Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.
Digital Object Identifier XX.XXXX/YYY.20XX.ZZZZZZZ

Unveiling Customer Segments in Credit Card Data

R HEMANTH REDDY **f20200905@hyderabad.bits-pilani.ac.in**
KULKARNI SREYAS **f20200076@hyderabad.bits-pilani.ac.in**
MITTAPALLI SREETHEERDHA **f20201889@hyderabad.bits-pilani.ac.in**

BITS Pilani Hyderabad Campus
CS F415 Data Mining Project

This work was completed as part of the Data Mining (CS F415) course project under the supervision of Dr. Aruna Malapati.

ABSTRACT Customer segmentation is critical for credit card companies to develop targeted marketing strategies. This study examines the use of clustering algorithms to group customers based on their credit card transaction histories. With a dataset covering the transaction behaviors of 8950 active credit card holders over the previous six months, the primary objective is to accurately segment customers by applying clustering techniques. This research holds immense potential to influence the competitiveness of credit cards in the market. Customer segmentation helps in effective customization of companies marketing strategies to specific customer segments. This improves the customer satisfaction and fostering loyalty. In this paper we talk about three clustering algorithms: K-means, Hierarchical clustering and DBSCAN. The results of these algorithms provide distinct customer segments characterized by transactional attributes. The insights from the experiments provide companies with guidance to refine their strategies and improve their product and service offerings. Companies can better cater to their clientele's diverse needs and preferences by understanding the various segments of their customer base, thus strengthening customer engagement and enhancing profitability. By comprehending the various segments of their customer base, companies can adeptly cater to their clientele's diverse needs and preferences, thereby fortifying customer engagement and bolstering profitability.

INDEX TERMS K-Means, Hierarchical clustering, DBSCAN, Agglomerative hierarchical clustering, Principal Component Analysis(PCA), Silhouette score, customer segmentation

I. INTRODUCTION

The data available at human's dispose is vastly abundant because of the billions of transactions and social media interactions happening every day. This data holds valuable information that can be extracted and used to generate actionable insights. The data we get from the financial sector is of utmost importance and data analytics there plays a major role in understanding consumer behavior and preferences. In this study, we aim to identify customer segments within credit card data using sophisticated data mining techniques to derive actionable insights which would assist the financial institutions in improving their products and services they offer.

A. PROBLEM AND OBJECTIVE

The primary objective of this research is to present customer segments within credit card data by studying and identifying the patterns and trends to differentiate the characteristics of groups of credit card users. The proliferation of use of credit card for daily transactions has led to an exponential increase in the transactional data, presenting both opportunities and challenges for financial institutions. While abundant data offers unparalleled insights into consumer behavior, its sheer volume and complexity pose significant hurdles in extracting meaningful information. This study solves the problems by clustering the user groups based on their spending behavior, the type of transactions(one off, instalments) and other pertinent factors.

B. INTEREST AND IMPORTANCE

Researching customer segmentation in credit cards is not only an academic study but also has important consequences for financial markets. Understanding the unique preferences, behaviors and needs of different customers allows financial institutions to tailor their products and services accordingly, ensuring users' product satisfaction and trust. Additionally, in an increasingly competitive environment where customer retention is important, a personal marketing strategy can be a powerful tool for differentiation. Using information from customers, financial institutions can create marketing plans for specific customers, thereby increasing engagement and improving the brand affinity.

The significance of this study is that, unlike traditional kinds of data such as demographic information or survey responses, the difficulties are exacerbated by the inherent complexities and nuances encoded in credit card data. Credit card data encompasses many transactional details, reflecting the intricacies of consumer spending patterns in real time. The greater challenge we face is to find the patterns amidst the noise, outliers and high dimensionality due to various factors which influence decision making for the customers. Therefore, the ability to effectively segment customers within this data realm holds immense promise in unlocking hidden opportunities and driving business growth.

C. DIFFICULTY AND CHALLENGES

Customer segmentation is hard to implement in real life scenarios because of the sheer volume and velocity of data that gets flooded in the database every single day. Terabytes of data gets stored daily as credit card has become preferred mode of payment for many. The harder task is that it requires robust computational structure and sophisticated algorithms capable of handling scale and complexity of the data.

Moreover, the diversity among credit card consumers and their expending habits introduces an additional level of intricacy in the segmentation procedure. In contrast to homogenous datasets where trends can be easily identified, evidence from credit card transactions frequently presents a variety of overlapping attributes among distinct customer groups. Naive techniques for segmentation, such as manually grouping individuals by general demographic categories, must account for subtle differentiations between segments to achieve optimal results.

Furthermore, the ever-changing nuances of consumer behavior serve as a constantly-shifting goal for segmentation endeavors, as trends and preferences shift. Conventional models for segmentation, reliant on lifeless data stills, must adjust to these temporal fluctuations in order to avoid becoming obsolete and inadequate. Therefore, the underlying obstacle is rooted in recognizing customer divisions and maintaining their pertinence and precision amidst fluid marketplace forces.

D. PREVIOUS SOLUTIONS AND LIMITATIONS

Previous efforts to tackle the issue of customer segmentation in credit card data have primarily depended on traditional techniques based on statistical analysis and heuristic methodologies. These methods typically entail manually dividing customers according to predetermined factors such as age, income, or spending habits. Though these were effective to some extent they had to be reevaluated so as to check whether it had practical usefulness.

Manual methods of segmentation require enhanced scalability and effectiveness, particularly when handling vast amounts of data consisting of millions of transactions and thousands of customers. As the dataset grows in size, it becomes increasingly difficult to enforce and define segmentation criteria. As a result, these technologies become impractical for larger-scale applications. Furthermore, traditional techniques must account for the nuances of consumer behaviour, which frequently shows as cryptic arrangements and linkages. As a result, it is critical to implement more precise partitioning algorithms capable of dealing with the intricacies and diversity inherent in credit card data.

Unsupervised learning, unlike supervised techniques, seeks to extract useful insights from data without a predetermined purpose. It gives equal attention to all aspects, emphasising the need of summarising data in an original manner. Customer segmentation involves grouping clients based on shared characteristics such as demographics, preferences, and spending behaviors. This procedure aids enterprises in understanding their clientele, identifying particular demographics, and crafting tailored promotional strategies. Clustering methods are commonly utilized to discover customer segments, grouping data points with analogous characteristics. Despite the potentially vast quantity of customer data, it usually falls into a small number of distinct segments with shared traits within each segment, but discrepancies between them. Cluster analysis is a type of categorization that runs autonomously and discovers links between data points by studying similarities among observations while discarding labels.methodologies based on statistical analysis and heuristic approaches. These methods usually include manual segmentation based on specified parameters like age, income, or expenditure categories.

