2021A7PS0040H-2021A7PS0372H-2021AAPS0639H (1).pdf

by Vamsi Krishna Gattupalli

Submission date: 02-May-2024 12:21AM (UTC+0530)
Submission ID: 2368063295
File name: 2021A7PS0040H-2021A7PS0372H-2021AAPS0639H_1_.pdf (1.99M)
Word count: 4887
Character count: 26756

Credit Card Customer Clustering

BITS Pilani Hyderabad Campus
CS F415 Data Mining Project

Vamsi Krishna Gattupalli f20210040@hyderabad.bits-pilani.ac.in
Rohit Reddy Daareddy f20210372@hyderabad.bits-pilani.ac.in
Sri Vishnu Patchava f20210639@hyderabad.bits-pilani.ac.in

1 Abstract

In this research, our objective is to categorize credit card customers by examining a range of details pertaining to their expenditure patterns and credit information. The chosen problem presents an intriguing opportunity, as it provides valuable insights into the identification of customer groups characterized by comparable financial conditions and spending behaviors; subsequent clustering enables organizations to discern clusters at heightened risk of credit default, thereby empowering them to proactively mitigate potential losses through targeted credit restrictions for clusters harboring customers with a history of defaults. In this paper, three different clustering algorithms were implemented to cluster customers based on the credit cards dataset. The algorithms were then compared based on their scores in terms of clustering performance metrics like silhouette coefficient. The outcome that we observed was DBSCAN was performing the best as per silhouette coefficient, and K-Means was performing the best as per Calinski–Harabasz index and Davies-Bouldin index.

Keywords: Data Mining, Clustering, Credit Card Customers, Unsupervised Learning, Customer Segmentation

2 Introduction

The primary problem addressed in this research revolves around the task of effectively clustering credit card customers based on various attributes, with the objective of managing risk of credit defaults within the financial sector. By segmenting customers according to their financial behaviors and spending patterns, organizations stand to gain invaluable insights into consumer preferences and tendencies, enabling them to vary credit limit as per the behaviour of other customers in their cluster. Additionally, the ability to identify clusters of customers at higher risk of defaulting on credit amounts helps organizations to implement measures to mitigate potential losses, thereby ensuring a more sustainable credit portfolio.

The task of comparing and contrasting different clustering algorithms within the context of credit card customer segmentation is complex and interesting one, primarily due to the complexity of the clustering process itself. Unlike traditional classification tasks where the target variable is known and predefined, clustering involves grouping data points solely based on their similarities or dissimilarities, making the evaluation process complex as well as interesting. Devising a robust methodology for comparing clustering algorithms while accounting for these complexities is an important task at hand as it helps organizations improve their risk management and revenue in the form of interest on credit amount.

Prior attempts to address similar challenges in credit card customer segmentation have mainly relied on applying a single clustering algorithm without systematically exploring various other methodologies or evaluating their comparative performance. Even though this approach is simple, it often overlooks the potential increase in the efficiency that can be obtained by applying diverse algorithms and picking the best from them. Consequently, there exists a clear gap in the literature regarding comprehensive

analysis of clustering algorithms within the context of credit card customer segmentation, underscoring the significance of our proposed approach.

In response to these challenges and gaps in the existing literature, our research adopts a methodology that encompasses several key components, beginning with extensive preprocessing on the dataset. This includes starting off with Exploratory Data Analysis (EDA) to identify patterns and anomalies, handling missing values through imputation or deletion, and applying robust scaling to normalize the data and avoid the impact of outliers. Additionally, we employed visualization techniques to identify feature correlations and assess the suitability of the dataset for clustering analysis. Thereafter, we used Principal Component Analysis (PCA) to reduce dimensionality and enhance computational efficiency, thereby facilitating more effective clustering.

Finally, we applyied three distinct clustering algorithms - Hierarchical Clustering, DBSCAN and Hierarchical Clustering - to the preprocessed dataset and carefully evaluated their performance using a range of clustering metrics like Silhouette score, Davies–Bouldin index, and Calinski–Harabasz index. These provided very good insights into the clustering quality and algorithmic efficiency. Through this rigorous efficiency testing, we came to a conclusion that DBSCAN performs best as per their silhoutte scores, K-Means performs best with respect to the Calinski–Harabasz index as well as the Davies–Bouldin index.

3 Related Work

Hierarchical clustering algorithms have been evaluated for document datasets in the study conducted by Ying Zhao et al. (2002). This research paper emphasized the importance of fast and high-quality clustering techniques for intuitive navigation and browsing mechanisms. In this research publication, the results of Hierarchical clustering algorithms on the dataset with their efficiencies were put forth.

In the domain of image data clustering, Vinod Kumar Dehariya et al. (2010) explored the application of K-Means and Fuzzy K-Means algorithms. The research highlighted the significance of image segmentation in clustering for efficient processing, particularly in the context of image databases. Both standard K-Means and Fuzzy K-Means algorithms were investigated, with the latter demonstrating improved segmentation results, especially in complex image analysis scenarios.

