Project Report

USML - DS 5230

Targeted deals recommendation system using credit card usage.

By

Dinesh Sai Pappuru

Nikitha Kanaparthi

Introduction:

Credit cards are extensively utilized in the USA for a wide range of purchases, with credit card companies frequently providing recommendations on various deals. However, these suggested deals often fail to resonate with users' preferences. As credit card users ourselves, from the inception of our credit card usage, we have observed a consistent lack of relevance in the recommended deals. Our primary project objective is to collaborate with credit card companies to devise an improved recommendation system. This system aims to better align offers with individual customer preferences, fostering increased satisfaction. The goal is to encourage customers to use their credit cards in more diverse ways, benefiting both the company and customers.

Approach:

For our data science initiative, we have adhered to a typical project workflow. We acquired a credit card usage dataset from Kaggle and embarked on applying unsupervised machine learning techniques. Specifically, we employed algorithms such as K-Means clustering, DBSCAN, and Agglomerative clustering to determine the most suitable approach for our dataset. The objective was to identify the algorithm that would best align with our goal of creating optimal clusters.

After evaluating the results, we selected the most effective clustering algorithm. Subsequently, we utilized profiling techniques to gain insights into the performance of each cluster. This analysis allows us to understand how customers are using their credit cards. By examining the behavior of each cluster, we can recommend offers that are tailored to the specific criteria and preferences of each group of customers. This approach enables us to enhance the relevance of recommendations and, ultimately, improve customer satisfaction and engagement.

Dataset:

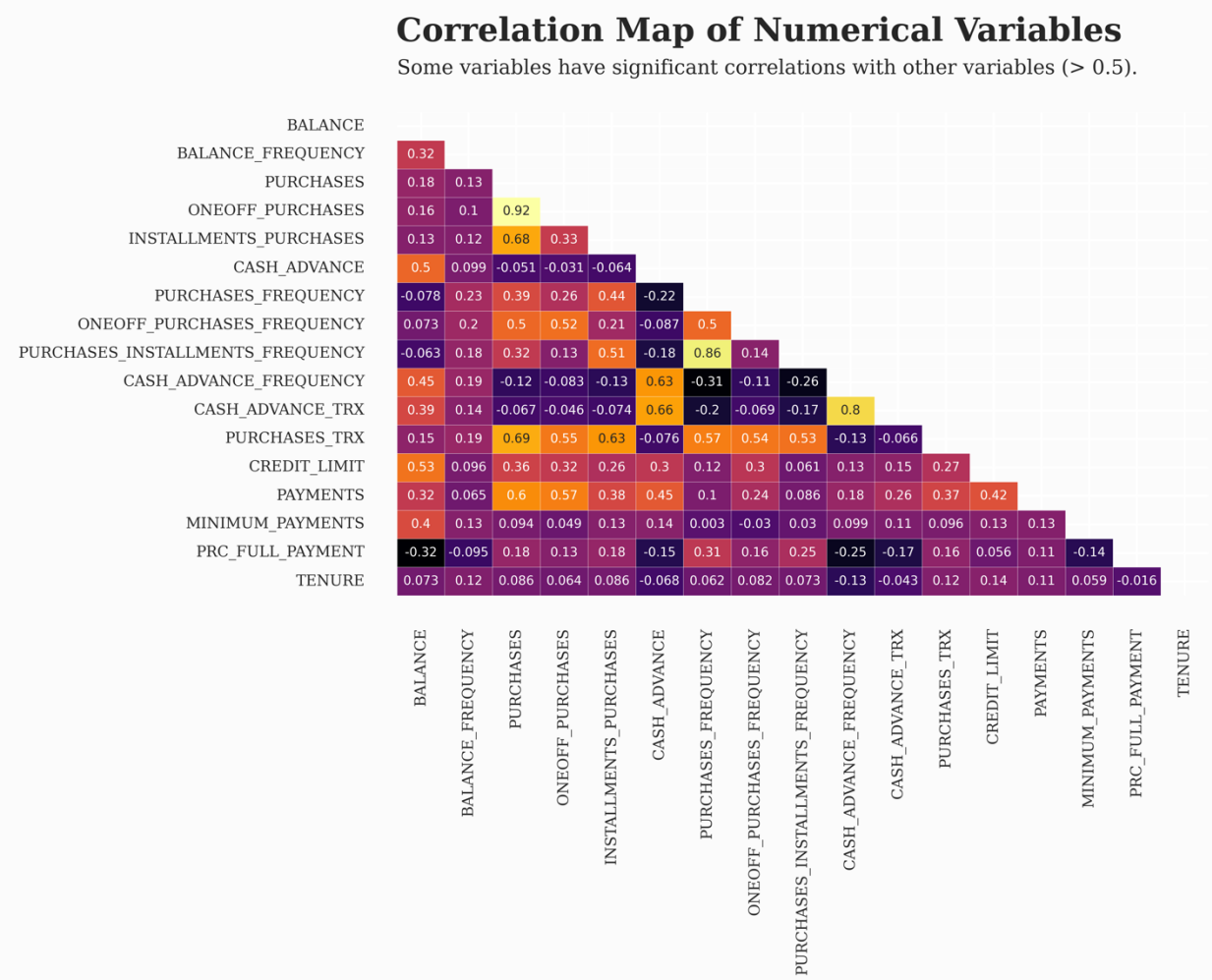
With a dataset encompassing the usage behavior of approximately 9,000 credit card users over the past six months, our primary objective is to employ clustering techniques for grouping customers based on their behavior. This segmentation process is crucial for developing a targeted and impactful credit card marketing strategy. By categorizing users into distinct groups, we aim to uncover patterns and trends in their credit card usage, enabling the design of personalized marketing initiatives that resonate with the specific needs and preferences of each customer cluster. This approach ensures that our marketing strategy is not only effective but also tailored to the diverse behaviors exhibited by different segments of credit card users.

