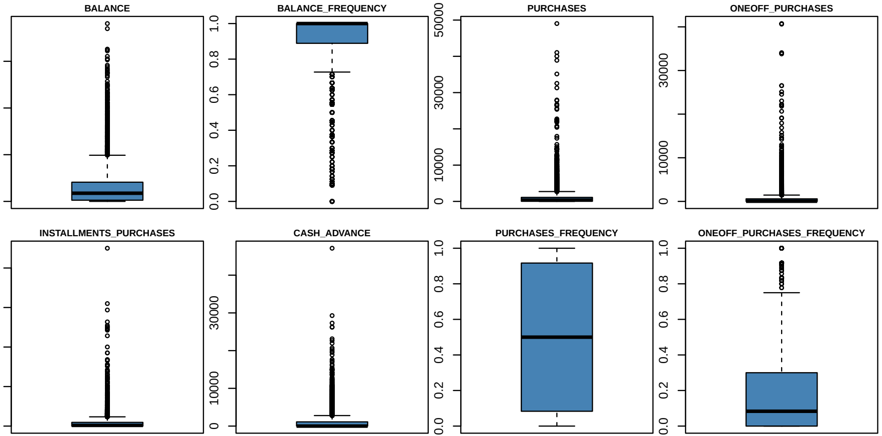
- Employing various **clustering models** to group customers.
- Conducting detailed interpretation and analysis of these customer segments (**profiling**).

The ultimate goal is to provide **strategic recommendations** to optimize credit card marketing efforts effectively.

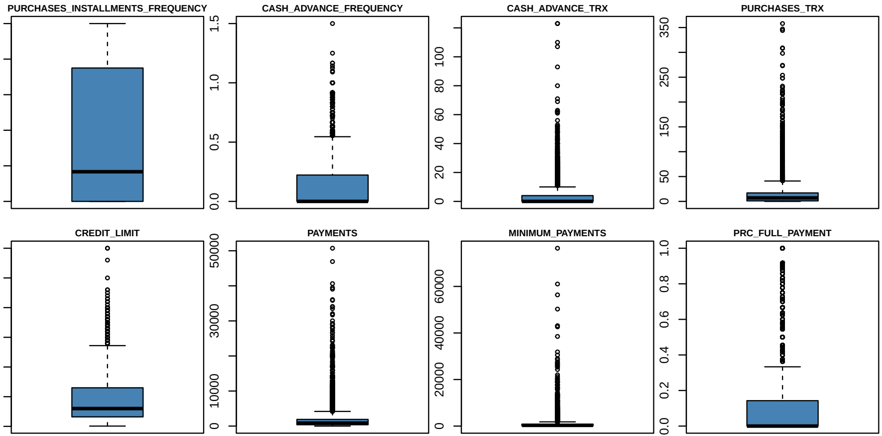
Dataset Description

- The dataset that will be used contains the usage behavior of around **9000** credit card users, **18** feature, for the last six months.
- Only one variable, 'CUST_ID,' has been removed

Data Visualization, box-plot (1/2)



Data Visualization, box-plot (2/2)



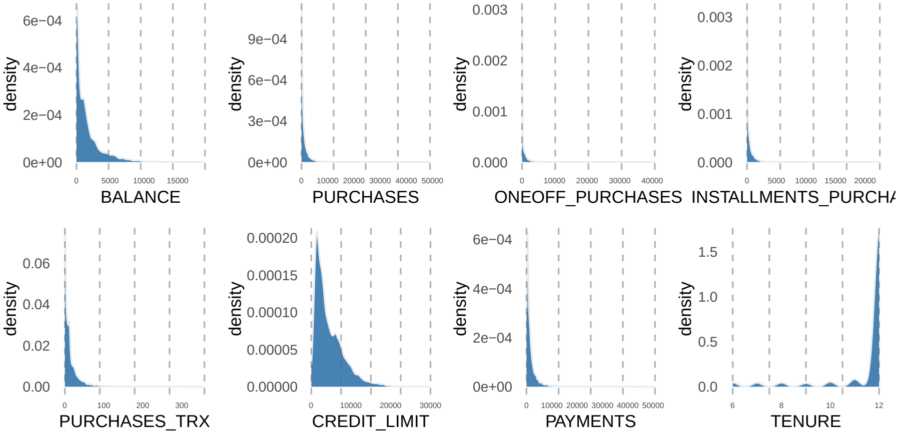
Outliers problem

- As we can see there are so **many outliers**
- If we were to remove all of them from the dataset, considering all variables, approximately half of the observations would be removed.
- So, only some extreme values in certain variables were removed (considered as noise, 49 observation)
- $BALANCE < 15000$, $PURCHASES < 40000$, $CASH\_ADVANCE < 40000$, $MINIMUM\_PAYMENTS < 60000$