Additionally, the study by Dingsheng Deng et al. (2020) focused on the DBSCAN clustering algorithm based on density, addressing the challenges posed by big data analysis. The research reiterated the effectiveness of DBSCAN in clustering arbitrarily shaped datasets with unknown distributions. It showcased the algorithm's superiority in performance in personalized clustering tasks on non-uniform density data sets, showing its potential in diverse clustering scenarios.

However, despite the insights provided by each of these research papers, a common limitation that was observed was: the absence of direct comparisons between the performance of the evaluated algorithms and alternative clustering methods. Although these studies offer in-depth evaluations of individual algorithms within their own contexts, the lack of comparative analyses leaves a gap in understanding the merits and demerits of each of the clustering algorithms. To address this limitation, our research helps by systematically comparing the performance of three clustering algorithms on a single dataset of credit card customers. By taking up such an approach, we aim to provide an assessment of the efficiency of three different clustering algorithms on the dataset we have picked.

4 Approach/Methodology

Marketing is crucial for businesses to thrive, enhancing brand awareness, customer engagement, and sales through promotional strategies. Regardless of industry, leveraging marketing benefits can significantly expand a business's reach and solidify its market position. Customer segmentation is a pivotal method for marketing teams to deeply understand their customer base. It involves dividing customers

into distinct groups based on shared characteristics like demographics, behaviors, or psychographics. This allows businesses to tailor marketing efforts effectively, resonating with specific segments and delivering targeted messages aligned with their unique needs and preferences.

The competitiveness in financial industries are getting harder in the next decade. One of this industry main source of revenue are Interest Income which they could get by giving loan or credit payment facilities to customer. Therefore, the more the credit are given, the more interest they get.

To effectively solve the customer segmentation problem, access to comprehensive customer data is paramount. This typically includes transactional data like installment purchases,cash advances,purchase transactions made and their frequencies.Demographic information, behavioral patterns, and any available psychographic data are important in assessment of the kind of correlations between the above parameters and our end goal.Acquiring and integrating these diverse data sources is crucial for building a holistic understanding of the customer base. Data acquisition can be achieved through various means, including internal databases, third-party data providers, or web scraping techniques, while adhering to ethical and legal guidelines.We have obtained our dataset from Kaggle consisting of 9000 records.

In this research paper to draw a critical role in facilitating accurate and insightful customer segmentation we have employed advanced analytics and machine learning on vast customer data, they can uncover intricate patterns and relationships, enabling more precise and granular customer segmentation for effective, personalized marketing strategies.

Discuss the dataset(s) and its properties. The dataset contains summary of the usage behavior of about 9000 credit users.Whilst we have done the exploratory data analysis, the results were as follows: 95 percent of user have credit limit below 13000 with balance keep below 8000. We assume this is the general population of the data reside. Some feature like purchases, oneoff purchases, INSTALLMENT purchases, and cash advance show the same trends as balance and credit limit. balance frequency for 86 percent of users are updated frequently. We assume this occured due to the balance updated when the purchase made, installment paid, withdrawal, deposit, and cash advance used. There are 2 majority group of customer that made purchase, which is the never or rarely made any purchase and the often one. This could be explained more in after clustering. There are more user that paid using installment rather than oneoff payment. We assume the data was obtained from the bank that have high selling points on its credit card facilities. By its frequency, there are more user that purchase by installment rather than one off payment.

We have implemented and performed comparative analyses of three clustering algorithms. The details of these algorithms are as follows:

4.1 Hierarchical clustering

Hierarchical clustering is a type of unsupervised learning technique used for grouping similar data points in clusters, it is a method of cluster analysis that seeks to build a hierarchy of clusters. Hierarchical clustering is a connectivity-based clustering model that groups the data points together that are close to each other based on the measure of similarity or distance. It is assumed that the data points that are close to each other are more similar or related than data points that are farther apart. Unlike other methods, we do not require to specify the number of clusters to be generated beforehand. By using the hierarchical clustering method we can quantitatively estimate the relation between every sample in the data and study how close each data point is related to each other in the data, which is done by using a dendrogram. A dendrogram is also used to find out the number of clusters to be generated by creating a tree of clusters, where each leaf of the tree represents a single data point, and the root of the tree represents the entire dataset. It explains the relationship between each point in the dataset.

Unlike a regular tree structure, a dendrogram does not always branch out at regular intervals from top to bottom in the vertical direction, which is the y-axis. This irregular branching represents the distance between clusters pictorically, giving a better insight into the relation between the sample points.

The dendrogram is generated for the dataset by iteratively splitting clusters based on a measure of similarity or distance between data points. By traversing down, the tree branches the data into smaller and smaller clusters, this results in the leaf nodes of the dendrogram being the individual data samples of the dataset. In contrast, while traversing upwards, we are combining smaller clusters into bigger clusters till we get one big cluster, which is the dataset.

At the top of the dendrogram the root is the complete set of data, while the bottom leaves are individual data points. To be able to generate different numbers of clusters, the slice of the dendrogram at different levels or heights can result in the clustering of the dataset with different numbers of clusters. There are two types of hierarchical clustering algorithms, which are Agglomerative Clustering and Divisive clustering. In this approach for clustering, we will be using Agglomerative Clustering, which is also known bottom-up approach. This clustering algorithm does not require us to prespecify the number of clusters. The process starts by considering each data point as its cluster. Then, it iteratively merges the two most similar clusters until all data points are in a single cluster.