II. RELATED WORK

Regarding the study paper of Azad Abdulhafedh(2021) [1], it is discussing clustering based data mining techniques on credit card data for a company that aims to enhance their marketing strategy through customer segmentation. Customer segmentation is utilized to divide customers into groups by identifying common characters, with an intention of discovering different customer groups available. The study applies two methods for customer segmentation: Hierarchical clustering and the K-means, and also conducting dimensionality reduction via Principal Component Analysis (PCA) on dataset. The K-means clustering algorithm appears to be more fitting for customer segmentation than Hierarchical clustering in this dataset. The results illustrate that data dimensions were lessened from 17 to 5. Clustering analysis should incorporate data normalization when there are various units used to measure features in the dataset. In addition to exploring all data insights, it is wise to try multiple different clustering algorithms. This is because different properties of the data are likely best exploited through fitting with various types of clusters. For example, when we used the data for this project, K-means clustering seemed to indicate the best fit for this particular set of input information. More exploration into the utilization of PCA with K-means and Hierarchical clustering could find relevance in this additional work.

The paper by S.Jessica Saritha(2010) [2] idea is to combine the K-means clustering algorithm with Bayesian classification for credit card data analysis, in order to increase credit card analysis accuracy. This new method tries to improve on the limitations of basic Bayesian classification models by using a hybrid model that combines both methods together. This way, it offers better function and brevity than just using simple Bayesian classification alone. Nevertheless, even though the method demonstrates potential to overcome issues in credit card analysis like enhancing precision and productivity, it still sparks queries regarding its supremacy over other classification techniques. Additionally, the paper recognizes problems with implementing bank credit rating systems in practice which signifies more study is needed to tackle these limitations. In general sense this is a useful addition for analyzing credit cards because it combines K-means clustering with Bayesian classification. But there should be more empirical validation and comparison to other methods so we can understand its effectiveness completely.

Recency, Frequency, and Monetary(RFM) is an easy method but very effective to apply in market segmentation. In this study, RFM analysis for product segmentation was given in terms of recent sales (R), frequent sales (F), and total money spent(M) using the data mining method. The study has suggested a new procedure for RFM analysis using the k-Means method as well as eight indexes of validity to decide on best number of clusters namely Elbow Method, Silhouette Index, Calinski-Harabasz Index, Davies-Bould in Index Ratkowski Index Hubert Ball-Hall Krzanowski-LaiIndex improving objectivity along with similarity within data during product segmentation thus enhancing accurateness at stock management process. The results of the evaluation show that three clusters (segmentation) is the best number when using k-Means method in RFM analysis. The variance value for this is 0.19113. In the basic idea of RFM analysis, datasets are divided into five clusters equally, with every cluster size being same at 20%. But in this study k-Means method was used and evaluated to get best number of clusters with eight index validity which led to a more objective product clustering showing high similarity in data values; its purpose seems to be improving accuracy for stock management process. For future work, you can make a comparison using particle swarm optimization (PSO), medoid or maximizing-expectancy method to obtain more optimal outcomes and then output them compared with results if the basic RFM analysis method is used. [3]

The paper (Ganesh Arora) [4] used clustering algorithms on credit card data. They evaluated the performance of 4 different clustering algorithms- Furthest First, Filtered Cluster, Density Based Scan, and Hierarchical Clustering through their comprehension about the data structure. The research also examined the impact of parameters to determine optimal cluster number for each algorithm. The methods utilized by this study can be applied for better understanding patterns in credit card use and detecting fraud activities in these transactions. They have found that partition algorithms are more suitable in these cases. The evaluation parameter was: they looked at the time it takes to build the model, along with how many instances were correctly clustered. The study noticed that the FF algorithm showed the highest efficiency on both sets of data. It required only 0.07 seconds to cluster data set 1 and took 0.31 seconds for clustering data set 2. Every other algorithm needed more time than FF algorithm did.

III. METHODOLOGY

A. PROBLEM STATEMENT

The aim of this project is to unveil credit card segmentation using clustering techniques. Credit card companies often need to categorize their customers into distinct segments based on their spending behavior, payment patterns, and other relevant factors. This segmentation enables personalized marketing strategies, risk assessment, and tailored product offerings. However, achieving optimal segmentation manually can be challenging due to the large volume and complexity of credit card data.

B. REQUIRED INFORMATION

To solve this problem, we need access to a comprehensive dataset containing relevant features such as transaction history, demographics, credit limits, and other pertinent variables. This data can be obtained from credit card companies, financial institutions, or publicly available datasets.

C. DATASET DESCRIPTION

We are developing a marketing strategy for a credit card company. The dataset taken into consideration contains 9000 card holders and 18 variables (Balance, Purchases, credit limit) Following is the Data Dictionary for Credit Card dataset :-

D. ALGORITHM SELECTION

For credit card segmentation, three clustering algorithms are chosen: K-means, Agglomerative (hierarchical), and DB-SCAN.

1) K-means Clustering Algorithm

K-means is a widely used clustering algorithm that partitions a dataset into K distinct, non-overlapping clusters. It operates iteratively to minimize the within-cluster sum of squares, aiming to place each data point into the cluster whose centroid (center) is nearest.

Algorithm:

CUST_ID	Customer ID
BALANCE	Amount left in the account balance
BALANCE_FREQUENCY	Balance Updation Frequency
PURCHASES	Total Purchases made
ONEOFF_PURCHASES	Maximum purchase done at a time
INSTALLMENTS_PURCHASES	Purchases in instalments
CASH_ADVANCE	Advances taken by User
PURCHASES_FREQUENCY	Purchase frequency
ONEOFFPURCHASESFREQUENCY	One off purchase frequency updation
PURCHASESINSTALLMENTSFREQUENCY	Instalments Updation Frequency
CASHADVANCEFREQUENCY	Cash Advance Frequency
CASHADVANCETRX	Transactions made in cash with advance
PURCHASES_TRX	Number of purchases
CREDIT_LIMIT	Credit Limit for a User
PAYMENTS	Payments done by User
MINIMUM_PAYMENTS	Minimum payment made at once
PRCFULLPAYMENT	Full payment percentage
TENURE	Expiry of credit card