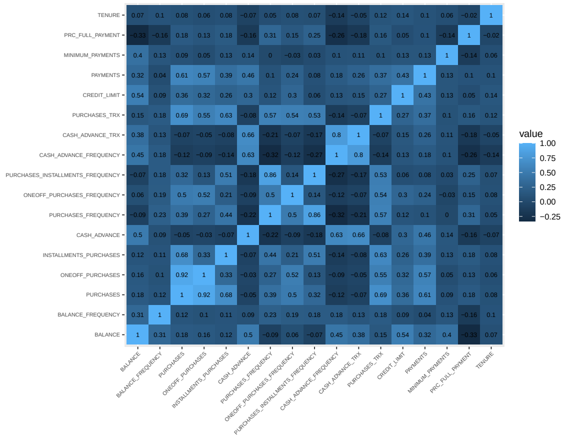
Variables imputation

- All observations where CREDIT_LIMIT was NA have been removed.
- For filling the missing values of MINIMUM_PAYMENTS:
  - If it has the value of PAYMENTS and is equal to zero, we consider the zero for it
  - If PAYMENTS is a value between 0 and PAYMENTS_MEAN, we use PAYMENTS
  - Otherwise, we use PAYMENTS_MEAN

Data Visualization, some density plot

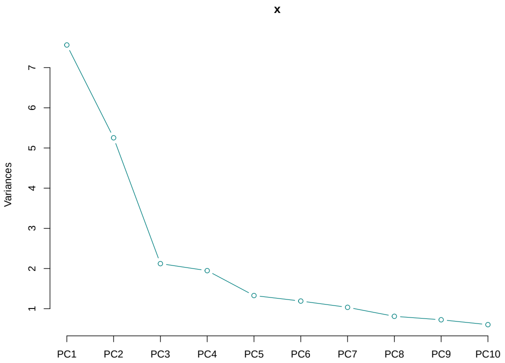


HeatMap



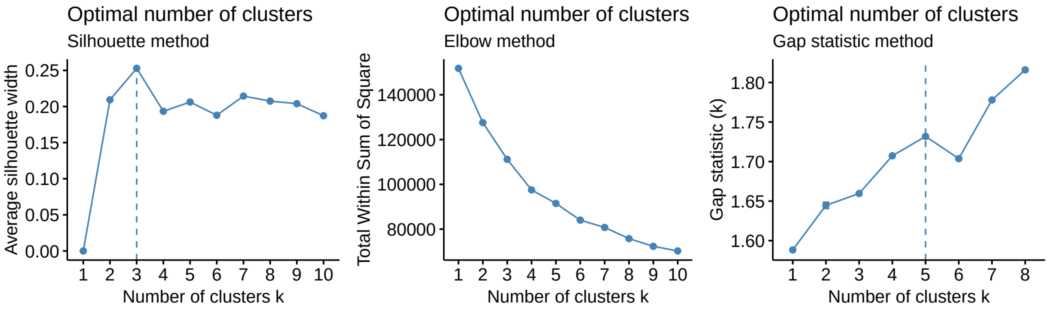
- Some variables have moderate to high correlation values to other variables (0.5 to 1)
- PURCHASES and ONEOFF_PURCHASES, with a 0.92 correlation value
- CASH_ADVANCE_TRX with CASH_ADVANCE_FREQUENCY with a 0.8 correlation value

Principal Component Analysis



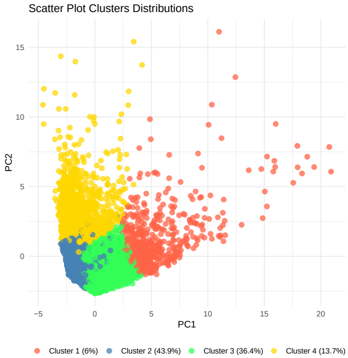
- In the presence of outliers, the estimates of the mean, variance and covariance can be seriously affected
- To solve the problem, a robust estimator of the variance-covariance matrix, specifically the Minimum Covariance Determinant (**MCD**) estimator, was used.