1. Start with each data point in the dataset as its cluster.
2. Calculate the distance between all pairs of clusters of the dataset.
    - Compute the distance between each pair of clusters using any of the distance metrics, which include Euclidean distance, Manhattan distance, and cosine similarity. The resulting distances are stored in a distance matrix representing the pairwise distances between clusters.
3. Merge the two closest clusters into one.
    - Identify the two clusters with the smallest distance from the distance matrix.
    - Merge these two clusters to form a single cluster, creating a new cluster hierarchy.
4. Update the distance matrix to take into account the newly formed cluster.
    - Update the distance matrix to reflect the distances between the newly formed cluster and the remaining clusters.
    - Depending on the linkage criterion like single, complete, average, Ward's, recalculate the distances between the new cluster and all other clusters.
5. Repeat steps 2,3 and 4 until only one cluster remains.

4.2 K-Means

K-means clustering is a method for grouping n observations into K clusters.The goal is to minimize the sum of squared distances between the data points and their corresponding cluster centroids, resulting in clusters that are internally homogeneous and distinct from each other. The algorithm is detailed below:

$X = \{x_1, x_2, \ldots, x_n\}$ Input dataset K Number of clusters $C = \{C_1, C_2, \ldots, C_K\}$ Cluster assignments

Initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K$ randomly convergence criterion is not met each data point $x_i \in X$ Assign x_i to the nearest cluster centroid:

$$C_j^{(t)} = C_j^{(t)} \cup \{x_i\} \text{ where } j = \arg\min_k \|x_i - \mu_k^{(t)}\|^2$$

each cluster C_j, $j = 1, 2, \ldots, K$ Recalculate the new cluster centroid:

$$\mu_j^{(t+1)} = \frac{1}{|C_j^{(t)}|} \sum_{x_i \in C_j^{(t)}} x_i$$

return $C = \{C_1, C_2, \ldots, C_K\}$

4.3 DBSCAN

The DBSCAN algorithm is a density-based clustering technique used to discover clusters of arbitrary shape and size in a spatial dataset. Since this is a density based clustering algorithms, high density regions are separated from other high density regions with low density regions. The first reason this algorithm was selected was because for DBSCAN, there is no need for prior knowledge of cluster count. Another reason of selecting DBSCAN was because of its ability to discover clusters of arbitrary shape and size. Also, DBSCAN is better than the other 2 approaches mentioned previously because of its robustness to noise and outliers. Therefore, after doing EDA of the dataset, we felt that this algorithm is good contender for clustering our dataset. The algorithm works as follows:

1. Choose values for ϵ (neighborhood radius) and $minPts$ (minimum points for dense region).
2. For each unvisited point p:
   - If p has at least $minPts$ points in its ϵ-neighborhood, mark p as a core point and create a new cluster.
   - Add all points density-reachable from p to the same cluster.
   - If p is not a core point but density-reachable from another core point, mark p as a border point.
   - If p is not a core point and not density-reachable, mark p as an outlier.
3. Repeat step 2 for all unvisited points.

5 Experiments

5.1 Dataset

In the data preprocessing stage, a thorough **outlier analysis** was conducted to identify and assess the presence of extreme or unusual data points within the customer dataset. We have plotted the Box plots to have a visual interpretation.Surprisingly, the analysis revealed that approximately 58 percent of the data points were classified as outliers. It was decided to retain these outliers in the dataset because removing a substantial portion of the data, in this case could potentially lead to the loss of valuable information.

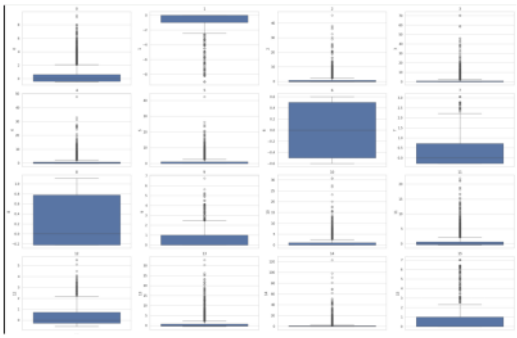

Figure 1: Box Plot for outliers

We have observed that almost all features have skewed distributions.Since skewed data, we need to use Robust Scaling.

We have handled missing Values using *Iterative Imputer* technique.We found that some feature have a missing value, which is "*minimum payments*" amounting to 313 and "*credit limit*" amounting to 1. We decided to handle it by the following :

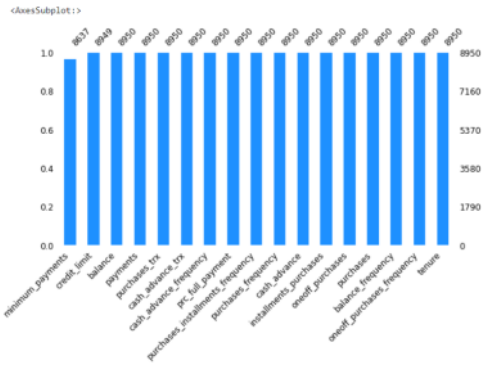

Figure 2: Iterative Imputing

1.Fill the the missing value for Credit Limit that had zero value in payments with 0, since we assume that if the payments is zero, the credit limit should be zero too.
2.Fill the rest of missing value using Simple Imputer with median as its initial strategy.