The analysis of the dataset and correlation matrix yields several noteworthy observations about the credit card usage behavior of customers. Firstly, the dataset contains missing values, with one missing value in the CREDIT_LIMIT column and 313 missing values in the MINIMUM_PAYMENTS column.

Correlation analysis reveals interesting relationships between variables. Notably, there is a high correlation (0.92) between PURCHASES and ONEOFF_PURCHASES. Similarly, CASH_ADVANCE_TRX and CASH_ADVANCE_FREQUENCY exhibit a significant correlation value of 0.8.

Exploring the BALANCE column further, it becomes apparent that many credit cards exhibit zero balances. This aligns with the presence of numerous zero purchase amounts in the PURCHASE column. This pattern suggests that certain users intentionally maintain low balances to secure higher credit limits, influencing credit utilization ratios and credit scores.

Below is the correlation map of all the columns in the dataset.



Examining the frequency of credit card usage, most accounts score a 1 in the BALANCE_FREQUENCY column, indicating frequent credit card use. However, this differs from the patterns observed in ONEOFF_PURCHASES and PURCHASES_INSTALLMENT_FREQUENCY, where most customers do not engage in one-time transactions or installment payments.

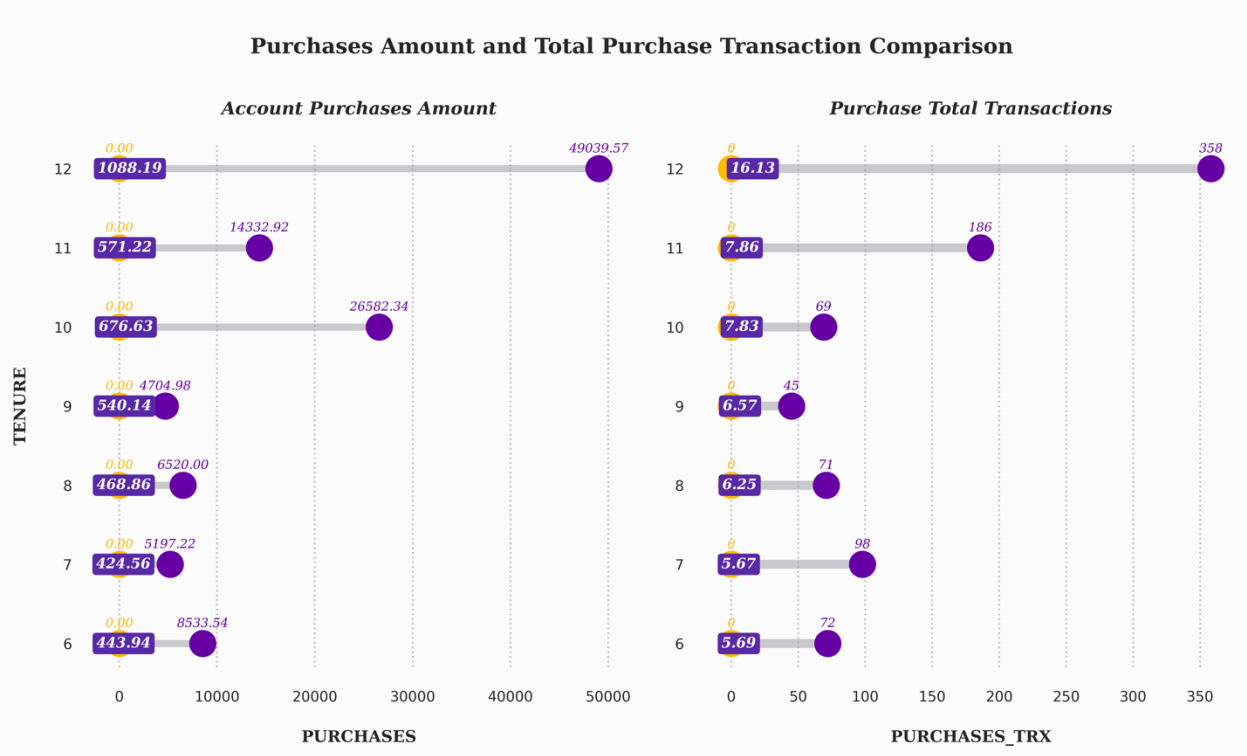
These insights provide a comprehensive understanding of customer behavior, offering valuable information for crafting targeted marketing strategies and potentially identifying areas for improvement or intervention in credit card offerings.

EDA (exploratory data analysis):

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.

EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate.