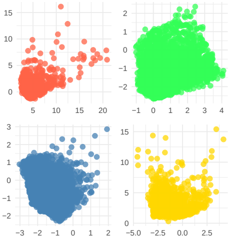
K-Means++: Determining the Number of Clusters

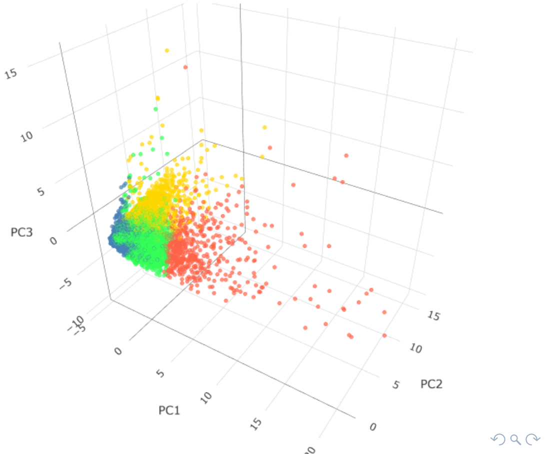


Based on these metrics, they suggest that the optimal number of clusters is between 3 and 5.

Visualization of clustering results using PCA



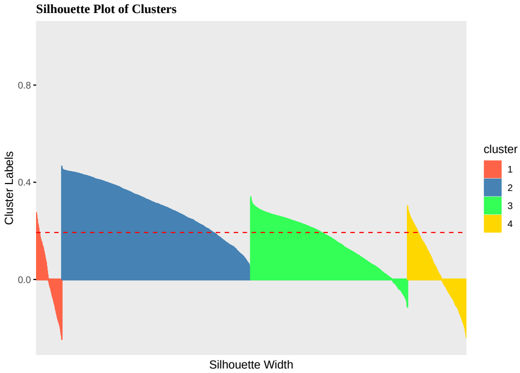




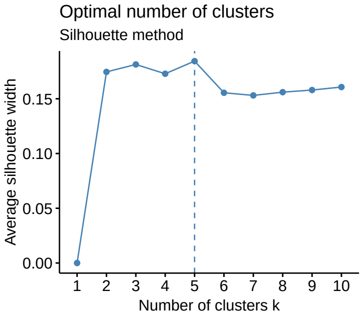
Score obtained

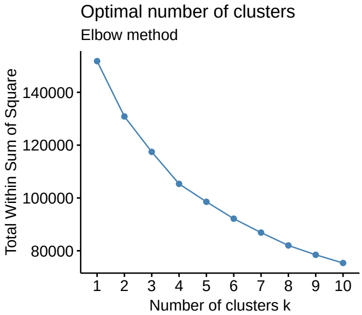
Score:

- Davies-Bouldin Index: 1.6
- Silhouette Score: 0.193
- Calinski Harabasz Index: 1657.17



