TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

Unsupervised Analysis of Bank Customer Segmentation

BMCS2114 MACHINE LEARNING

2023/2024

Student's name/ ID Number	:	Khong Wei Xian 22WMR02868
Student's name/ ID Number	:	Chin Wei Lun 22WMR04032
Programme	:	Bachelor of Science in Management Mathematics with Computing
Tutorial Group	:	1
Tutor's name	:	Ms Lim Siew Mooi

Unsupervised Analysis of Bank Customer Segmentation

Abstract—**The financial industry's ongoing transformation is largely fueled by the deluge of customer data emanating from digital transactions. In particular, banks, with their traditionally broad customer base, face the intricate task of understanding a vast array of individual financial behaviors and preferences. This research takes on the challenge of customer segmentation through the lens of unsupervised learning, analyzing over one million transactions by more than 800,000 customers of a bank in India. It employs sophisticated algorithms like K-means, DBSCAN, and hierarchical clustering to categorize customers based on a range of attributes, including demographic details, account balances, and transactional behaviors. The aim is to transcend the confines of generic services and forge a pathway toward personalized offerings, risk minimization, and compliance with regulatory norms. This study positions segmentation as a cornerstone for strategic decision-making in banking operations, enhancing customer satisfaction and fostering financial security.**

Keyword—Financial Analysis, Clustering Algorithms, Unsupervised Learning, K-Means, Spectral Clustering.

I. INTRODUCTION

In the realm of banking, the conventional methodology of customer segmentation has been rudimentary and broad-brushed, often missing the mark in addressing the nuanced financial needs of individual clients. As digital interactions amass a wealth of transactional and demographic data, banks are impelled to adopt a more granular and customer-centric approach to service design. Amidst heightened expectations for personalized banking experiences, this research seeks to redefine the art and science of customer segmentation.

The advent of digitalization in banking has been a double-edged sword, bringing about an abundance of data while simultaneously raising the bar for customer expectations. The modern bank customer no longer sees banking as a mere transactional relationship but as a dynamic and personalized financial partnership [15]. This paradigm shift calls for a segmentation model that is both data-informed and adaptable, capable of discerning the evolving patterns of customer behavior and financial standing.

Our research tackles the complexities inherent in a sizable and diverse data set, grappling with the challenge of extracting meaningful segments that encapsulate the multifaceted nature of customer profiles. By capitalizing on advanced machine learning techniques, we aim to illuminate the latent structures within the data, thus empowering banks to tailor their offerings more effectively and responsively.

II. PROBLEM STATEMENT

The task at hand is multifaceted: to accurately parse and make sense of the extensive and diverse array of customer data for the purpose of segmentation. Traditional segmentation techniques have proved insufficient in capturing the dynamic and complex nature of customer behavior, especially in light of the rapid evolution in banking practices and customer expectations. This deficiency calls for a more sophisticated, analytics-driven approach to segment the customer base, allowing for precise identification and categorization of varied customer groups.

Given the scale and intricacy of the data—a melting pot of demographics, transaction histories, and account details—the challenge extends beyond mere statistical analysis. The resulting customer segments must not only reflect statistical validity but must also resonate with practical banking applications, thereby facilitating the personalization of services and products . Moreover, these segments are expected to provide a strategic framework for the bank's operational decisions, aligning with the dual

objectives of enhancing customer satisfaction and improving risk management.

Consequently, this study endeavors to construct a robust and dynamic segmentation model using a blend of unsupervised learning algorithms. These algorithms will be assessed for their effectiveness in delivering actionable insights, with a particular focus on their ability to accommodate the complexity and scale of the dataset. The proposed segmentation model aspires to be more than an academic exercise; it seeks to be a practical tool for banks to adapt their strategies in an ever-shifting financial landscape, providing personalized services that meet individual needs while simultaneously safeguarding against financial risks.

III. OBJECTIVES

This study explored 5 Clustering Algorithms: K-Means Clustering, Hierarchical Clustering, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), Gaussian Mixture Models (GMM), Spectral Clustering. The objectives are:

- To apply the 5 clustering algorithms to categorize bank customers into distinct groups based on their risk profiles.
- To develop a clustering algorithm based on the 5 unsupervised learning techniques.
- To evaluate the performance of the 5 clustering algorithms using appropriate metrics.

IV. LITERATURE REVIEW

A. K-Means Clustering

K-means clustering is a popular unsupervised machine learning algorithm used for clustering data into groups or clusters based on their similarities. The algorithm partitions a dataset into K distinct, non-overlapping clusters, where each data point belongs to the cluster with the nearest mean, serving as the prototype of the cluster. The K-Means algorithm is an iterative process that partitions a dataset into K clusters. The value of K is determined by the user and represents the number of clusters the algorithm should create. The algorithm works by assigning each data point to the nearest cluster center, which is also known as the centroid. The algorithm then recalculates the centroid of each cluster based on the data points assigned to it and repeats the process until the centroids no longer move significantly [9].

K-Mean can be used to group customers into clusters based on spending habits and transaction history to offer targeted marketing campaigns [1]. It also classifies customers into different risk categories based on their financial behaviors and demographics to tailor credit offers.

In summary, K-Means clustering is a powerful technique for identifying patterns in data. Understanding the basics of the algorithm, the role of centroids, and the importance of choosing the right number of clusters are key to successfully applying this technique.