After performing EDA on our dataset, we found the following information useful.
Dumbbell chart:



The insights drawn from the dumbbell chart highlight a distinct pattern among credit card customers based on their tenure periods. Specifically, customers with a 12-month tenure exhibit a higher willingness to engage in purchase transactions and tend to have larger total purchase amounts compared to customers with other tenure periods.

Furthermore, the chart suggests that a subset of customers deliberately refrains from making any transactions (displaying 0 purchases and transactions). This strategic behavior is driven by the intent to secure a higher credit limit, thereby influencing credit utilization ratios and contributing to an increase in credit scores. This intentional approach to maintaining low transaction activity reflects a conscious effort by some customers to optimize their credit profile.

In essence, the tenure-specific trends depicted in the dumbbell chart provide valuable insights into the varying behaviors of credit card users. These findings can inform the development of targeted strategies, allowing credit card companies to better tailor their offerings to the preferences and habits of customers across different tenure periods.

Preprocessing:

The first stage is to **remove variables that are not needed** for the clustering process. In our dataset case, **CUST_ID will be removed** since it has unique values.

1. Imputation:
   Since the dataset is about clustering, imputation will use KNNImputer() to avoid biased clustering results. The mean value from the nearest n_neighbors found in the dataset is used to impute the missing values for each sample.

2. Scaling:
   The next step is scaling the dataset. Scaling is essential since it manages the dataset's variability, transforms data into a defined range using a linear transformation to produce high-quality clusters, and boosts the precision of clustering algorithms. In this case, a standard scaler used to standardize the feature by removing the mean and scaling to unit variance.

3. Hopkins test:
   The Hopkins statistic (introduced by Brian Hopkins and John Gordon Skellam) is a way of measuring the cluster tendency of a data set. It belongs to the family of sparse sampling tests. It acts as a statistical hypothesis test where the null hypothesis is that the data is generated by a Poisson point process and are thus uniformly randomly distributed. If individuals are aggregated, then its value approaches 0, and if they are randomly distributed, the value tends to 0.5.

   The following is the hypothesis of the Hopkins statistical test.
   H0: The dataset is not uniformly distributed (contains meaningful clusters).
   H1: The dataset is uniformly distributed (no meaningful clusters).
   Criteria:
   If the value is between {0.7, ..., 0.99}, accept H0 (it has a high tendency to cluster).

   Hopkins Test:

   Result: 0.9664
   From the result above, it has a high tendency to cluster (contains meaningful clusters)

   Conclusions: Accept H0

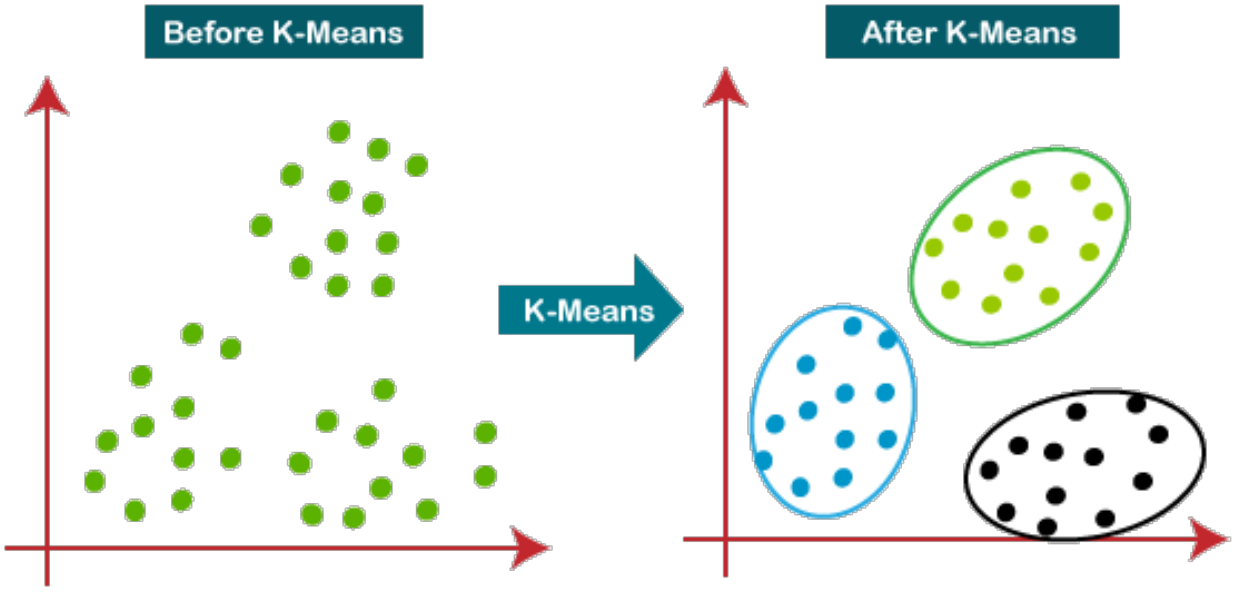
4. PCA:
   Principal component analysis (PCA) is a method used in unsupervised machine learning (such as clustering) that reduces high-dimension data to smaller dimensions while preserving as much information as possible.