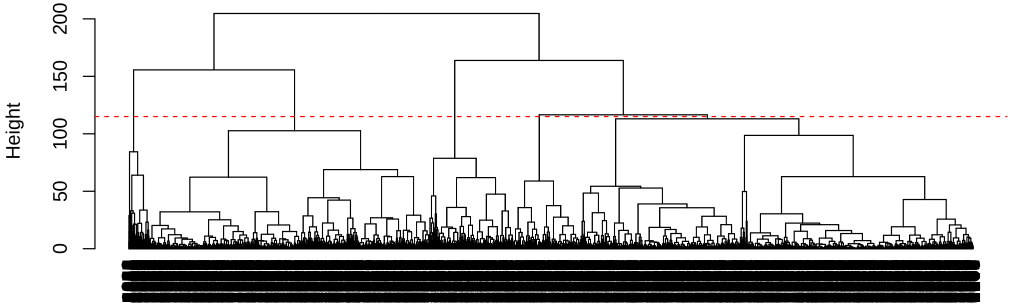
Determining the Number of Clusters



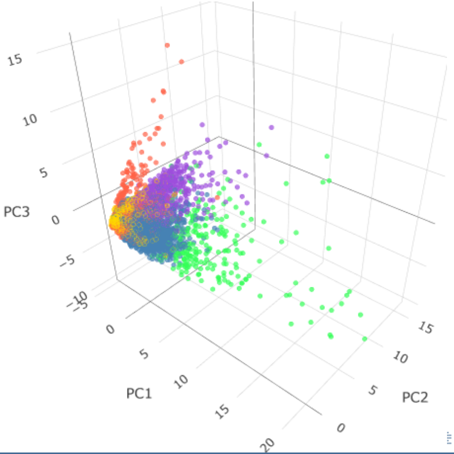


Based on these metrics, they suggest that the optimal number of clusters is 5.

Dendrogram



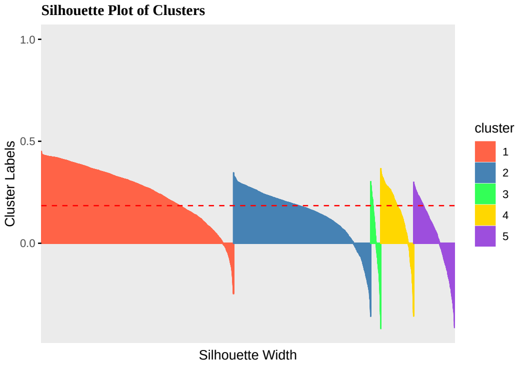
Visualization of clustering results using PCA



Score obtained

Score:

- Davies-Bouldin Index: 1.58
- Silhouette Score: 0.18
- Calinski Harabasz Index: 1206.12



What is Spectral Clustering?

- Spectral clustering is a technique that uses the spectral (**eigenvalues and eigenvectors**) properties of a matrix derived from the data to identify groups of similar points.
- The similarity matrix may be defined as a symmetric matrix W, where $W_{ij} \geq 0$ represents a measure of the similarity between data points with indices i and j.
- The general approach to spectral clustering is to use a **standard clustering method on relevant eigenvectors** of a Laplacian matrix of W

Benefits of Spectral Clustering

Advantages:

- Can handle non-linear cluster structures better than traditional methods like K-means.
- Particularly effective for datasets with complex structures.

Spectral clustering: Theoretical Introduction

Core points (naive):

- Constructing the **Similarity Matrix** W
  - Gaussian (Kernel) Similarity (RFB): $W_{ij} = \exp\left(-\sigma \|x_i - x_j\|^2\right)$
    If two points are close then $W_{i,j} \approx 1$, when two points are far $W_{i,j} \approx 0$
  - Euclidean distance: $W_{ij} = -\|x_i - x_j\|^2$
- Derive the **Laplacian Matrix**

  $L = D - W$ (unnormalized)
  $L = I - D^{-1/2} W D^{-1/2}$ (normalized), where D is the diagonal degree matrix.
- Computing the first k the eigenvectors of L

Benefits of normalizing L

- **Stability with respect to data scale**: The matrix $L = D - W$ directly depends on the absolute values of the similarities $W_{i,j}$. This means that variations in the data scale can significantly influence the structure of L.
- **Enhanced clustering performance**
- **Convergence of clustering algorithms**

Relationships with other clustering algorithms

The ideas behind spectral clustering may not be immediately obvious. Spectral clustering has strong relationships with other clustering algorithms, including:

- **Relationship with K-Means:** K-Means problem shares the objective function with the spectral clustering problem.
- **Relationship to DBSCAN:** In the trivial case of determining connected graph components, the optimal clusters with no edges cut, spectral clustering is also related to a spectral version of DBSCAN clustering that finds density-connected components.