Since our purpose is clustering (customer segmentation) and only process the clustering using a customer behaviour related feature, we decided to drop several feature.Since the amount contains no specific information we also made a change to *oneoff purchase* to *oneoff proportion* and installments purchase to *installments proportion* for better understanding in the purchase behaviour.

We have dissected what each feature is contributing so as to do the **Feature Subset Selection**.We have gauged it from the correlation matrix and the features with highest correlation were combined. Further we have done data modelling and for that we have used ***Principal Component Analysis***.Since PCA could help to reduce the number of feature for easier interpretation and simplify the complex pattern in modeling.We clustered the new feature with several model like Hierarchial Clustering , K-Means, and DBscan and then compare the result based on some metrics and the number of clusters.

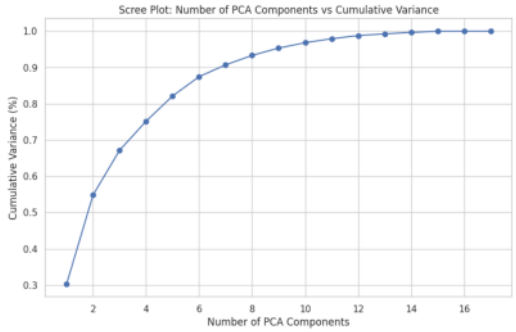

Figure 3: PCA

Final dataset consists of 8947 tuples with 14 features(captured variance of 95 percent) and 58 percent of outliers.The final dataset is consists of no null values with each null value being imputed by mean of corresponding feature.

5.2 Evaluation method / Metrics

To evaluate the clustering models' results, several evaluation metrics can be used: **Silhouette Score** measures how well each data point fits into its assigned cluster compared to other clusters. It ranges from -1 to 1, where a higher score indicates better clustering. A score close to 1 suggests that the data point is well-matched to its cluster and poorly matched to neighboring clusters. A score close to -1 indicates that the data point is likely assigned to the wrong cluster. **Calinski-Harabasz Index** evaluates the ratio of the sum of between-cluster dispersion and inter-cluster dispersion. A higher value of this index indicates better clustering, as it suggests that the clusters are dense and well-separated. **Davies-Bouldin Index** tells us the average similarity of how similar each cluster is, to its most similar cluster. A lower value of the Davies-Bouldin index indicates better clustering, as it suggests that the clusters are not similar to each other.

5.3 Experimental setup

Hierarchical Clustering: Hierarchical clustering is a type of unsupervised learning algorithm that builds a hierarchy of clusters. In general, it can be agglomerative (bottom-up) or divisive (top-down). Agglomerative clustering is more commonly used, where each data point starts as its own cluster, and clusters are merged based on their proximity. In hierearchical clustering, we need to decide upon the linkage to be used (Complete/Single/Average) and the type of distance measure to be used (Euclidean/Manhattan). It is well-suited for applications where the number of clusters is not known in advance. Hierarchical clustering is not suitable for large datasets due to its computational complexity. It is sensitive to outliers and can produce elongated or irregular-shaped clusters.

K-Means Clustering: K-Means is a popular centroid-based clustering algorithm. It requires the number of clusters (K) to be specified in advance. It works by iteratively minimizing the sum of squared distances between data points and their assigned cluster centroids. K-Means is efficient for large datasets and globular clusters. It is sensitive to outliers and can produce suboptimal clusters if the initial centroids are poorly chosen. K-Means tends to split up larger clusters into smaller ones.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise): DBSCAN is a density-based clustering algorithm. It does not require the number of clusters to be specified in advance. It groups together data points that are close to each other based on density reachability. DBSCAN can identify clusters of arbitrary shape and is robust to outliers. It is well-suited for datasets with varying densities and noise. DBSCAN may struggle with clusters of varying densities and requires tuning of the parameters (ϵ and $minPts$).

6 Results and Discussion

6.1 DBSCAN

The DBSCAN algorithm previously mentioned is a general description of DBSCAN algorithm and expects ϵ and $minPts$ as input values. But, in our case, we need to check through a few possible combinations of the 2 values and come to a conclusion as to which values we should use. Therefore, we executed the DBSCAN algorithm over a range of values of ϵ and $minPts$ and compared the values of the Silhouette score, Calinski Harabasz and Davies Bouldin coefficients. We then sorted the scores and checked which pair of parameters gave the optimal score values for each of the 3 scores used.

We then picked the value of $\epsilon = 1.9$ and $minPts = 14$ as this pair of parameters was giving the best values for Silhouette score and moderately good values for Calinski Harabasz index unlike all other order pair values which either gave very good value on just a single coefficient, and bad values on the others.