   By using PCA before applying clustering algorithm, it allows to reduce dimensions, data noise, and decrease computation cost. In this notebook, the number of features will be reduced to 2 dimensions so that the clustering results can be visualized.

Clustering Models

1. K-Means Clustering:

K-Means Clustering is a popular unsupervised machine learning algorithm designed for the partitioning of data into distinct, non-overlapping groups or clusters. The algorithm's objective is to minimize the variance within each cluster while maximizing the separation between clusters. Operating iteratively, K-Means assigns data points to clusters based on the similarity of their features, with the number of clusters, 'k,' predetermined by the user. The algorithm employs a centroid-based approach, where each cluster is represented by a centroid, and data points are assigned to the cluster with the closest centroid. The process continues until convergence, with centroids recalculated in each iteration. K-Means is widely utilized for various applications, including customer segmentation, image compression, and anomaly detection, owing to its simplicity, scalability, and effectiveness in identifying inherent patterns within datasets.



Before implementing K-Means, the first step is to calculate the optimal number of clusters using the elbow score. Besides that, the Calinski-Harabasz index will be utilized to determine the ideal number of clusters.

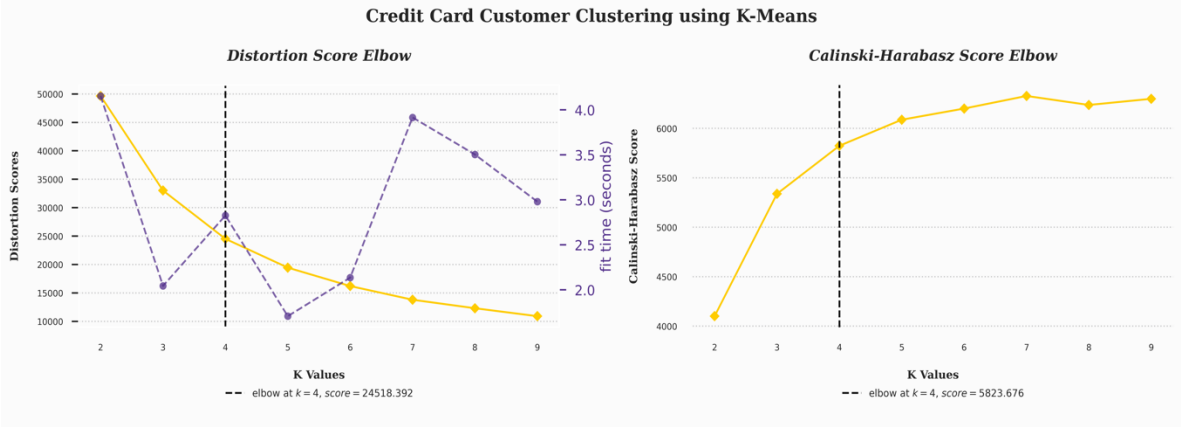
Distortion Score Elbow:

The elbow method simply entails looking at a line graph that (hopefully) shows as more centroids are added the breadth of data around those centroids decreases. In this case, the breadth of data is called distortion or sum of square errors (SSE). Distortion could decrease rapidly at first then slowly flatten forming an “elbow” in a line graph.

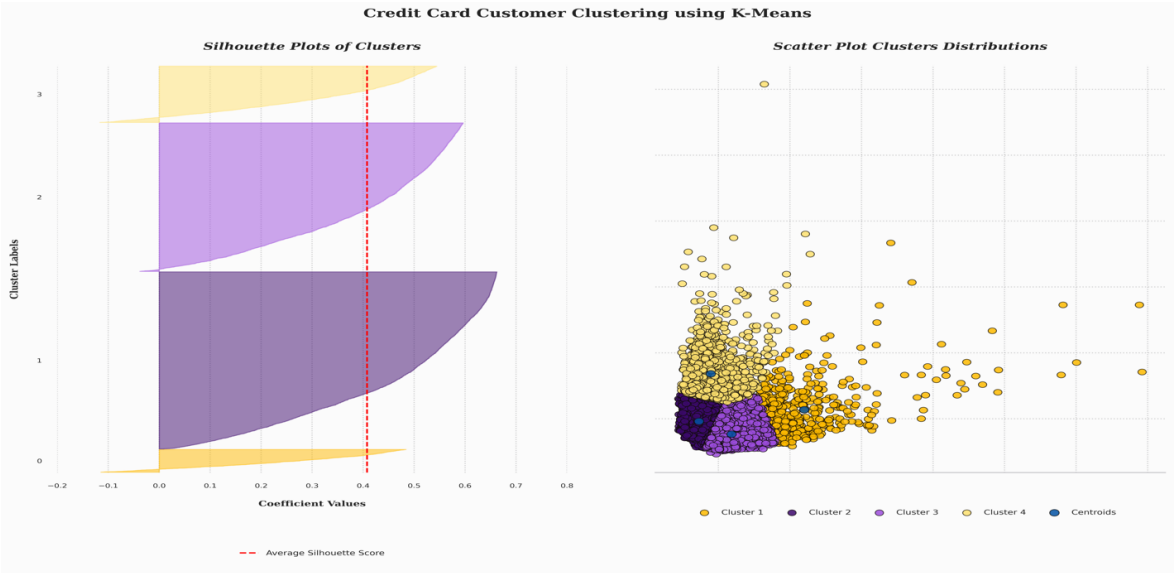
Calinski-Harabasz Score Elbow:

The elbow method involves running the K-Means algorithm for a range of values of k and calculating the Calinski-Harabasz (CH) score for each. The CH scores are then plotted against the number of clusters, forming a curve. The "elbow" of the curve represents a point where adding more clusters does

not significantly increase the CH score. In other words, the optimal number of clusters is often identified at the point where the CH score starts to diminish, creating an elbow-like bend in the plot.