B. Hierarchical Clustering

Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. In general, the strategies for hierarchical clustering fall into two types:

Agglomerative: This is a "bottom-up" approach. Initially, each data point is considered as an individual cluster. Then, pairs of clusters are merged as one moves up the hierarchy, based on the similarity between clusters. This process is iterated until all points are merged into a single overarching cluster. The similarity between clusters can be determined in various ways, such as the distance between the closest points in each cluster (single linkage), the distance between the farthest points in each cluster (complete linkage), the average distance between all points in the clusters (average linkage), or the increase in variance for the clustered items (Ward's method).

Divisive: This is a "top-down" approach. Initially, all points are considered as part of one big cluster. Then, this cluster is divided into smaller clusters. This process is repeated recursively, as one moves down the hierarchy, until each data point stands alone as a cluster. The divisions are made by trying to maximize the distance (or minimize the similarity) between the resulting clusters [10].

The result of hierarchical clustering is usually presented in a dendrogram, a tree-like diagram that records the sequences of merges or splits.

In the financial industry, it can be used to identify groups of customers with similar buying behaviors to develop bundled product offerings that meet the specific needs of each segment [3]. Beside that, enhance CRM strategies by identifying hierarchies or layers of customer groups based on their value, loyalty, or engagement levels.

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

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular clustering algorithm characterized by its ability to identify clusters of varying shapes and sizes in a data set, based on the density of data points. Here are the key features and steps involved in DBSCAN:

Density-based: Unlike centroid-based algorithms like k-means, DBSCAN groups together points that are closely packed together, marking them as part of a cluster, while labeling points that lie alone in low-density regions (whose nearest neighbors are too far away) as noise [11].

Core Points: In DBSCAN, a core point is defined as a point that has at least a minimum number of other points (MinPts) within a given radius (ε). These MinPts and ε are parameters provided by the user.

Border Points: These are points that are not core points (as they have fewer than MinPts within a distance ε), but are close to a core point. They are considered part of a cluster, but they do not possess the ability to expand the cluster.

Noise: A point that is neither a core nor a border point is considered noise.

Process: The algorithm starts with an arbitrary point and retrieves all points density-reachable from this point based on ε and MinPts. If this point is a core point, a cluster is formed. Then, all reachable points from this core point are added to the cluster, and the process repeats for each of these points, expanding the cluster. If the point is not a core point, DBSCAN visits the next point. This process continues until all points have been processed.

DBSCAN is particularly effective for tasks such as identifying clusters of arbitrary shapes and sizes, separating noise, and finding outliers. However, it may struggle with data of varying densities and is sensitive to the selection of ε and MinPts parameters. Despite these challenges, it remains a widely used algorithm due to its versatility and the fact that it does not require pre-specification of the number of clusters in the data.

The use of DBSCAN can be used in fraud detection. Segment customers into clusters based on unusual spending patterns to identify potential fraudulent activity. Also, detects dense clusters representing high-value customers to focus retention efforts and personalized services [4].

D. Gaussian Mixture Models (GMM)

A Gaussian Mixture Model (GMM) is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. It is used to model the presence of subpopulations within an overall population without requiring that an observation explicitly belong to one subpopulation or another; this is achieved through the concept of mixture modeling.

Here are the key aspects of Gaussian Mixture Models:

Mixture of Gaussians: A GMM sums multiple Gaussian density functions, each identified by parameters mean (μ), variance (σ^2), and a prior probability (the mixture component weights). The number of these components (i.e., Gaussian distributions) is a parameter of the model and must be determined beforehand or through model selection criteria.

Expectation-Maximization (EM): The parameters of GMM (the means, variances, and mixture weights of each Gaussian component) are typically estimated using the Expectation-Maximization (EM) algorithm. EM iterates between determining the expected component membership (expectation step) and updating the parameters of the Gaussians (maximization step), thereby maximizing the likelihood of the data given the model.

Soft Clustering: Unlike hard clustering methods like K-means, GMM provides soft clustering. For each data point, the model estimates probabilities (responsibilities) of belonging to each of the Gaussian distributions, which represents the data point's membership in each cluster. This can be particularly useful in scenarios where the boundaries between clusters are not well-defined.

Cluster Shapes: Because each cluster is modeled as a Gaussian distribution, GMM can accommodate clusters that have different sizes and different orientations (unlike K-means, which assumes that all clusters are spherical).

Application Areas: GMMs are widely used in various fields such as image processing, pattern recognition, data clustering, and anomaly detection. They are particularly popular in areas involving complex, real-world data that exhibits varied densities and structures [12].

Challenges: The main challenges in using GMMs include deciding the number of Gaussian components to use (which can significantly impact model performance), ensuring convergence of the EM algorithm to the global maximum likelihood (since it can get stuck in local maxima), and computational costs for large datasets or models with many components.

GMM provides a flexible and probabilistic approach to clustering, offering insights into the probability of membership in each cluster rather than forcing a hard assignment, which can be particularly valuable in uncertain or overlapping data scenarios.

GMM can be used in Customer Lifetime Value Prediction, grouping customers based on their transactional data to predict future profitability and lifetime value. For the Market Segmentation, identify different market segments within the bank's customer base to tailor product development and marketing strategies [12].

E. Spectral Clustering

Spectral clustering is a type of clustering algorithm that uses the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions. The similarity matrix represents how similar or close every pair of points is in the dataset. Spectral clustering is particularly noted for its ability to identify clusters that are not necessarily globular and can capture complex cluster structures that traditional methods like k-means clustering might not be able to identify.