After deciding the values of ϵ and $minPts$, we ran the algorithm on the dataset and did the clustering, analyzed the clusters formed, drew pair plots of all features with respect to the clusters formed and finally analyzed the percentage of data points per cluster.

The coefficient values we obtained for the DBSCAN algorithm when we used the optimal pair of parameters $\epsilon = 1.9$ and $minPts = 14$ are:

Metric	Value
Silhouette	0.323649
Calinski Harabasz	973.983228
Davies Bouldin	2.419036

Table 1: Cluster Evaluation Metrics

On applying the DBSCAN algorithm, 2 clusters were observed to exist, one marked in yellow and the other marked in dark blue. The clustering of the data points that we obtained is as follows:

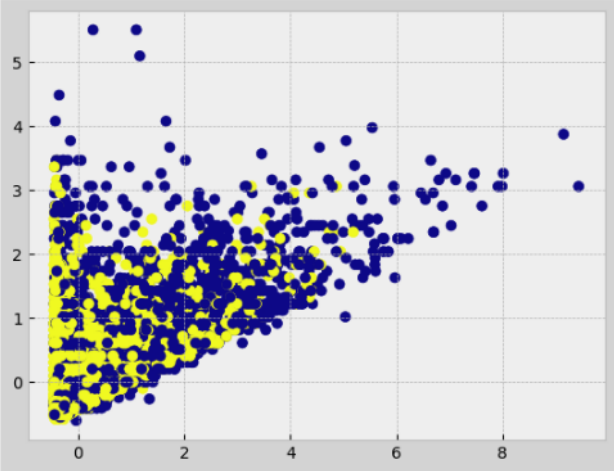

Figure 4: DBSCAN clustering plot

In order to further analyze the number of points in each cluster, we plotted the number and percentage of points in each cluster. It was observed that 2278 points which is around 25.5% of the data points exist in one cluster and that the remaining 6671 points which is 74.5% of the data points exist in another cluster. Which means that each cluster has significant number of points, unlike clustering algorithms that produce clustering where one of the clusters has most of the points and the remaining ones have very few points.

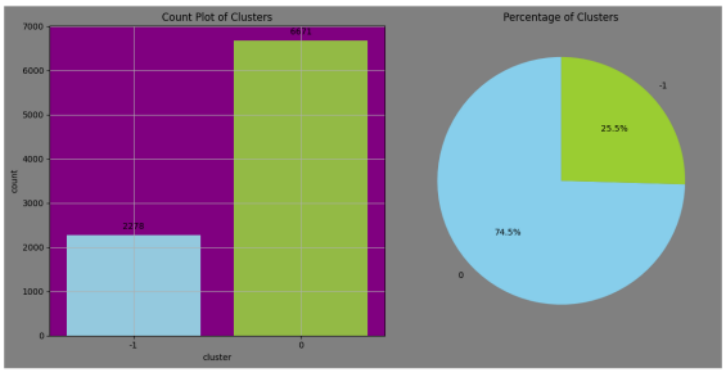

Figure 5: DBSCAN point distribution

Then we proceeded to analyze how the clusters are separated when the data points are plotted between features. Since there are multiple pairs of features that can be picked, we drew the pair

plots for the features and marked the 2 different clusters in different colours to be visually able to differentiate between the two.

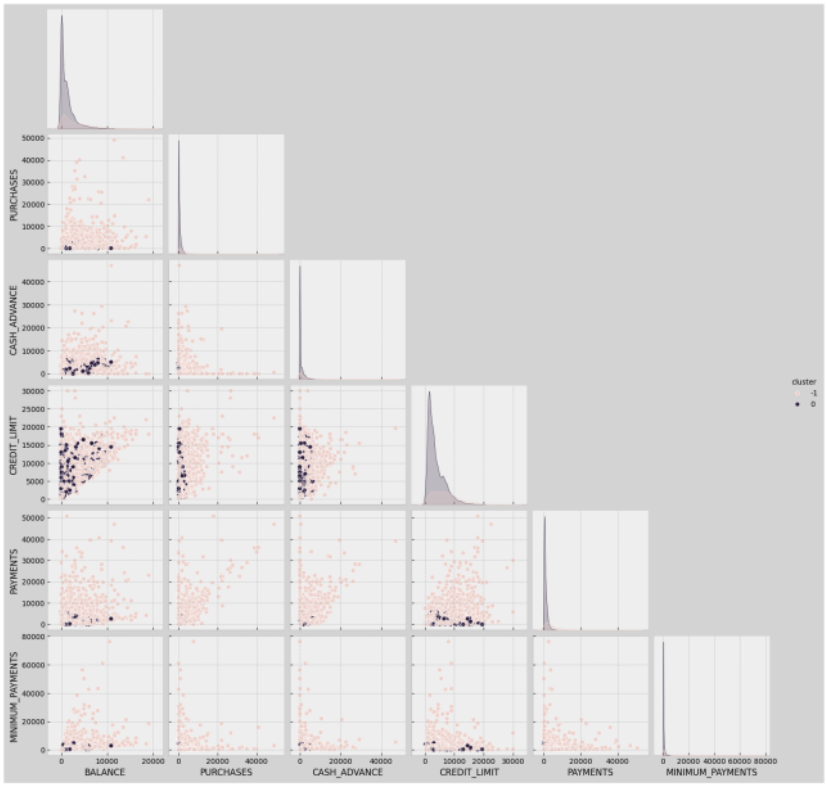

Figure 6: DBSCAN pair plots

6.2 K-Means

The K-means algorithm was run by changing the parameters for better clustering.Despite subjecting it to different fine tuning of parameters the coefficients i.e Silhouette score, Calinski Harabasz and David Bouldin were inert.Therefore we can conclude that changing parameters doesnt affect the result.