Based on the results of the elbow method and Calinski Harabasz score above, it can be concluded that the best clustering number for the K-Means algorithm is 4 clusters. The following steps will apply the number of optimal clusters, visualize clusters distribution plot, and silhouette plots to evaluate their performance.



Clustering Evaluation:

Davis-Bouldin Index is a metric for evaluating clustering algorithms. It is defined as a ratio between the cluster scatter and the cluster's separation. Scores range from 0 and up. 0 indicates better clustering.

Silhouette Coefficient/Score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1. The higher the score, the better. 1 means clusters are well apart from each

other and clearly distinguished. 0 means clusters are indifferent/the distance between clusters is not significant. -1 means clusters are assigned in the wrong way.

Calinski-Harabasz Index (also known as the Variance Ratio Criterion), is the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters, the higher the score, the better the performances.

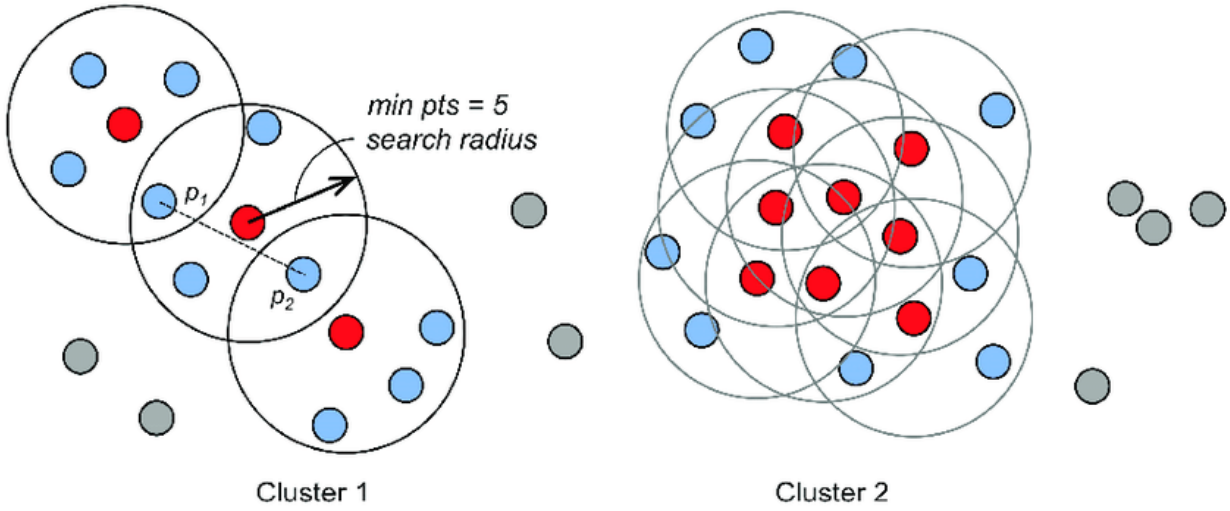
Results from Evaluation:

Davies-Bouldin Index: 0.801
Silhouette Score: 0.408
Calinski Harabasz Index: 5823.676

Based on the evaluation score above, the clustering quality using K-Means with 4 clusters is decent. T his is due to overlapping between clusters, as shown in the scatter plot.