Here’s a breakdown of how spectral clustering works:

Similarity Graph Construction: First, construct a similarity graph to represent the data. Each node in the graph represents a data point, and the weight of the edge between two nodes reflects the similarity between the two corresponding data points. There are different ways to construct this graph, such as using the ε-neighborhood graph, the k-nearest neighbor graph, or the fully connected graph, with weights typically determined using the Gaussian (radial basis function) kernel.

Graph Laplacian: After the similarity graph is constructed, compute the Laplacian matrix of the graph. The Laplacian is a matrix representation that captures the structure of the graph. It is used in spectral clustering to understand the graph's connectivity and to find groups of nodes that are more connected to each other than to the rest of the graph.

Eigenvalue Decomposition: Perform eigenvalue decomposition on the Laplacian matrix. The eigenvectors that correspond to the smallest non-zero eigenvalues provide useful information about the clustering structure of the data.

Dimensionality Reduction: Select a number of eigenvectors (based on the desired number of clusters or other criteria) and use these to form a new dataset with reduced dimensionality. Each point in the new dataset corresponds to a point in the original dataset but is represented in the new space formed by the selected eigenvectors.

Clustering: Apply a standard clustering technique, such as k-means, to the points in the reduced dimensional space to assign each original data point to a cluster.

Spectral clustering is particularly effective when dealing with clusters of different shapes and sizes, and it's often used when the structure of individual clusters is non-convex. Unlike many other clustering techniques, it does not make strong assumptions about the form of the clusters. However, spectral clustering can be computationally intensive, especially for large datasets, because it involves constructing a similarity matrix and computing eigenvectors of the Laplacian matrix. Despite these computational challenges, spectral clustering is favored in various applications, including image segmentation, social network analysis, and bioinformatics, due to its flexibility and effectiveness in capturing complex cluster structures.

Application of Spectral Clustering can be used in discovering communities within the customer base to foster network-based marketing or referral programs. Use this technique to make Cross-Selling Strategies. Group customers based on their product usage patterns to identify opportunities for cross-selling and upselling [5].

V. RESEARCH METHODOLOGY

A. Data Collection

This dataset encompasses bank customers recorded in the India from 2016 with 1,048,567 million records and a total of 9 attributes (https://www.kaggle.com/datasets/shivamb/bank-customer-segmentation/data)

All 9 features were used in this bank customers analysis as shown in Table 1.

TABLE I. Description of Features

Features	Description
TransactionID	Unique Transaction ID
CustomerID	Unique Customer ID
CustomerDOB	Date of Birth
CustGender	Gender
CustLocation	Location
CustAccountBalance	Account Balance
TransactionDate	Transaction Date
TransactionTime	Transaction Time (using Unix timestamp)
TransactionAmount (INR)	Amount in INR

B. Data Preprocessing

In this analysis, several preprocessing steps will be done. There are handling missing values, outliers, contaminated data, inconsistent data, duplicate data and data type issues. Initially, the dataset needs to be clean. The first step is to handle the missing value. The dataset has four features contain missing value, 'CustomerDOB' column has 3,397 missing values, 'CustGender' column has 1,100 missing values, 'CustLocation' column has 151 missing values, and 'CustAccountBalance' column has 2,369 missing values. To handle these missing values, fill the Unknown for the 'CustomerDOB', 'CustGender' , 'CustLocation' which are a categorical column. Also, fill the mean value for the 'CustAccountBalance' which is a numerical column.

The next step is to detect the outliers. There are several methods to do this, using the Interquartile Range Rules (IQR) here to detect outliers. Selected 'CustAccountBalance' and 'TransactionAmount (INR)' columns to detect. 'CustAccountBalance' column has 139,723.00 number of outliers and 'TransactionAmount (INR)' column has 112,134.00 numbers of outliers. Removing the outliers from the data set will lower the total dataset rows from 1,048,567 to 827,120, a total of 221,447 records removed.

Following step is to find the contaminated data. For example, it may have negative values in 'CustAccountBalance' and 'TransactionAmount (INR)', or impossible dates in 'CustomerDOB' or 'TransactionDate'. There are no negative values, which is appropriate since account balances should not be negative. There are no negative values, which is correct as transaction amounts should be non-negative. There are no invalid dates, which is good, but for the 'CustomerDOB' with invalid dates, replace it with a placeholder like Unknown.

Sometimes there are different labels meaning the same thing (like "male", "Male", "M"). So it needs to check for the inconsistent data for the "CustGender" column. This will be the last part of data preprocessing.

C. Exploratory Data Analysis

Figure 1 shows the graph of Account Balances Among Customers to understand how the account balances are distributed, whether most customers have low, medium, or high balances, and to identify any outliers in account balances. The majority of account balances seem to fall below 20,000 INR, as indicated by the high frequency (height of the bars) in this range. There is a long tail extending to the right, indicating that there are a small number of customers with very high account balances, up to 140,000 INR.