TABLE 1. Dataset Description

```
Select $K$ points as the initial centroids.
repeat
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

2) Agglomerative Hierarchical Clustering

Agglomerative Hierarchical Clustering is based on the concept of merging similar clusters iteratively until a single cluster is formed. The process can be summarized as follows:

- Compute the proximity matrix
- Let each data point be a cluster

Repeat:

- Merge the two closest clusters
- Update the proximity matrix

Until only a single cluster remains,

Key operation is the computation of the proximity of two clusters, which involves different approaches to defining the distance between clusters distinguish the different algorithms

3) Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering algorithm that groups together data points that are closely packed based on their density. Here's a brief overview of the DBSCAN algorithm:

- Core point: This is a point in a high-density region.
- Density Reachable: A point X is called density reachable if there is a core point from which you can traverse through core points and reach X. i.e we can move by eps steps from a core point B at a time and reach X.
- Density Connected: Two points, I and J, are density-connected if there exists a core point K from which both I and j are density reachable

The combination of K-means, Agglomerative, and DBSCAN clustering algorithms provides a comprehensive approach to credit card segmentation. Each algorithm offers unique advantages in handling different aspects of the data, such as predefined cluster numbers, hierarchical structures, and density-based clusters. By applying these techniques, we aim to enhance the understanding of credit card customer segments and facilitate targeted marketing strategies and risk management practices.

IV. EXPERIMENTS

A. PRE-PROCESSING METHODS

1) Data Cleaning

Data cleaning involves handling missing values to ensure the integrity and completeness of the dataset. Null values are replaced with the mean of the respective attributes to maintain data consistency. We addressed missing values in the dataset by replacing them with the mean of the corresponding attribute, preserving the overall distribution of the data.

2) Discretization

Discretization transforms continuous numerical attributes into categorical attributes, facilitating the handling of frequency-based attributes. This step is particularly useful for attributes related to transaction frequency. We discretized certain numerical attributes into categorical ones, enabling easier interpretation and analysis of transaction frequency patterns.

3) Feature Selection

Feature selection aims to remove redundant or irrelevant attributes from the dataset, optimizing the input space for clustering algorithms. In this dataset, redundant categorical attributes were identified and removed to streamline the analysis process. We pruned redundant categorical attributes from the dataset, ensuring that only relevant features were retained for clustering analysis.

4) Normalization

Normalization scales numerical attributes to a standard range, ensuring uniformity and comparability across features. Z-score normalization was employed to standardize the attributes, centering them around a mean of 0 and a standard deviation of 1. We applied Z-score normalization to the numerical attributes in the dataset, standardizing their scales to facilitate fair comparison and accurate clustering.

5) Dimensionality Reduction

Dimensionality reduction techniques like Principal Component Analysis (PCA) are utilized to reduce the number of features while preserving the most relevant information. By creating new principal components, PCA mitigates the curse of dimensionality and enhances clustering performance. PCA was employed to derive new principal components from the dataset, reducing its dimensionality and capturing the essential variance for clustering analysis.

B. RESULTANT DATASET

The resultant dataset following initial preprocessing steps like data cleaning, discretization, and feature selection, the dataset underwent Z-score normalization to standardize numerical attributes. This ensured fair comparison and prevented features with larger magnitudes from dominating the analysis. Subsequently, PCA was applied to reduce dimensionality while retaining variance.

Post-PCA, the dataset transformed into **three** principal components, each explaining a significant proportion of variance. These components, derived from the most informative combinations of attributes, served as the input features for subsequent clustering algorithms.

C. EVALUATION METHOD/METRICS

The following evaluation metrics were used to determine the performance of the 3 clustering algorithms, which are explained in brief as follows:

1) Silhouette Score

The silhouette score measures the cohesion and separation of clusters. It computes the average distance between each data point and its neighbouring points in the same cluster (a(i)), and the smallest average distance between the data point and points in other clusters (b(i)). A high silhouette score indicates well-defined clusters with instances that are closer to their own cluster than to neighbouring clusters.

The silhouette score s(i) for a data point i is computed as:

$$s(i) = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)}$$

Where:

- a(i) is the average distance from i to other points in the same cluster.
- b(i) is the smallest average distance from i to points in a different cluster.

The overall silhouette score for the dataset is the average of the silhouette scores for all data points.

2) WCSS