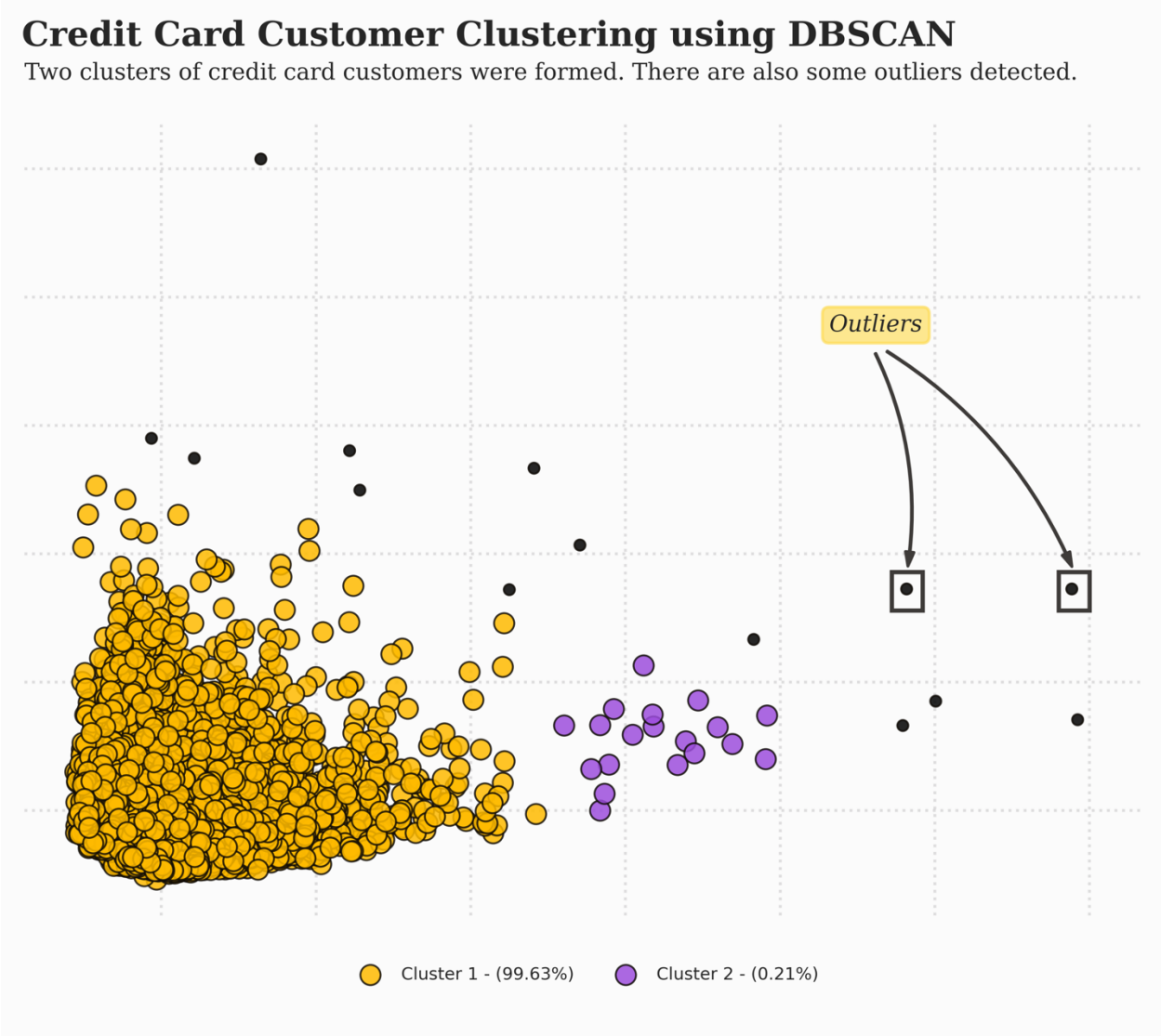
2. DBSCAN:

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) groups points based on the lowest number of points and the Euclidean distance. It also marks as outliers the points that are in low-density regions. The two DBSCAN parameters are MinPoints and Epsilon.



DBSCAN implementation:

From DBSCAN implementation, there are 2 clusters formed. Cluster 1 has the most data points compared to cluster 2. However, there are some outliers detected since some points are too far from the other data points (DBSCAN considered it as an outlier and assigned -1 label to those points). The following step is to assess the clustering quality that DBSCAN provides.



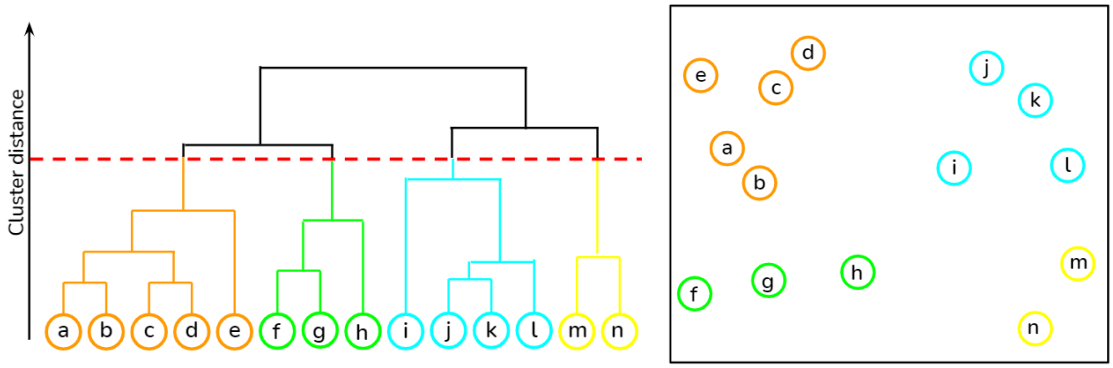
Results from Evaluation:

Davies-Bouldin Index: 1.287
Silhouette Score: 0.803
Calinski Harabasz Index: 685.303