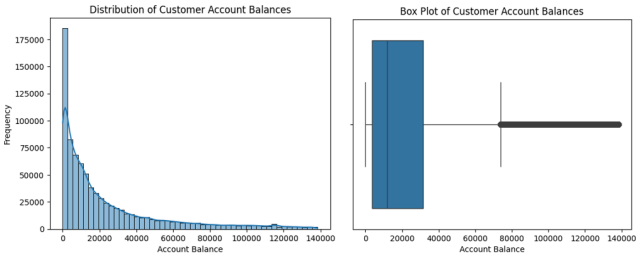

Fig. 1 Distribution of Customer Account Balances and identify any outliers

Figure 2 shows the box plot is to compare the distribution of account balances between different genders, helping identify any disparities. The distributions for Female and Male categories show that while most customers have account balances within a similar range, there are a few with much higher balances. The Unknown category, on the other hand, mostly consists of lower account balances, with very few high-balance accounts.

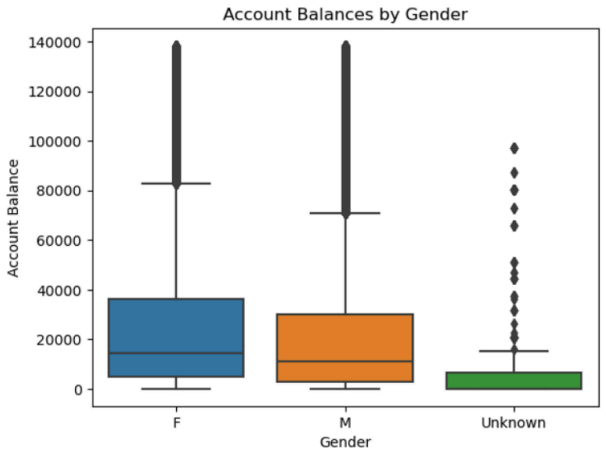

Fig. 2 Differences in Account Balances Across Different Genders

Figure 3 depicts the relationship between account balances and transaction amounts for customers. Despite the wide range of account balances and transaction amounts, there isn't a discernible linear relationship where higher account balances correlate with higher transaction amounts. This means that having a higher balance does not necessarily lead to larger transactions.

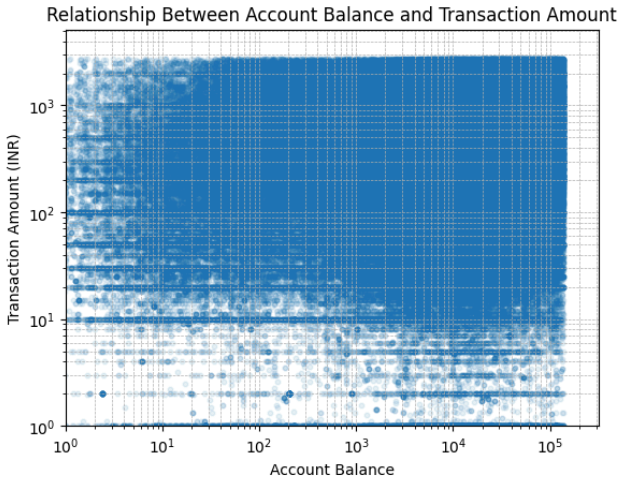

Fig. 3 Relationship Between Account Balance and Transaction Amount

Figure 4 is to explore if there is a correlation between the age of customers and their account balances. This histogram shows the age distribution of the customer base. Most customers are in their 20s and 30s, as shown by the peak in the histogram. The frequency gradually decreases for older age groups. The distribution is right-skewed, meaning there are more younger customers than older ones.The largest age group among the customers is the young adult demographic, highlighting a potentially tech-savvy or younger market segment.While there are customers in the older age ranges, they make up a smaller portion of the overall customer base.

This scatter plot illustrates how account balances vary across different ages. Despite the spread, there is no clear upward or downward trend connecting age with account balance. This indicates that age may not be a determining factor in the size of account balances.

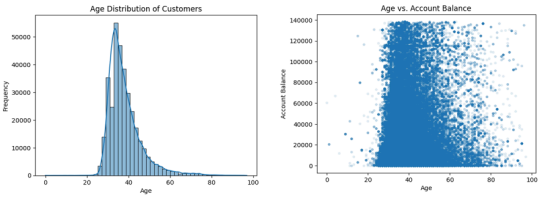

Fig. 4 Age Demographics and Their Relation to Account Balance

Figure 5 is the distribution of transactions throughout the week and across different hours of the day. The bar chart shows that transaction activity is higher on weekdays, with Tuesday being the peak day for transactions, followed closely by Monday and Wednesday.

There is a gradual increase in transactions starting from around 8 AM, with activity steadily rising as the day progresses. Transaction activity peaks in the late afternoon and evening, between 1 PM and 8 PM, with the highest number of transactions typically occurring around 6 PM to 7 PM. This could reflect people conducting transactions after work hours.

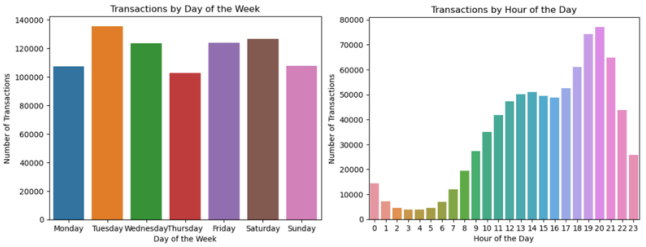

Fig. 5 Transaction Activity Over Time

D. Algorithms used

1) *StandardScaler:* StandardScaler is a preprocessing technique in machine learning. Its role is to standardize features by removing the mean and scaling to unit variance. This process ensures that the features are on a similar scale, which can be essential for certain algorithms to perform optimally.