WCSS measures the compactness of clusters by summing the squared distances of each data point to its assigned cluster centroid. Lower WCSS values indicate tighter, more homogeneous clusters.

$$WCSS(K) = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2$$

Where:

- n is the number of data points.
- X_i is the ith data point
- μ_j is the centroid of the cluster to which X_i is assigned.
- C is the set of clusters.

3) Elbow Method

In addition to silhouette score and WCSS, the elbow method is often used to determine the optimal number of clusters K. It involves plotting the inertia as a function of K and identifying the "elbow" point, where the rate of decrease in WCSS sharply decreases.

4) Davies-Bouldin Index

The Davies-Bouldin Index quantifies the average similarity between each cluster and its most similar cluster, relative to the average dissimilarity between clusters. A lower index indicates better clustering, with well-separated clusters

$$DB = \frac{1}{n} \sum_{i=1}^{n} max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

Where:

- n is the number of clusters.
- σ_i and σ_j are the average distances from the centroid of cluster i and cluster j to their respective points.
- d(ci,cj) is the distance between the centroids of cluster i and cluster j.

5) Calinski-Harabasz Index

The Calinski-Harabasz Index measures the ratio of between-cluster dispersion to within-cluster dispersion. Higher index values indicate better-defined, well-separated clusters.

$$CH = \frac{B(K)}{W(K)} \times \frac{n-K}{K-1}$$

Where:

- B(K) is the between-cluster dispersion.
- W(K) is the within-cluster dispersion.
- n is the number of data points.
- K is the number of clusters.

6) Elapsed Time

Elapsed time measures the duration taken by each clustering algorithm to complete its execution. It provides insights into the computational efficiency and scalability of the algorithms.

D. EXPERIMENTAL SETUP

1) K-means Clustering

Hyperparameters:

K: The number of clusters. This hyperparameter defines the number of clusters into which the dataset will be partitioned. It is crucial to explore a range of K values to find the optimal number of clusters that best represents the underlying structure of the data. Techniques such as the elbow method and silhouette analysis can aid in selecting an appropriate K value.

Initialization Method: K-means clustering can be initialized using various methods, including random initialization and k-means++. Random initialization selects K data points randomly as initial cluster centroids, which may lead to suboptimal solutions. In contrast, k-means++ selects initial centroids based on a probability proportional to the distance from the existing centroids, which often results in faster convergence and better clustering outcomes.

Characteristics and Properties:

Scalability:K-means is known for its efficiency and scalability, particularly when dealing with large datasets. Its time complexity is linear with the number of data points, making it suitable for datasets with millions of records. Additionally, K-means can benefit from parallelization, allowing it to take advantage of multi-core processors and distributed computing frameworks to further accelerate computation.

Cluster Shape: One of the fundamental assumptions of K-means clustering is that clusters are spherical and isotropic, meaning they have similar shapes and sizes. However, this assumption may not always hold true for real-world datasets, where clusters can exhibit complex shapes and varying densities. As a result, K-means may struggle to accurately capture the underlying structure of non-spherical clusters, leading to suboptimal clustering results.

Robustness: K-means clustering is sensitive to outliers and noise in the data, which can significantly impact the quality of the resulting clusters. Outliers can disproportionately influence the position of cluster centroids and distort the boundaries between clusters. Consequently, it is essential to preprocess the data to remove outliers or employ robust clustering techniques that are less affected by outliers, such as DBSCAN or hierarchical clustering.

Convergence: Although K-means typically converges to a solution, it may converge to a local optimum rather than the global optimum, depending on the initialization of cluster centroids. Multiple runs of the algorithm with different initializations or the use of advanced initialization techniques like k-means++ can mitigate the risk of convergence to suboptimal solutions. Additionally, monitoring convergence criteria such as the change in cluster centroids between iterations can help assess the stability of the clustering process. Parallelization: K-means clustering can be parallelized to expedite computation, especially for datasets with a high dimensionality or a large number of data points. Parallelization techniques leverage the parallel processing capabilities of modern hardware architectures, such as multi-core CPUs and distributed computing frameworks like Apache Spark. By distributing computations across multiple processing units, parallelized K-means implementations can significantly reduce the time required to converge to a solution, enabling faster analysis of large-scale datasets.

2) Agglometative Hierarchical

Hyperparameters:

Linkage Criterion: Agglomerative Hierarchical clustering requires a linkage criterion to determine the distance between clusters during the merging process. Common linkage criteria include:

- Single Linkage: Distance between the closest points of two clusters.
- Complete Linkage: Distance between the farthest points of two clusters.
- Average Linkage: Average distance between all pairs of points in two clusters.