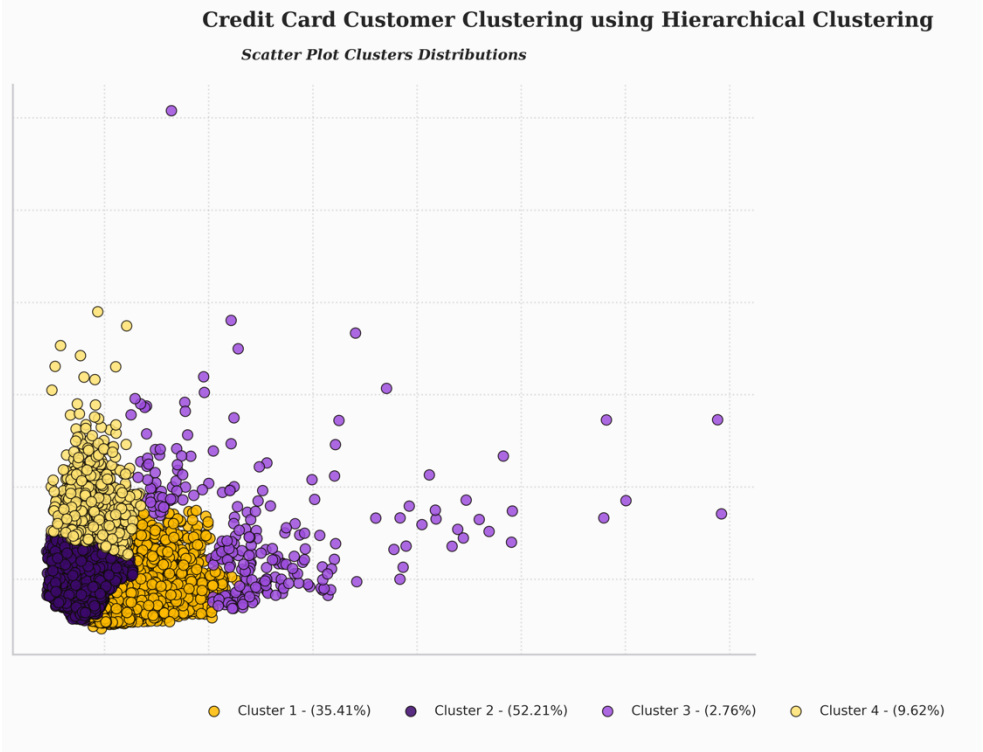
The clustering quality using DBSCAN with two clusters and outliers is fair according to the evaluatio n score above. The silhouette score is better than K-Means since there are one large cluster and one sm all cluster formed, although the Davies-Bouldin index is higher than K-Means, which indicates fair cl ustering. However, the Calinski-Harabasz index obtained is much lower than K-Means.

3. Hierarchical Clustering (Agglomerative):

Hierarchical clustering works via grouping data into a tree of clusters. Hierarchical clustering begins by treating every data point as a separate cluster. Then, it repeatedly identify the 2 clusters which can be closest together, and merge the 2 maximum comparable clusters until all the clusters are merged together. In hierarchical clustering, the objective is to produce a hierarchical series of nested clusters. Dendrograms will be used to visualize the history of groupings and figure out the optimal number of clusters. Then using generated dendograms, we determine the largest vertical distance that doesn't intersect any of the other clusters. Afther that, draw a threshold/horizontal line at both extremities. Finally, the optimal number of clusters is equal to the number of vertical lines going through the horizontal line. For eg., in the below case, best choice for no. of clusters will be 4.



Hierarchical Clustering implementation:



Results from Evaluation:

Davies-Bouldin Index: 0.863
Silhouette Score: 0.388
Calinski Harabasz Index: 4797.51

Based on the results of evaluating the quality of clustering using hierarchical clustering, it can be seen that the results obtained are slightly different from K-Means. By using hierarchical clustering, the silhouette score obtained is close to 0, indicating overlapping clusters. In addition, a high Davies-Bouldin index indicates decent clustering quality. Compared to K-Means, the silhouette score for hierarchical clustering is 0.06 higher. And for the Davies-Bouldin index, the results obtained are 0.02 lower. The Calinski-Harabasz index obtained is slightly lower compared to K-Means, but higher compared to DBSCAN.

Model Accuracy Comparison:

Model	Davies-Bouldin Index	Silhouette Score	Calinski-Harabasz Index
K-Means	0.801000	0.408000	5823.676000
Hierarchical Clustering	0.863000	0.388000	4797.510000
DBSCAN	1.287000	0.803000	685.303000

The table above shows that the K-Means algorithm has the lowest Davies-Bouldin index compared to the other two algorithms, so it can be concluded that K-Means has the decent clustering quality compared to the other two algorithms. However, by silhouette score, K-Means has the second highest silhouette score.

Furthermore, clustering using the hierarchical clustering algorithm has similar clustering quality results as K-Means. The Davies-Bouldin index is slightly higher, and the silhouette score is slightly lower than K-Means. Finally, clustering using DBSCAN shows has the worst Davies-Bouldin index but the best silhouette score compared to other algorithms.

From the results of the Calinski-Harabasz index, it can be seen that K-Means has the highest index compared to other algorithms. This indicates that K-Means performs better and is dense than other algorithms.

It can be seen that K-Means has the best clustering quality of the three algorithms due to the lowest Davies-Bouldin index value and slightly better overlapping clusters than hierarchical clustering.