The idea behind StandardScaler is it transforms the data such that its distribution has a mean of 0 and a standard

deviation of 1. This transformation doesn't change the shape of the distribution but rather places it in a standard form that is easier to work with for many machine learning algorithms. The formula of StandardScaler is as follows:

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

Where x is the individual value in datasets, μ is the mean of datasets, and σ is the standard deviation of datasets.

1) *Silhouette Score:* Silhouette score is a metric used to evaluate the quality of clusters formed by a clustering algorithm. It quantifies how well each data point fits within its assigned cluster, providing a measure of cluster cohesion and separation. A higher silhouette score indicates that the data points are well-clustered, with tight clusters and good separation between them, while a lower score suggests that the clusters may be overlapping or poorly defined [8]. The formula of Silhouette Score is as follows:

$$s(i) = \frac{b(i)-a(i)}{max\{a(i),b(i)\}} \quad (2)$$

It ranges from -1 to 1. A score close to 1 indicates that the data point is well-clustered, with a high degree of cohesion within its cluster and good separation from neighboring clusters. A score close to -1 suggests that the data point may have been assigned to the wrong cluster, as it is closer to points in another cluster than to those in its own cluster. A score around 0 indicates that the data point is close to the boundary between clusters. To obtain the overall silhouette score for a clustering solution, you take the average of the silhouette scores for all data points. The overall silhouette score ranges from -1 to 1, with higher values indicating better clustering quality.

The silhouette score provides a quantitative measure of how well a clustering algorithm has partitioned the data into clusters. It is particularly useful for comparing different clustering solutions or selecting the optimal number of clusters for a dataset.

2) *Principal Component Analysis (PCA):* Principal Component Analysis (PCA) is a dimensionality reduction technique widely used in data analysis and machine learning. Its primary objective is to transform high-dimensional data into a lower-dimensional space while retaining as much of the original information as possible. PCA achieves this by identifying the directions, called principal components, along which the data varies the most. These principal components are orthogonal to each other, meaning they capture different aspects of the data's variability. By selecting a subset of the principal components that explain the most variance, PCA effectively reduces the dimensionality of the dataset. This reduction can lead to more efficient computation, visualization, and improved performance in downstream tasks such as clustering or classification. PCA is particularly valuable when dealing with high-dimensional datasets or when seeking to understand the underlying structure of the data [14].

VI. RESULTS AND DISCUSSIONS

1. *K-Means*

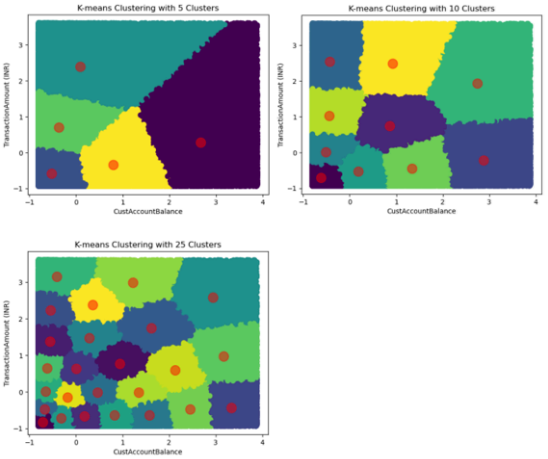

Fig. 6 Visualization of K-Means Clustering

The figure above shows the results of K-means clustering with different numbers of clusters. From left to right, the plots represent clustering with 5, 10, and 25 clusters respectively. Each plot has a two-dimensional space with 'CustAccountBalance' on the x-axis and 'TransactionAmount (Mln)' on the y-axis.

In the first plot with 5 clusters, the data points are grouped into clearly defined, large regions of different colors. The centroids of these clusters are marked with red dots, indicating the average position of the data points within each cluster.

The second plot with 10 clusters shows a more segmented space with smaller and more numerous colored regions, again with the red dots marking the centroids of each cluster. This segmentation suggests a finer granularity in the data grouping compared to the first plot. However, there are some regions where the clustering boundaries seem jagged, especially between the yellow and green clusters, and the blue and purple clusters. This might suggest that data points near these boundaries were closer in distance to multiple centroids, making the assignment less clear-cut.

The third plot with 25 clusters presents a highly detailed clustering, with many small regions each having a centroid. The distinction between some of the clusters appears more nuanced, reflecting a very detailed data categorization.

The presence of small 'islands' of one cluster within the territory of another (for example, the yellow patches within the blue and purple areas) indicates that some data points are more similar to a centroid that's not geographically the nearest. This can happen in cases where the K-means algorithm's assumption of spherical clusters with similar sizes does not fit the actual distribution of data [16].

Additionally, we observe that some centroids (the red dots) are not centrally located within their clusters. In an ideal scenario, each centroid would be at the center of a cluster, indicating a clear mean of the cluster's features. The non-central positions of some centroids might suggest that the clusters are skewed or that there are outliers pulling the mean away from the center.