The coefficient values obtained for K-means algorithm are:

Metric	Value
Silhouette	0.250
Calinski Harabasz	1604.88
Davies Bouldin	1.597

Table 2: Cluster Evaluation Metrics

1 We will decide the numbers of cluster by using Elbow Method and Silhouette Method. Where in Elbow Method, the number of clusters are decided when the addition of one cluster does not provide significant change in the level of similarity, while in silhouette method, the number of cluster is decided by how close each point in one cluster is to points in the neighboring clusters.

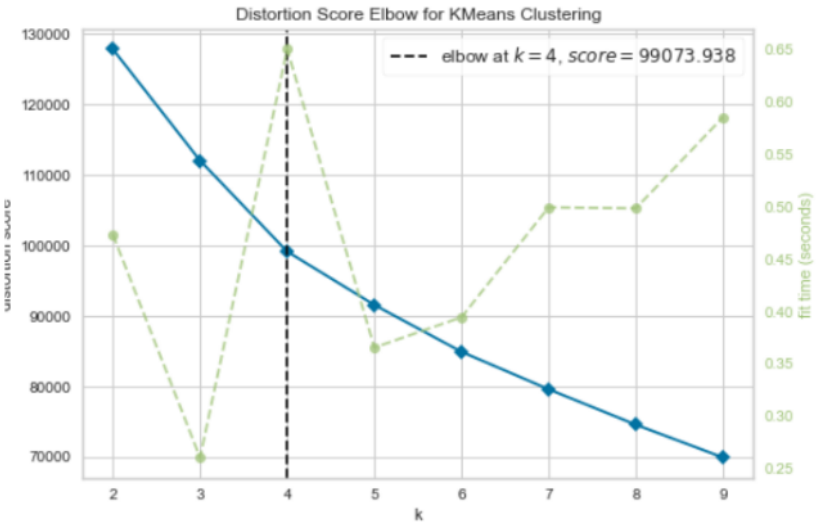

Figure 7: No of optimal clusters=4

Plotting the clusters in 2-D:

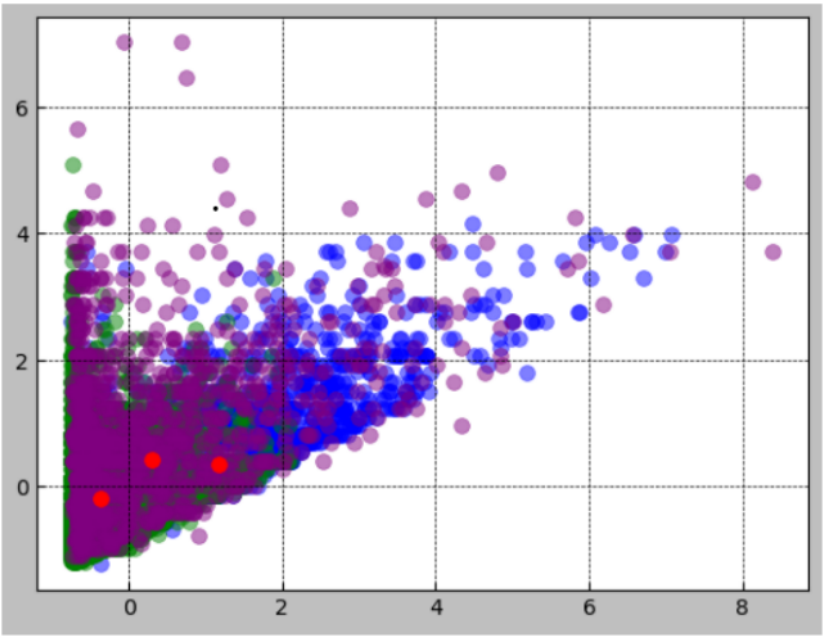

Figure 8: Clusters using K-Means

In order to analyze the same more objectively ,we depicted the percentage of points belonging to each cluster through histogram.It was observed 6104 points belonged to cluster 2 which constitutes around 68.2%.Cluster 1 and cluster 2 constituted 1593 and 1242 points respectively.The clustering is decent since optimal number of points belonged to each clusters.