Conclusion:

Summary of Each Clusters:

Column Name	Metrics	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Overall
BALANCE	mean	3401.840056	1012.915503	824.180354	4510.612798	1564.474828
BALANCE_FREQUENCY	mean	0.988427	0.799792	0.919997	0.963486	0.877271
PURCHASES	mean	6894.613917	223.081279	1236.499406	462.647357	1003.204834
ONEOFF_PURCHASES	mean	4511.889901	157.576608	621.738764	300.127390	592.437371
INSTALLMENTS_PURCHASES	mean	2383.916859	65.833785	614.965725	162.619301	411.067645
CASH_ADVANCE	mean	773.154467	614.588758	147.442197	4401.478579	978.871112
PURCHASES_FREQUENCY	mean	0.954443	0.190099	0.862440	0.266258	0.490351
ONEOFF_PURCHASES_FREQUENCY	mean	0.726667	0.074901	0.301883	0.129829	0.202458
PURCHASES_INSTALLMENTS_FREQUENCY	mean	0.808946	0.109962	0.675116	0.166187	0.364437
CASH_ADVANCE_FREQUENCY	mean	0.084806	0.122459	0.030698	0.470904	0.135144
CASH_ADVANCE_TRX	mean	2.363817	2.264421	0.569732	13.771084	3.248827
PURCHASES_TRX	mean	82.902584	2.960949	21.317296	6.703614	14.709832
CREDIT_LIMIT	mean	9541.650099	3109.010550	4250.051843	7458.798832	4494.293646
PAYMENTS	mean	6723.271522	856.400147	1328.949020	3542.319312	1733.143852
MINIMUM_PAYMENTS	mean	1830.297811	589.839851	600.410547	2065.568910	868.716633
PRC_FULL_PAYMENT	mean	0.288014	0.065965	0.282861	0.034947	0.153715
TENURE	mean	11.960239	11.364216	11.661693	11.439357	11.517318

Based on the table above, it can be concluded that each cluster has the following characteristics:

Cluster 1 (Full Payers Users):
Customers in this cluster are active users of the bank's credit card. This can be seen from the frequency of the balance which frequently changes and the balances amount is high enough compared to other clusters. In addition, when compared to other clusters, this cluster has higher mean value in several aspects than other clusters. Credit card customers in this cluster also actively use credit cards to facilitate transactions and installments. Cash advances, transactions, and installments in this cluster also occur more frequently. The relatively high tenure also shows that the credit scoring in this cluster is very good.

Cluster 2 (Starter/Student users):
In contrast to cluster 1, customers rarely/almost never use credit cards for transactions and installments in this cluster. This is because the customer has a relatively small balance, the frequency of the balance rarely changes, and the installments are very low. In addition, a low credit limit also shows that customers rarely/almost never use credit cards to process credit transactions, and customers in this cluster also rarely make cash advances. So, it can be assumed that customers use credit cards for cash advance processes only with sufficient frequency. In addition, the low balance allows customers in this cluster to be students or new users who use credit cards at this bank.

Cluster 3 (Installment Users):
In this cluster, customers use credit cards specifically for installment purposes. This is due to the relatively high level of transactions using installments in this cluster. Moreover, customers in this cluster often make transactions with very large amounts per transaction and the frequency and transactions of cash in advance are very small. Customers in this cluster very rarely make payments and cash in advance and have a relatively small cash-in-advance frequency and amount of payments. It can be concluded that the customers in this cluster are very suitable for credit cards specifically for installment needs.

Cluster 4 (Cash Advance/Withdraw Users):
Customers in this cluster have high balances, the balances frequency are always changing, and the frequency of cash in advance and cash in advance is high. In addition, customers in this cluster have the lowest interest rates compared to other clusters and have the second highest credit limit and payments out of the four clusters. However, credit card users in this cluster rarely make installments or one-off purchases and have the third-highest tenure of the four clusters. Thus, it can be concluded that customers in this cluster only use credit cards for the need to withdraw money or cash advances.

In conclusion, the clustering analysis has successfully segmented the credit card dataset into four distinct clusters, each representing a unique category of credit card users. This segmentation allows for a more nuanced understanding of customer behavior and preferences. With this knowledge, targeted deals or recommendations can be tailored to each specific cluster, ensuring a more personalized approach in marketing strategies. By aligning promotions with the distinct needs and habits of each cluster, credit card companies can enhance customer satisfaction, engagement, and overall experience. This targeted approach is poised to be more effective in meeting the diverse expectations of different customer segments, ultimately contributing to a more successful and customer-centric marketing strategy.

References:

EDA:
https://www.ibm.com/topics/exploratory-data-analysis

Hopkins test:
https://en.wikipedia.org/wiki/Hopkins_statistic

PCA:
https://builtin.com/data-science/step-step-explanation-principal-component-analysis

K- Means:
https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

Elbow Method:
https://avidml.wordpress.com/2016/10/29/easily-understand-k-means-clustering/

Davis-Bouldin Index:
https://pyshark.com/davies-bouldin-index-for-k-means-clustering-evaluation-in-python/

Silhouette Analysis:
https://www.analyticsvidhya.com/blog/2021/05/k-mean-getting-the-optimal-number-of-clusters/

Calinski-Harabasz index:
https://pyshark.com/calinski-harabasz-index-for-k-means-clustering-evaluation-using-python/

DBSCAN:
https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html

Agglomerative Clustering:
https://www.statisticshowto.com/agglomerative-clustering/