2. *DBSCAN*

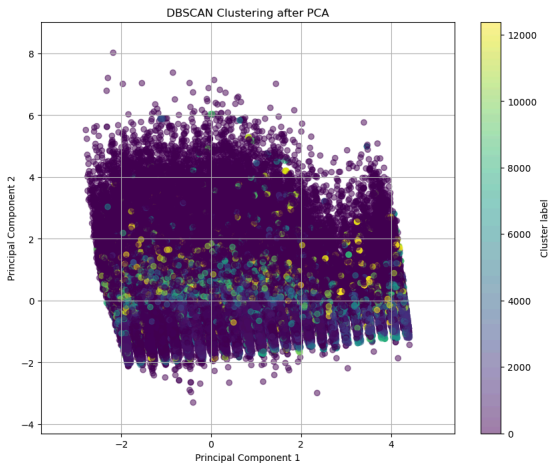

Fig. 7 Visualization of DBSCAN

The figure above shows the result of DBSCAN clustering applied to data that has undergone Principal Component Analysis (PCA). The colors of the points indicate different clusters found by DBSCAN, with the color scale reflecting the cluster label assigned to each point. The clusters are not as clearly distinct as you might see with K-means, because DBSCAN does not force every point into a cluster; it allows for the possibility of noise, which is represented by points that are not assigned to any cluster.

A noticeable feature of this plot is that many data points are concentrated along certain horizontal lines. This could be an artifact of the PCA transformation, or it might reflect some intrinsic property of the original data.

The majority of data points are colored with the darker end of the spectrum, suggesting that DBSCAN may have identified these points as belonging to one large cluster or possibly as noise (it depends on the labeling convention used). The lighter-colored points indicate separate, smaller clusters. The distribution and size of clusters suggest that there might be some underlying structure in the data, but the clusters are not as well-separated as might be desired for clear categorization [11].

3. *GMM*

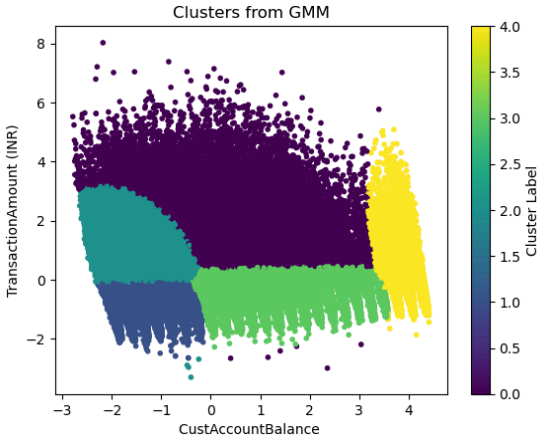

Fig. 8 Visualization of Gaussian Mixture Models

The graph presents results from clustering using a Gaussian Mixture Model (GMM). It shows a scatter plot where each dot represents a data point in the space defined by two features, labeled 'CustAccountBalance' on the x-axis and 'TransactionAmount (INR)' on the y-axis. The color of each point corresponds to the cluster that the GMM algorithm has assigned it to, as indicated by the color bar on the right, which maps colors to cluster labels.

GMM is especially useful for identifying clusters that are not necessarily spherical and can have different variances. This capability is evident in the plot where we see that clusters appear to have different shapes and densities:

- The yellow cluster on the right appears to have a fairly elongated shape, stretching along the 'First Feature' axis.

- The purple cluster at the top of the plot is more diffuse, spreading out over a large range of the 'Second Feature'.

- The blue and green clusters at the bottom seem more compact and denser than the others.

These characteristics suggest that the underlying data distributions have different variances and possibly different covariance structures, which GMM can accommodate [12]. This is in contrast to K-means, which assumes that clusters are spherical and have similar variances.

For Hierarchical Clustering and Spectral Clustering, we will use a subsample of the entire dataset with n = 10000 to reduce the use of memory. So the diagram will be different compared with the previous one every time I run the code.

4. *Hierarchical Clustering*

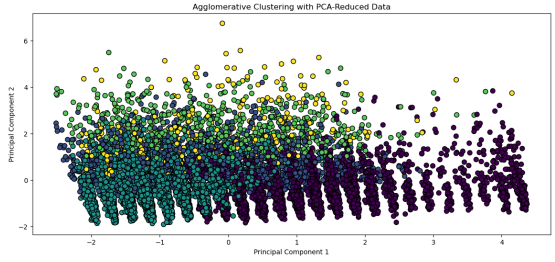

Fig. 9 Visualization of Agglomerative Clustering with PCA-Reduced

The image appears to be a scatter plot showing the results of agglomerative clustering on a dataset that has been processed with Principal Component Analysis (PCA) with Silhouette Score for Agglomerative Clustering is 0.2789191. Here's a comprehensive interpretation of what the graph illustrates:

The x-axis and y-axis represent the first two principal components, which are the results of PCA. This technique reduces the dimensionality of the data, typically retaining the components that capture the most variance [14]. The two axes show the new feature space with reduced dimensions where each point is a data sample.

The clusters seem to spread along both principal components, suggesting that the variance captured by PCA contains meaningful information for clustering.

Some points are scattered far from the main clusters, which could be considered outliers. Their presence could affect the clustering process, particularly if these points are significantly different from the others [10].

This graph is useful for visualizing the natural groupings in the data after reducing the complexity with PCA. Each cluster might represent different underlying phenomena or types of behavior within the dataset, and understanding the original variables that contribute to the principal components would be crucial for interpreting these clusters in a real-world context.