Resume of spectral clustering algorithm

Algorithm steps:

1. Create a similarity matrix W
2. Calculate the Laplacian L (or the normalized Laplacian) for W
3. Calculate the first k eigenvectors (the eigenvectors corresponding to the k smallest eigenvalues of L)
4. Create a matrix V from the first k eigenvectors of L
5. Aplly KMeans clustering on V to obtain k cluster

Algorithm analisys

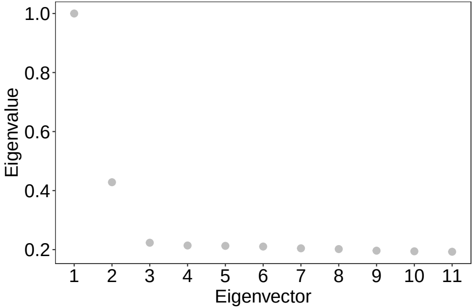
Time complexity depends on the sparsity of the data matrix (Improved by sparsification).

- Construction of the similarity matrix $\mathcal{O}(n^2)$
- Construction of the Laplacian matrix $\mathcal{O}(n^2)$
- Computing eigenvalues and eigenvectors $\mathcal{O}(n^3)$ (exact methods) or $\mathcal{O}(nk^2)$ / $\mathcal{O}(n^2k)$ (approximate methods, Lanczos algorithm)
- Clustering of eigenvectors $\mathcal{O}(nkt)$

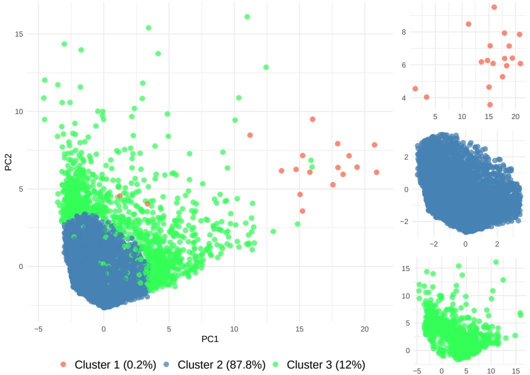
Select number of clusters

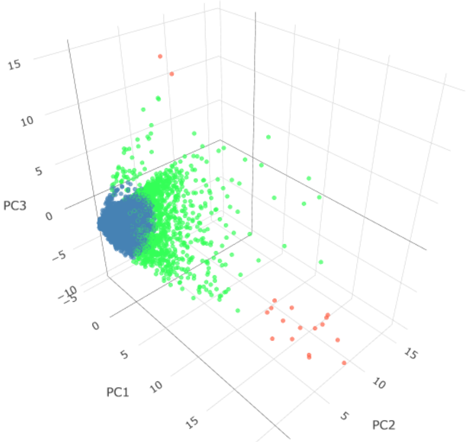
Methods:

- Look at the eigenvalues of the graph Laplacian and chose the K corresponding to the maximum drop-off (In the our case $k = 2/3$).
- Rotating the eigenvectors and minimising a cost function using gradient descent for each K. Then K is the one with the lowest cost.
- Standard method.



Visualization of clustering results using PCA

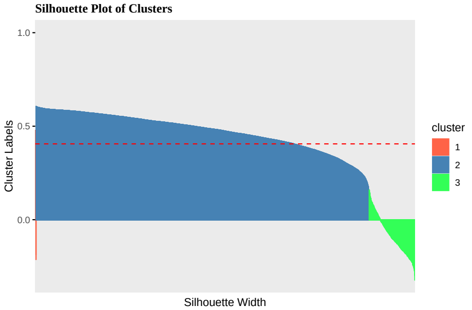




Score obtained

Score:

- Davies-Bouldin Index: 1.714453
- Silhouette Score: 0.4055048
- Calinski Harabasz Index: 867.6819