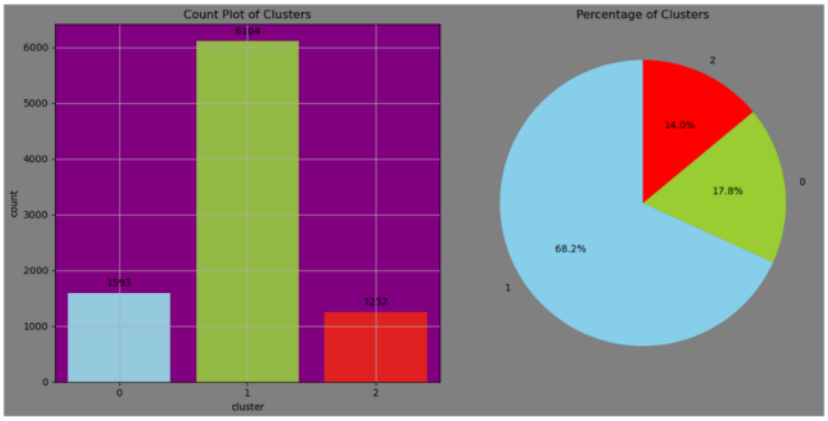

Figure 9: K-Means point distribution

We extended to plot the pairwise plot for the features and indexed the clusters from 0-3 to differentiate. K-means clustering has linear time complexity ,so is better than hierarchial clustering.Objectively it

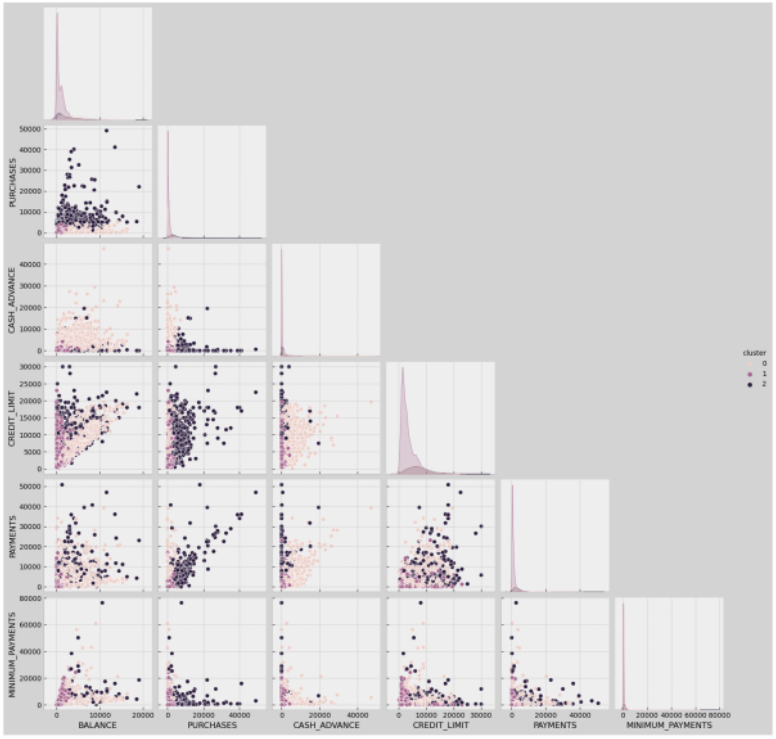

Figure 10: Pair plot K-Means

can handle the larger datasets K-means should perform better than hierarchial clustering.Parallezing is easy in k means clustering and is simple in terms of its algorithm and implementation.

6.3 Hierarchical Clustering

The clustering algorithm can be performed by using different distance measurement techniques and different linkage criteria, each resulting in different results. We loop through each possible combination of these parameters to find the best pair for the given dataset. We decide the best pair of parameters for the formation of the dendrogram by comparing the scores obtained by the three different scoring methods.

By using the above parameters and applying the hierarchical clustering algorithm to the given data set, we get 2 clusters. The majority of the points belong to the cluster indicated by the color blue, the minority being yellow. The dendrogram showing the clustering also conveys the different clusters. The smaller cluster is indicated by blue colour, and the bigger one is in red.