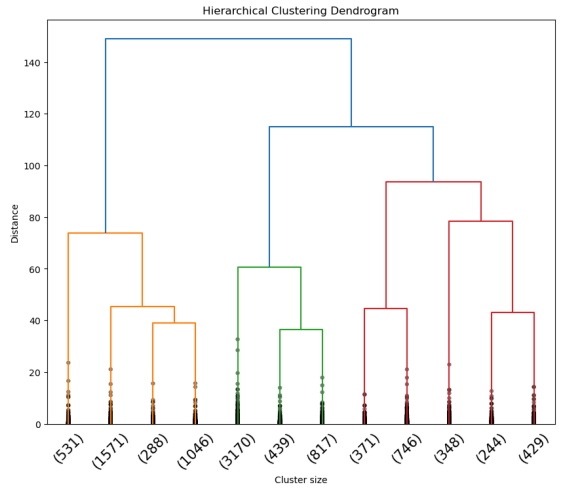

Fig. 10 Visualization of Hierarchical Clustering

In the dendrogram, the y-axis represents the distance or dissimilarity between clusters, where the 'distance' can refer to the Euclidean distance or another distance metric used by the clustering algorithm. The x-axis has labels indicating the size of the clusters at the points where the tree is cut.

The bottom of the dendrogram (the leaves) represents individual data points or clusters of a single element. As you move up the tree, these leaves begin to merge. Each horizontal line indicates a merge between two clusters or data points. The height of the horizontal line shows the distance at which the two clusters were joined. A higher line means that clusters are less similar to each other. The numbers at the bottom indicate the number of data points in the cluster at the point where the dendrogram is cut horizontally. For instance, '(407)' means there is a cluster consisting of 407 data points. The colors (blue, orange, green, red) may represent the sequence of cluster formations or might be arbitrary. Sometimes colors are used to indicate different clusters at a particular threshold. By cutting the dendrogram at a specific height, you determine the number of clusters. For example, cutting it at a height of 70 might yield six clusters, as suggested by the color coding.

Interpreting this dendrogram suggests several clusters of varying sizes, with the largest distance merge happening above the 120 mark on the y-axis, indicating a significant dissimilarity between two major clusters in this hierarchy.

5. *Spectral Clustering*

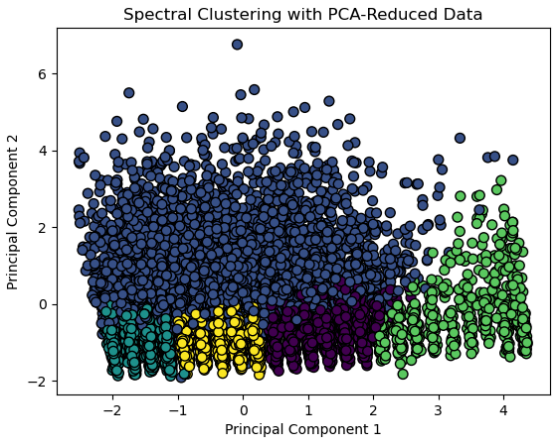

Fig. 11 Visualization of Spectral Clustering with PCA-Reduced

The image shows a scatter plot resulting from spectral clustering applied to data that has been dimensionally reduced via PCA (Principal Component Analysis) with Silhouette Score for Spectral Clustering: 0.075306.

This clustering algorithm is used to group similar points together. It works well with data where the clusters are not necessarily spherical and might have complex shapes. It uses the eigenvalues of a similarity matrix of the data to reduce dimensions before clustering in fewer dimensions [13]. The points are color-coded based on the cluster to which they've been assigned by the spectral clustering algorithm. The colors indicate that the algorithm has identified several distinct groups within the data. We can see that the clusters seem to be spread across both principal components, with no cluster being restricted to a narrow region of the PCA-reduced feature space. This suggests that the original high-dimensional data is fairly well-spread out across its principal axes of variance. There is some overlap between clusters in certain regions, particularly where blue and green points are close to each other. However, the yellow cluster seems to be more distinctly separated from the blue cluster, which might indicate a clearer distinction in the data's underlying structure for that cluster. There are a few points that are distant from the main body of data points, which could be considered outliers. These are the points that are farther away along the 'Principal Component 2' axis.

VII. CONCLUSION

In conclusion, leveraging advanced unsupervised learning techniques for customer segmentation can significantly enhance the capabilities of banks and financial institutions in managing and understanding their diverse customer bases. Each of the discussed techniques—K-Means Clustering, Hierarchical Clustering, DBSCAN, Gaussian Mixture Models, and Spectral Clustering—brings unique advantages that can be applied to various aspects of customer data analysis.

K-Means Clustering is ideal for straightforward, large-scale segmentation tasks, providing clear groupings based on defined criteria, which can be highly effective in targeted marketing and risk categorization. Hierarchical Clustering offers a more nuanced exploration of data structure through its tree-like representation, making it suitable for detailed customer lifecycle analysis and behavioral segmentation. DBSCAN excels in identifying outliers and handling noise, making it invaluable for fraud detection and ensuring the robustness of the segmentation process in diverse datasets. Gaussian Mixture Models provide probabilistic insights into customer groupings, offering a deeper understanding of customer behaviors and more personalized customer service based on the likelihood of customers' actions. Spectral Clustering addresses complex and non-linear structures in data, perfect for applications like social network analysis and intricate pattern detection that are beyond the reach of traditional clustering methods.

The strategic implementation of these clustering techniques allows banks to not only tailor their services and products more effectively but also to enhance risk management and compliance measures. By understanding and segmenting their customers more accurately, banks can improve operational efficiency, customer satisfaction, and ultimately, financial performance in the competitive market. The choice of technique will depend on the specific needs and characteristics of the data, highlighting the importance of a tailored approach in utilizing these advanced analytical methods.