- Ward's Linkage: Minimizes the increase in variance when merging two clusters.

Number of Clusters: Unlike K-means, Agglomerative Hierarchical clustering does not require specifying the number of clusters in advance. Instead, the analyst can control the number of clusters by setting a threshold on the linkage distance or by using dendrogram visualization to decide where to cut the dendrogram to obtain the desired number of clusters.

Characteristics and Properties:

Hierarchical Nature:Agglomerative Hierarchical clustering builds a hierarchy of clusters by iteratively merging the most similar clusters until a single cluster containing all data points is formed. This hierarchical structure provides insights into the relationships and similarities between data points at different levels of granularity, allowing for flexible interpretation and analysis.

Scalability: The time complexity of Agglomerative Hierarchical clustering is O(n3), where n is the number of data points. As a result, it can be computationally expensive, especially for large datasets. However, optimizations such as using efficient data structures (e.g., distance matrices) and heuristic methods (e.g., nearest-neighbor chains) can help mitigate scalability issues.

Cluster Shape: Unlike K-means, Agglomerative Hierarchical clustering does not assume clusters to be spherical or isotropic. Instead, it can handle clusters of arbitrary shapes and sizes, making it suitable for datasets with complex geometries and non-linear separability.

Sensitivity to Distance Metric: The choice of distance metric can significantly impact the clustering results in Agglomerative Hierarchical clustering. Common distance metrics include Euclidean distance, Manhattan distance, and cosine similarity. It is essential to select a distance metric that is appropriate for the data and the underlying domain knowledge.

Interpretability : Agglomerative Hierarchical clustering produces a dendrogram, which visualizes the hierarchical structure of the clusters. The dendrogram provides valuable insights into the relationships between data points and allows analysts to explore different levels of clustering granularity. This interpretability makes Agglomerative Hierarchical clustering particularly useful for exploratory data analysis and hypothesis generation.

Memory Usage: Agglomerative Hierarchical clustering requires storing the pairwise distances between all data points in memory, which can lead to high memory usage, especially for large datasets. Techniques such as memory-efficient data structures and sparse distance matrices can help alleviate memory constraints and enable clustering of larger datasets.

3) DBSCAN

Hyperparameters:

Epsilon ϵ: Epsilon defines the radius within which to search for neighboring points. Points within this radius are considered part of the same neighborhood.

Minimum Number of Points (MinPts): MinPts specifies the minimum number of points required to form a dense region (core point). Points that have at least MinPts within their epsilon neighborhood are considered core points.

Characteristics and Properties

Density-Based Clustering: DBSCAN is a density-based clustering algorithm that groups together closely packed points into clusters based on the density of data points in the feature space. It identifies core points, border points, and noise points, thereby allowing for the discovery of clusters of arbitrary shapes and sizes.

Adaptive to Data Density: DBSCAN is robust to variations in data density and can automatically adapt to different densities within the dataset. It can handle clusters of varying shapes and densities without requiring prior knowledge of the number of clusters or their shapes.

Noise Tolerance: DBSCAN is capable of identifying and handling noise points, which are data points that do not belong to any cluster. Noise points are typically isolated points or points in low-density regions that do not meet the criteria for core points or border points.

Cluster Shape Flexibility: Unlike K-means, which assumes spherical clusters, DBSCAN can discover clusters of arbitrary shapes, including non-linear and irregular shapes. It does not make any assumptions about the shape, size, or density of clusters, making it suitable for datasets with complex geometries.

Parameter Sensitivity: The performance of DBSCAN can be sensitive to the choice of hyperparameters, particularly epsilon ϵ and the minimum number of points (MinPts). Selecting appropriate values for these parameters is crucial for obtaining meaningful clustering results and avoiding overfitting or underfitting.

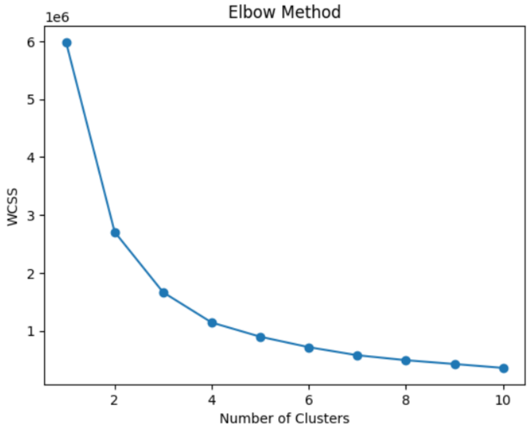
Handling Large Datasets: DBSCAN's time complexity is O(nlog(n)) for indexing nearest neighbors and O(n^2) for the clustering step. While it can be efficient for moderately sized datasets, it may face scalability issues with very large datasets. Approximate nearest neighbor search methods and parallel implementations can help mitigate scalability concerns.

V. RESULT

A. NUMBER OF CLUSTERS

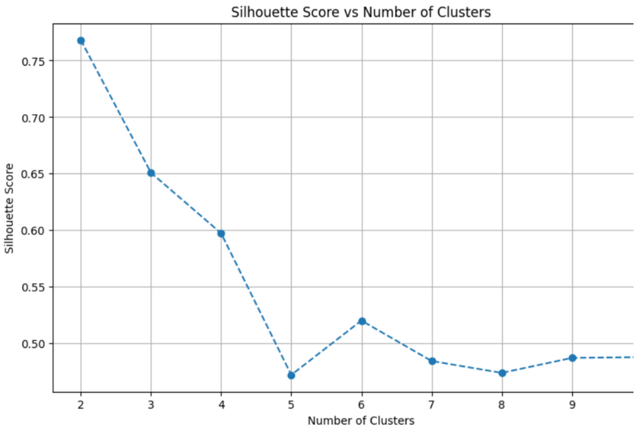
1) Elbow Method

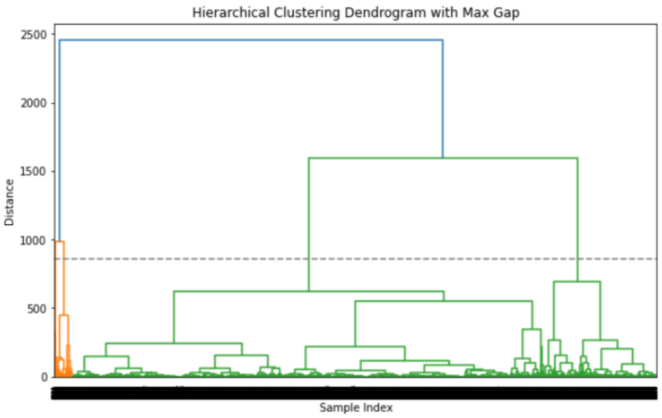
The optimal number of clusters for the K-means algorithm was calculated to be 3 using the elbow method. Within-cluster sum of squares (WCSS) is calculated for each value of cluster(k). The elbow method plot is shown below.



2) Silhouette Score

The optimal number of clusters for the K-means algorithm was calculated to be 2 using the Silhouette score. The silhouette score is calculated as the difference between the min of points in other cluster to the average distance of points in the same cluster. The Silhouette score plot is shown below.

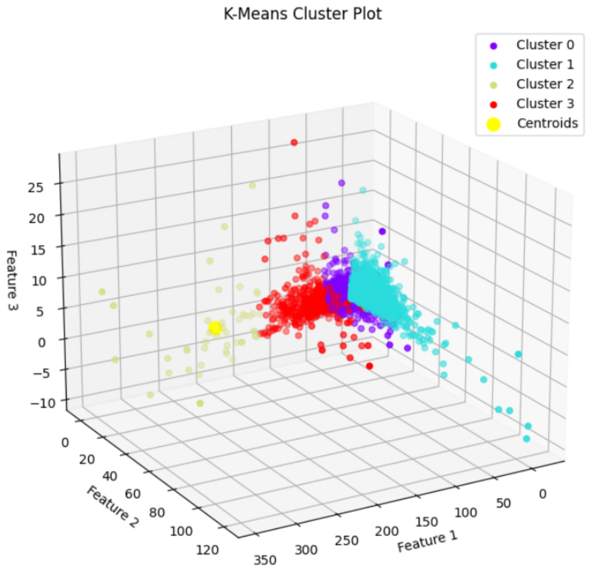