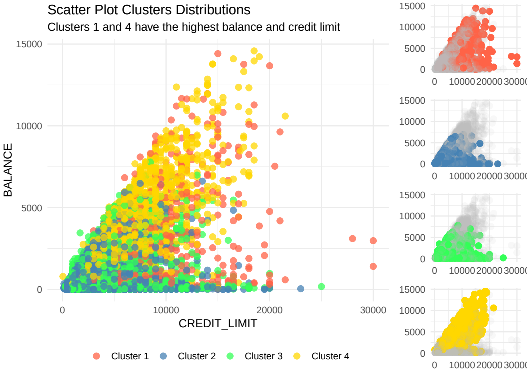


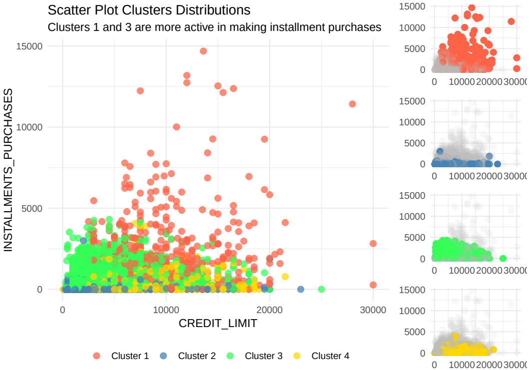
Model Comparison

	Davies-Bouldin Index	Silhouette Score	Calinski Harabasz Index
K-Means++	1.60	0.193	**1657.17**
Hierarchical	**1.58**	0.184	1206.12
Spectral	1.71	**0.405**	867.68

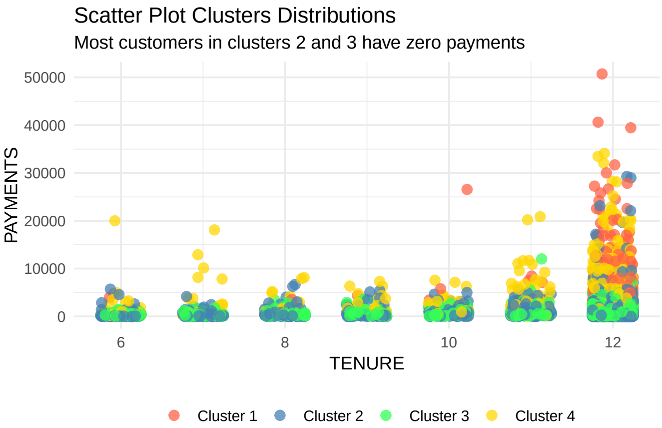
- In terms of this metrics used, the results obtained with the KMeans++ and spectral clustering algorithm are the best.
- The other methods also produced generally good results.

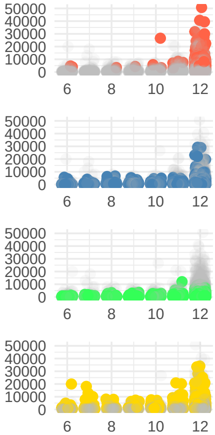
Some scatter Plot of K-Means++ (1/2)



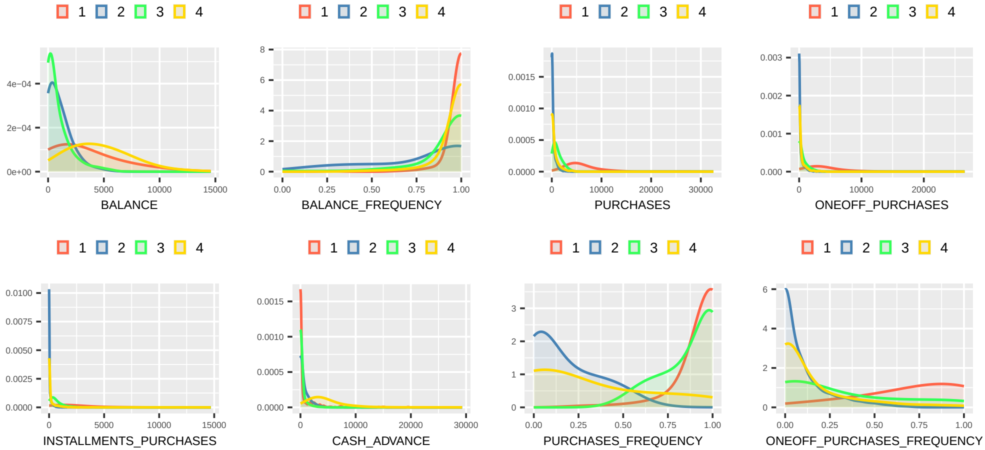


Some scatter Plot of K-Means++ (2/2)