REFERENCES

[1] E. Y. L. Nandapala and K. P. N. Jayasena, “The practical approach in Customers segmentation by using the K-Means Algorithm,” 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), Nov. 2020, doi: https://doi.org/10.1109/iciis51140.2020.9342639.

[2] Sakshi Priyadarshni, Rakshan Fathima, Siddhaling Urolagin, A. M. Bongale, and Deepak Sudhakar Dharrao, “Unveiling Customer Segmentation Patterns in Credit Card Data using K-Means Clustering: A Machine Learning Approach,” Dec. 2023, doi: https://doi.org/10.1109/mosicom59118.2023.10458783.

[3] Saumya Parag Phadkar, Chinmay Singhania, S. Poddar, Jai Suryawanshi, and Swati Chandurkar, “Customer Segmentation for E-Commerce Using Recency Frequency Monetary and Hierarchical Clustering,” Aug. 2023, doi: https://doi.org/10.1109/iccubea58933.2023.10392053.

[4] Z. S. Al-Sudani and Musaab Riyadh, “Fraudulent Taxi Driver detection in Baghdad City based on DBSCAN and A* Algorithm,” Jul. 2023, doi: https://doi.org/10.1109/aiccit57614.2023.10217860.

[5] Y. Li, Y. Zhan, and X. Wang, “A community detection algorithm based on multi-domain adaptive spectral clustering,” Oct. 2016, doi: https://doi.org/10.1109/imcec.2016.7867421.

[6] B. Lian, H. Chen, C. Wu, and M. Chen, “Fast Spectral Clustering algorithm based on wavelet basis decomposition,” 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Jun. 2020, doi: https://doi.org/10.1109/itnec48623.2020.9084950.

[7] F. I. Mahlidah, A. K. Sukarno, Y. Yustiawan, and M. D. R. Bakry, “Human-Centered Machine Learning Implementation in Banking: Case Study in BRILink (BRI Branchless Banking) Agent Acquisition, Upgrade, and Activation,” *2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Dec. 2022, doi: https://doi.org/10.1109/ieem55944.2022.9989784.

[8] K. R. Shahapure and C. Nicholas, “Cluster Quality Analysis Using Silhouette Score,” *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, Oct. 2020, doi: https://doi.org/10.1109/dsaa49011.2020.00096.

[9] S. Shah and M. Singh, “Comparison of a Time Efficient Modified K-mean Algorithm with K-Mean and K-Medoid Algorithm,” *2012 International Conference on Communication Systems and Network Technologies*, May 2012, doi: https://doi.org/10.1109/csnt.2012.100.

[10] Z. Nazari, D. Kang, M. R. Asharif, Y. Sung, and S. Ogawa, “A new hierarchical clustering algorithm,” *2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Nov. 2015, doi: https://doi.org/10.1109/iciibms.2015.7439517.

[11] D. Deng, “DBSCAN Clustering Algorithm Based on Density,” *IEEE Xplore*, Sep. 01, 2020. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9356727 (accessed Mar. 24, 2022).

[12] W. Lin, “An improved GMM-based clustering algorithm for efficient speaker identification,” Dec. 2015, doi: https://doi.org/10.1109/iccsnt.2015.7491011.

[13] D. Huang, C.-D. Wang, J.-S. Wu, J.-H. Lai, and C.-K. Kwoh, “Ultra-Scalable Spectral Clustering and Ensemble Clustering,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1212–1226, Jun. 2020, doi: https://doi.org/10.1109/tkde.2019.2903410.

[14] A. Rehman, A. Khan, M. Ali, Muhammad Umair Khan, Shafqat Ullah Khan, and L. Ali, “Performance Analysis of PCA, Sparse PCA, Kernel PCA and Incremental PCA Algorithms for Heart Failure Prediction,” 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Jun. 2020, doi: https://doi.org/10.1109/icecce49384.2020.9179199.

[15] Mustafa Raza Rabbani, M. Selim, Salma Ayman, Zainab Sayed Mohamed, A. Hakeem, and Sarah Khalid Ahmed, “Factors determining the financing decision of Islamic banks in Bahrain - An empirical analysis,” Oct. 2022, doi: https://doi.org/10.1109/sibf56821.2022.9939838.

[16] M. Aryuni, E. Didik Madyatmadja, and E. Miranda, “Customer Segmentation in XYZ Bank Using K-Means and K-Medoids Clustering,” *2018 International Conference on Information Management and Technology (ICIMTech)*, Sep. 2018, doi: https://doi.org/10.1109/icimtech.2018.8528086.

[17] X. Vasques, “Concepts, Libraries, and Essential Tools in Machine Learning and Deep Learning,” pp. 1–33, Jan. 2024, doi: https://doi.org/10.1002/9781394220649.ch1.

[18] J. Mange, “Effect of Training Data Order for Machine Learning,” Dec. 2019, doi: https://doi.org/10.1109/csci49370.2019.00078.

[19] P. Patel, B. Sivaiah, and R. Patel, “Approaches for finding Optimal Number of Clusters using K-Means and Agglomerative Hierarchical Clustering Techniques,” IEEE Xplore, Jul. 01, 2022, doi: https://ieeexplore.ieee.org/abstract/document/9862439/

[20] A. Mirzal, “Statistical Analysis of Microarray Data Clustering using NMF, Spectral Clustering, Kmeans, and GMM,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, pp. 1–1, 2020, doi: https://doi.org/10.1109/tcbb.2020.3025486.