```
Optimal number of clusters: 4
```

B. K-MEANS CLUSTERING

The dataset was clustered using k=4. The cluster plot for k-means is shown in the below figure.

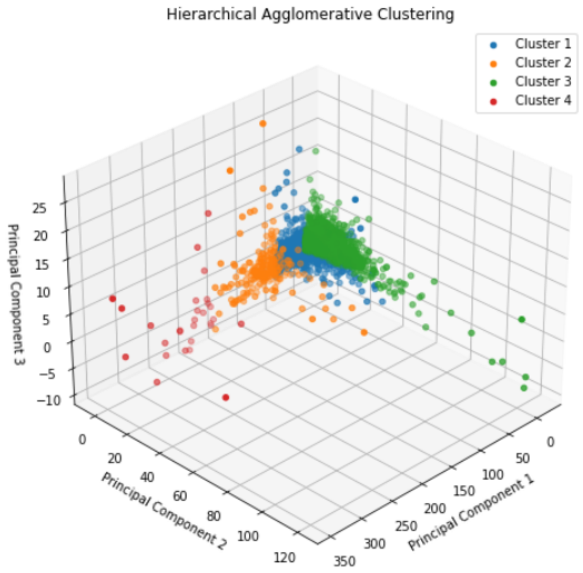


The metrics taken to evaluate the performance of the model were: Silhouette score, Davis-Bouldin Index, Calinski-Harabasz Index, Inertia and Elapsed time. The total sum of squares(TSS) was found to be 1.071×10^{11}, the within cluster sum of squares(WCSS) was 8.99×10^5 and percentage of total variance explained was 99.99%. The clusters had frequencies of 6849 (C1), 1664 (C0), 391 (C3), and 46 (C2).

C. HIERARCHICAL CLUSTERING

Hierarchical clustering is performed on the dataset. The dendrogram obtained after clustering is given be-

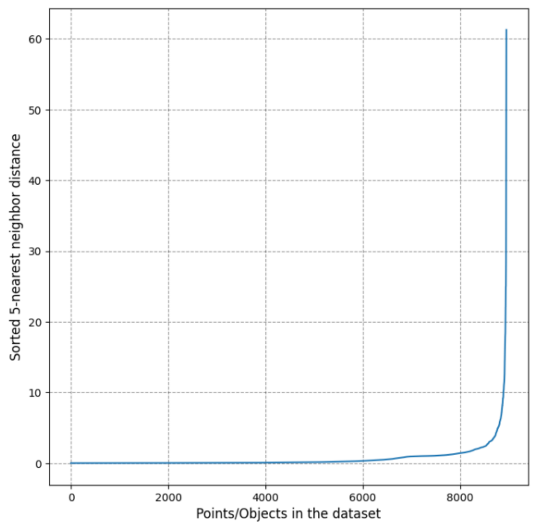
We identify the largest vertical gap in the dendrogram and look for largest difference between consecutive merges and draw a line at that height to get the best clusters. The cluster plot for the Hierarchical agglomerative clustering is also shown below.



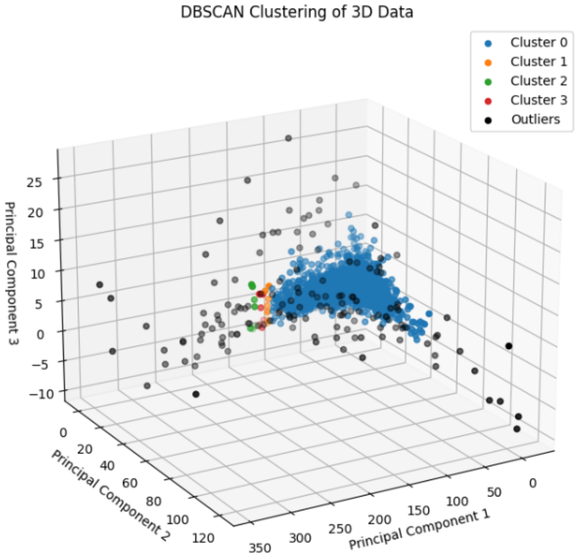
The optimal number of clusters we observed is 4 using the max gap. The frequencies obtained in each clusters are 6998(C2), 1684(C0), 240(C1), 28(C3).

D. DBSCAN

Density based clustering (DBSCAN) is a clustering algorithm that groups together data points that are closely packed based on their density. The ϵ value is calculated using the sorted 5-nearest neighbor distance to the point in the dataset.



From the k dist plot we get the ϵ value to be between 1-10. We perform hyperparameter tuning on the ϵ value and min samples to find the optimal value. We get the optimal value of $epsilon$ as 4.81, optimal minimum samples to be 5 and the best silhouette score obtained is 0.783. The cluster plot we get after implementing the DBSCAN algorithm is given below.



.

We observe that 142 data points came out as noise points and in the remaining majority of customers belong to Cluster 0.

E. EVALUATION METRICS/METHODS

For each clustering algorithms, the metrics used for evaluating are Silhouette Score, Davies-Bouldin Index, Calinski-Harabasz Index, Inertia and Elapsed time. The values of these are provided in the table.

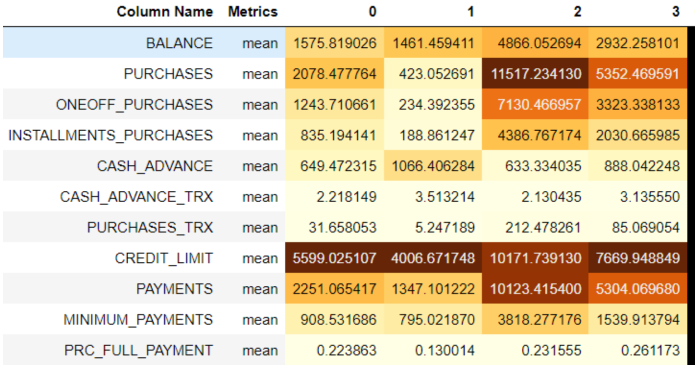
Higher Silhouette Score, Lower Davies-BouldIn Index, Higher Calinski-Harabasz Index and Lower Inertial and Elapsed time is required to say that algorithm performed clustering the better as clusters would be well seperated.

	K-means Clustering	Hierarchical Clustering	DBSCAN Clustering
Silhouette Score	0.768	0.618	0.783
Davies-Bouldin Index	0.563	0.572	1.953
Calinski-Harabasz Index	108244.88	11646.67	879.16
Inertia	2.707^6	6.27*10^6	3.6585*10^5
Elapsed Time	0.0086	5.798	0.82189

F. CLUSTER CHARACTERISTICS

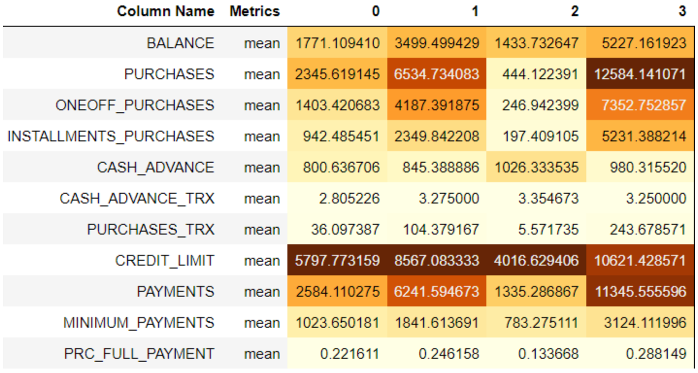
The characteristics of each cluster for each algorithm is given below. We added the cluster labels to the original dataframe and computed mean of our features for a particular cluster to see how customer habits lie in that cluster.

1) K-means Clustering



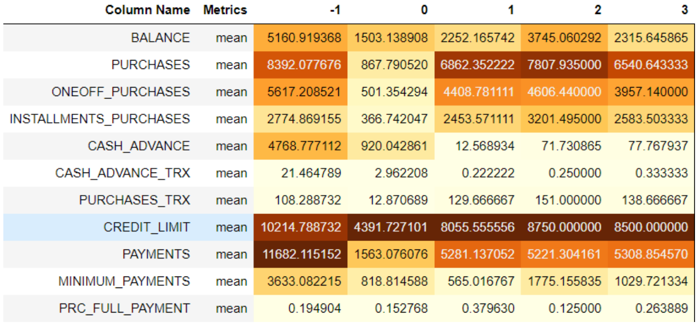
Column Name	Metrics	0	1	2	3
BALANCE	mean	1575.819026	1461.459411	4866.052694	2932.258101
PURCHASES	mean	2078.477764	423.052691	11517.234130	5352.469591
ONEOFF_PURCHASES	mean	1243.710661	234.392355	7130.466957	3323.338133
INSTALLMENTS_PURCHASES	mean	835.194141	188.861247	4386.767174	2030.665985
CASH_ADVANCE	mean	649.472315	1066.406284	633.334035	888.042248
CASH_ADVANCE_TRX	mean	2.218149	3.513214	2.130435	3.135550
PURCHASES_TRX	mean	31.658053	5.247189	212.478261	85.069054
CREDIT_LIMIT	mean	5599.025107	4006.671748	10171.739130	7669.948849
PAYMENTS	mean	2251.065417	1347.101222	10123.415400	5304.069680
MINIMUM_PAYMENTS	mean	908.531686	795.021870	3818.277176	1539.913794
PRC_FULL_PAYMENT	mean	0.223863	0.130014	0.231555	0.261173

2) Hierarchical Clustering



Column Name	Metrics	0	1	2	3
BALANCE	mean	1771.109410	3499.499429	1433.732647	5227.161923
PURCHASES	mean	2345.619145	6534.734083	444.122391	12584.141071
ONEOFF_PURCHASES	mean	1403.420683	4187.391875	246.942399	7352.752857
INSTALLMENTS_PURCHASES	mean	942.485451	2349.842208	197.409105	5231.388214
CASH_ADVANCE	mean	800.636706	845.388886	1026.333535	980.315520
CASH_ADVANCE_TRX	mean	2.805226	3.275000	3.354673	3.250000
PURCHASES_TRX	mean	36.097387	104.379167	5.571735	243.678571
CREDIT_LIMIT	mean	5797.773159	8567.083333	4016.629406	10621.428571
PAYMENTS	mean	2584.110275	6241.594673	1335.286867	11345.555596
MINIMUM_PAYMENTS	mean	1023.650181	1841.613691	783.275111	3124.111996
PRC_FULL_PAYMENT	mean	0.221611	0.246158	0.133668	0.288149

3) DBSCAN Clustering



Column Name	Metrics	-1	0	1	2	3
BALANCE	mean	5160.919368	1503.138908	2252.165742	3745.060292	2315.645865
PURCHASES	mean	8392.077676	867.790520	6862.352222	7807.935000	6540.643333
ONEOFF_PURCHASES	mean	5617.208521	501.354294	4408.781111	4606.440000	3957.140000
INSTALLMENTS_PURCHASES	mean	2774.869155	366.742047	2453.571111	3201.495000	2583.503333
CASH_ADVANCE	mean	4768.777112	920.042861	12.568934	71.730865	77.767937
CASH_ADVANCE_TRX	mean	21.464789	2.962208	0.222222	0.250000	0.333333
PURCHASES_TRX	mean	108.288732	12.870689	129.666667	151.000000	138.666667
CREDIT_LIMIT	mean	10214.788732	4391.727101	8055.555556	8750.000000	8500.000000
PAYMENTS	mean	11682.115152	1563.076076	5281.137052	5221.304161	5308.854570
MINIMUM_PAYMENTS	mean	3633.082215	818.814588	565.016767	1775.155835	1029.721334
PRC_FULL_PAYMENT	mean	0.194904	0.152768	0.379630	0.125000	0.263889

VI. CONCLUSION

A. K-MEANS INTERPRETATION

The customers in each cluster belong to

1) **Cluster 0(Cash Convenience Consumers)**: Customers in this cluster hold moderate balances, the frequency of balances are always changing, and the frequency of cash in advance and cash in advance is high comparatively. The interest rate for customers in this cluster is at a low level among all other clusters. They possess the second highest credit limit and payments among the four clusters. However, users of credit cards rarely make installments or one-off purchases here; they also have tenure, which ranks third out of four clusters. Hence, we can understand that the clients in this group solely utilize credit cards for the purpose of money withdrawals or cash advances.