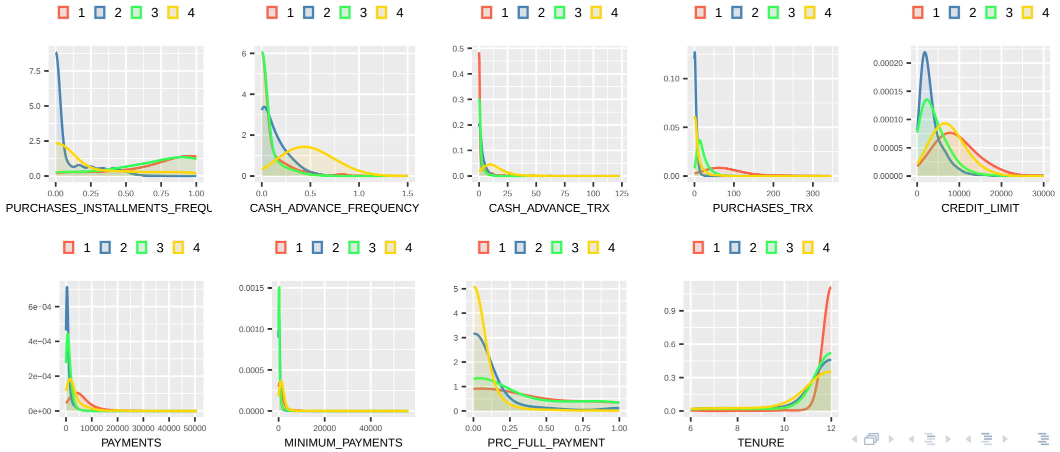




Density Plot of K-Means++ Clusters (1/2)



Density Plot of K-means++ Clusters (2/2)



Clusters profiling

- **Cluster 1 (Full Payers Users)**: Customers in this cluster are active users of the bank's credit card. This is evident from the frequently changing balance and the relatively high amount of balances compared to other clusters.
- **Cluster 2 (Starter/Student users)**: Customers in this cluster rarely or almost never use credit cards for transactions and installments. This is because these customers have relatively low balances, the balance changes infrequently, and the installment amounts are very low. Additionally, a low credit limit indicates that customers rarely or almost never use credit cards for processing credit transactions.

Clusters profiling

- **Cluster 3 (Installment Users)**: Customers use credit cards specifically for **installment purposes**. This is due to the relatively high level of transactions using installments in this cluster. Additionally, customers in this cluster often make transactions with very large amounts per transaction, and the frequency and transactions of cash advances are very low.
- **Cluster 4 (Cash Advance/Withdraw Users)**: Customers in this cluster have **high balances**, with balances changing frequently and a high frequency of cash advances. Additionally, customers in this cluster have the **lowest interest rates** compared to other clusters, the second-highest credit limit, and payments among the four clusters.

Marketing strategy suggestions

- Customers in Cluster 1 could become the primary target for credit card marketing. This is because customers in this cluster are highly active in using credit cards and have the highest seniority and credit limits compared to other clusters. By focusing marketing efforts on this cluster, banks can increase their profits through more frequent credit card usage and optimize marketing costs incurred.
- For Cluster 2, banks can offer special credit cards tailored for beginners or students (entry-level cards) who may not have an extensive credit profile.

Marketing strategy suggestions

- For credit cards specifically geared towards installment payments, banks can focus their marketing efforts on customers in Cluster 3. This is because customers in Cluster 3 are more likely to conduct credit card transactions for installment purposes. Banks can offer installment programs with low or 0% interest rates that can be used for various installment needs, aiming to attract customers in this cluster to use credit cards.
- Since customers in Cluster 4 tend to make cash advances, banks can offer special credit cards with various benefits. These benefits may include low or zero fees for cash advances or administrative fees, low interest rates, high seniority perks, etc.

Conclusions and possible new investigations

Conclusions

- All the tested algorithms achieved good results, however, upon deeper analysis, K-Means++ proves to be the best.
- There are also other algorithms, such as K-Median and DBSCAN, which exhibit better robustness to outliers or allow for outlier detection.

References

BHASIN, A. (2018).
Credit card dataset for clustering.
Kaggle Dataset, https://www.kaggle.com/datasets/arjunbhasin2013/ccdata.

Tibshirani, R., Walther, G., and Hastie, T. (2001).
Estimating the number of clusters in a data set via the gap statistic.
Journal of the Royal Statistical Society Series B, 63:411–423.

von Luxburg, U. (2007).
A tutorial on spectral clustering.