2) **Cluster 1(Instalment Users)**: In this cluster, the credit cards are used by customers mainly for installment purposes. This aligns with the fact that there is a somewhat high level of transactions using installments in this group. Additionally, customers in cluster 1 have characteristics such as very large amounts per transaction and minor frequencies, along with transactions of cash in advance being small too. The customers in this cluster do not often make cash in advance payments. They have a low rate of frequency and amount for this method, indicating that they are a good fit for credit cards, which are mainly used to pay in installments.
3) **Cluster 2(Full-Payment Pioneers)**: The customers in this cluster use the credit card of the bank actively, as can be seen from the balance that often changes and its amount is more compared to other clusters. Moreover, this cluster shows a higher mean value in a number of aspects when contrasted with other clusters. Customers with credit cards in this cluster also use credit cards for transactions and installments. The frequency of transactions and installments is higher in this cluster. The tenure's relatively high value indicates that the credit scoring is very good in this cluster. However, we see that the cash advance frequency is low. This means that these customers do not borrow money in advance often; rather, they use credit cards more frequently.
4) **Cluster 3(Starters/Laggards)**: in this cluster, the customer hardly uses credit cards for transactions and installments because they have a low balance which rarely changes frequency. Their installments are also very low. Plus, a small credit limit indicates that customers use their credit cards to process credit transactions very little or almost never. It is same with customers in this cluster who hardly make cash advances as well. Therefore, we can suppose that people in this group use credit cards for cash advance procedures with an appropriate regularity. The small balance could imply that the customers in this category are students or new users of credit cards at this particular bank. They might also be individuals who don't comprehend a lot about how credit cards work or those who feel that using one is just a way for banks to scam people by giving them more money than they can repay back.

B. HIERARCHICAL INTERPRETATION

The customers in each cluster belong to

1) **Cluster 0(Steady Spenders)**: Customers in this group show low one off purchase frequency and cash advance frequency. They have expected spending habits. We understand they use credit cards because of the high value of their purchases, yet it seems to be under control. They possess balanced credit and have moderate credit limits. The payments show regularity in repaying debts, which helps to maintain good credit scoring. Almost 40% of their purchases are made via installment buying.
2) **Cluster 1(Instalment Users)**: For the case of credit cards in this cluster, it's clear that customers use them mostly for installment. It goes along with the truth that there is a somewhat high level of transactions using installments in this group. Moreover, customers from cluster 1 have traits like very large amounts per transaction and high frequencies for purchases, but cash in advance is small. The people in this cluster don't usually make cash in advance payments. They have very high purchase installment frequencies, indicating that these customers are better suited to credit cards, which are normally used for paying later by installment.
3) **Cluster 2(Starters)** ; In this cluster we observe that the spending patterns are less compared to other clusters concluding that they are either new to understanding the use and benefits of credit cards. They have relatively low balances and low credit limits. They have the lowest tenure average among the other groups. It is also noteworthy that in hierarchical clustering, many customers fall under this category while for K-means the frequency for this category was considerably small.
4) **Cluster 3(Full Payment Users)**: The customers in this cluster are active users of the bank's credit card, as indicated by their balance that frequently changes and its quantity is more than other groupings. Additionally, when compared with different clusters, this cluster shows a higher mean value in several aspects. Customers who possess a credit card here also employ it for transactions and installments. The frequency of transactions and installments is higher in this cluster. The tenure's relatively high value indicates that the credit scoring is very good in this cluster. However, we see that the cash advance frequency is low. This indicates that these customers do not typically borrow money beforehand but instead utilise credit cards more regularly. The high value of balance, credit limits, and transaction frequencies suggest they are very active users.

C. DBSCAN INTERPRETATION

The customers in each cluster belong to

1) **Cluster 0(Cash Access Consumers)**: Customers having credit cards, who mostly use them for cash advances, usually display particular financial habits and choices. These persons often depend on their credit cards to get quick money. They may use the card when a situation arises where they need cash urgently or unexpectedly; hence, they could give importance to ease and availability by using features of cash advance for emergencies or unexpected costs. Yet, this kind of usage could show a problem with having enough liquid money or planning financially. Because cash advances normally bring about more interest rates and charges when matched against other credit card dealings. Hence, customers who often make use of cash advances might have a greater possibility to pile up debts and face financial pressure. To handle the requirements and dangers linked with this customer group, financial institutions can customize their services and offerings by giving out monetary education as well as providing substitute solutions for temporary cash requirements. Most users are present in this cluster,
2) **Cluster 1(High One-off Purchase Customers)**:People who use credit cards in a high one-off manner are those who occasionally do big, single transactions instead of many small buys. Such users might use their credit card for large expenses like buying a costly item or handling an unforeseen emergency situation. The way they utilize their card signifies a liking for the flexibility and ease-of-use that comes with credit, helping them handle substantial costs without requiring immediate payment from their own pocket. Nevertheless, this could also imply a necessity for cautious financial organization. Such dealings have the potential to create significant balances and possibly increased interest costs if not handled with prudence. We observe the ratio to the one off purchases total purchases is nearly 64% which says that significant amount of transactions happen this way. The minimum payments made by the user is also less.
3) **Cluster 2(Active Cardholders)**: We can observe from the cluster interpretation that these customers make use of credit cards the most. Having greater amount of purchases and higher purchase frequency is an indication of that. Credit card companies more often target these type of customers because of the volume of transactions these go through, they are the star performers in company's terms. So companies shower them with higher credit limit and tenure too.
4) **Cluster 3(Responsible Users)**: Regular customers who make high payments on their credit card accounts consistently demonstrate admirable financial restraint and responsible spending habits. These people place a high value on paying off credit card debt on time in order to avoid paying hefty interest and maintain excellent credit scores. Their tendency to pay more could be an indication of a sound financial position, showing that they have enough money and can manage their credit commitments through careful planning. Financial institutions may treat these consumers well, granting them perks like higher credit limits, lower interest rates, or rewards for responsible use. They can also receive tailored financial advice and products that are made to fit their unique credit management needs.

Based on the provided model performances for K-means, hierarchical clustering, and DBSCAN, we can draw several conclusions regarding their effectiveness in clustering credit card customer data.

K-means clustering demonstrates the highest Silhouette score of 0.7681, indicating well-defined and dense clusters with good separation between them. The Davies-Bouldin Index of 0.5632 suggests low inter-cluster similarity and high intra-cluster coherence, further supporting the effectiveness of K-means in creating distinct clusters. The Calinski-Harabasz Index of 10824.88 reinforces these findings by indicating compact and well-separated clusters. However, the relatively high inertia of 2707751.00 may suggest some degree of dispersion within the clusters.

Hierarchical clustering yields a Silhouette score of 0.6184, indicating moderate clustering quality with some overlapping clusters. The Davies-Bouldin Index of 0.5723 suggests relatively low inter-cluster similarity and moderate intra-cluster coherence. The Calinski-Harabasz Index of 11646.67 suggests well-separated clusters, although it is slightly higher compared to K-means. Hierarchical clustering also has a longer elapsed time of 5.7988 seconds, indicating higher computational overhead.

DBSCAN achieves the highest Silhouette Score of 0.7831, indicating dense and well-separated clusters with clear boundaries. However, the Davies-Bouldin Index of 1.9530 suggests higher inter-cluster similarity and lower intra-cluster coherence compared to K-means. The Calinski-Harabasz index of 879.16 indicates less compact clusters compared to K-means and hierarchical clustering. Despite this, DBSCAN demonstrates the lowest inertia of 365858.62, suggesting tight clustering with minimal dispersion. The elapsed time of 0.8219 seconds is higher than K-means but lower than hierarchical clustering.

K-means clustering performs well in creating compact and well-separated clusters, making it suitable for this credit card customer data. However, DBSCAN shows promising results with higher Silhouette Score and lower inertia, indicating denser and more coherent clusters. Hierarchical clustering, while effective, exhibits moderate performance compared to K-means and DBSCAN, with longer computational time and slightly lower clustering quality. Therefore, based on the provided metrics, K-means may be the preferred choice for clustering credit card customer data.

REFERENCES

[1] A. Abdulhafedh, “Incorporating k-means, hierarchical clustering and pca in customer segmentation,” Journal of City and Development, vol. 3, no. 1, pp. 12–30, 2021.

[2] S. J. Saritha, P. P.Govindarajulu, K. R. Prasad, S. R. Rao, and C.Lakshmi, “Clustering methods for credit card using bayesian rules based on k-means classification,” International Journal of Advanced Computer Science and Applications, vol. 1, DOI 10.14569/IJACSA.2010.010416, no. 4, 2010. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2010.010416

[3] R. Gustriansyah, N. Suhandi, and F. Antony, “Clustering optimization in rfm analysis based on k-means,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 18, DOI 10.11591/ijeecs.v18.i1.pp470-477, no. 1, pp. 470–477, 2019. [Online]. Available: https://doi.org/10.11591/ijeecs.v18.i1.pp470-477

[4] G. Arora and D. Kishore, “Evaluation of clustering algorithms for credit card data set using weka,” Academic Social Research, vol. 8, no. 1, 2019, impact Factor: 6.209 Peer-Reviewed, International Refereed Journal. [Online]. Available: https://asr.academicsocialresearch.co.in/index.php/ASR/article/view/639