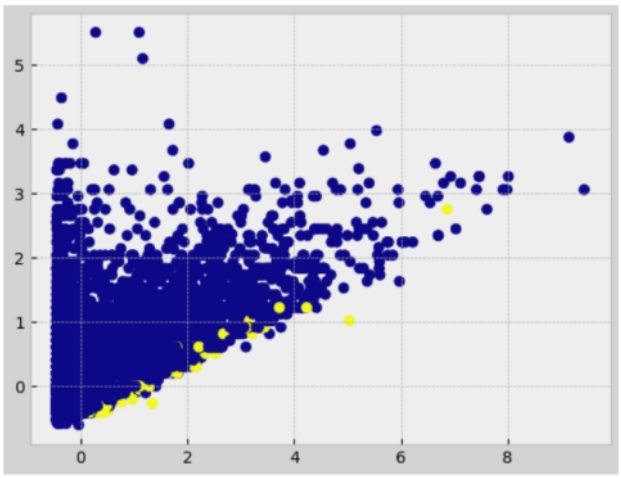

Figure 11: hierarchical clustering plot

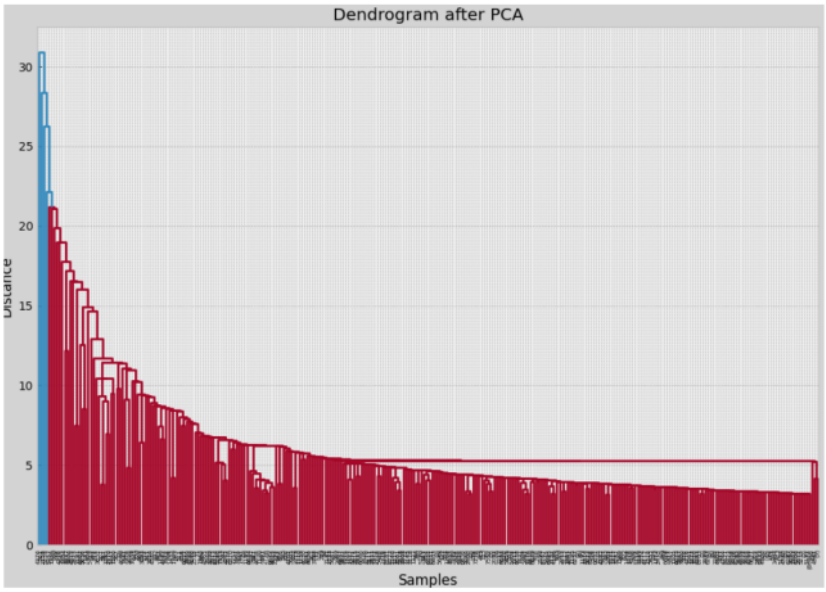

Figure 12: Dendrogram plot

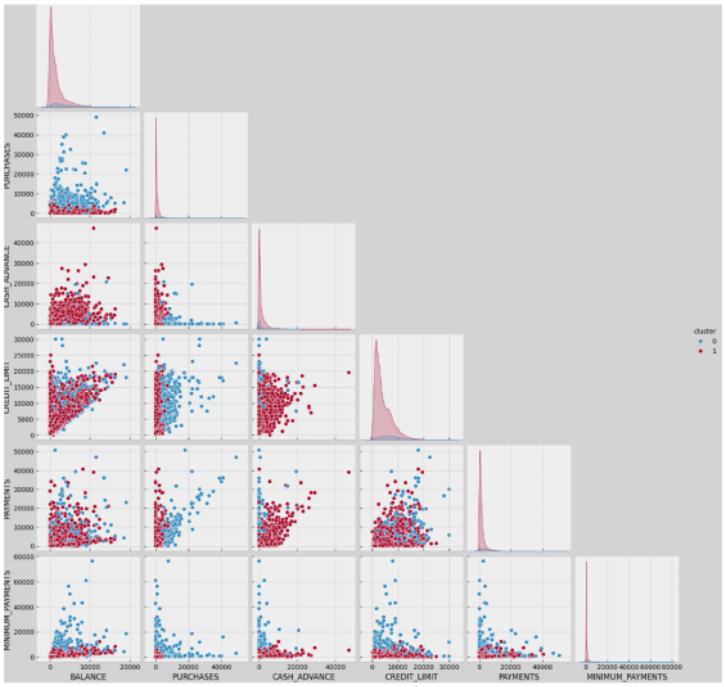

Figure 13: hierarchical clustering scatter plots

To get a detailed analysis of the number of points in each cluster, we plot the percentage split of the dataset and the number of points in each cluster. The majority has a percentage of 91.5% with 8187 points. And the minority has a percentage of 8.5% with 762 points. Although the dendrogram can decide the number of clusters to be formed to best suit the dataset, we only get two clusters which is indicated by the two different colors in the dendrogram plot.

The plots help in analyzing the cluster distribution when the dataset is plotted between two of the features. These plots clearly show the difference in the number of data points between the clusters, also showing how the outliers are mostly in the majority cluster.

6.4 Inferences

Taking into consideration Calinski Harabaz and Davies Bouldin's metrics into consideration, we get that K-Means is the better algorithm of the three. As this is a clustering problem, we have considered that Silhouette coefficients better describe the outcome of a clustering problem. Following this, we get that DBSCAN outperforms the other two algorithms.

Also. DBSCAN is good at identifying these outliers as noise because it focuses on dense regions. K-means and hierarchical clustering can be sensitive to outliers, potentially distorting the cluster centers or forcing outliers into inappropriate clusters. at identifying these outliers as noise because it focuses on dense regions. K-means and hierarchical clustering can be sensitive to outliers, potentially distorting the cluster centers or forcing outliers into inappropriate clusters. Spending patterns can vary greatly, leading to clusters with irregular shapes that DBSCAN can capture effectively. Unlike K-means, which requires specifying the number of clusters upfront, DBSCAN discovers the number of groups automatically based on data density. This is valuable for credit card data, where the natural customer groupings based on spending habits may be obscure. Hierarchical clustering also doesn't require the number of clusters to be formed upfront, but it is slow and computationally heavy.

DBSCAN handles noise, irregular cluster shapes, or flexible cluster numbers better. Consider K-means for well-separated clusters or hierarchical clustering for exploring spending hierarchies. This makes DBSCAN very scalable, making it the better algorithm for the given dataset.

7 Conclusion & Future Scope

Overall we have done an analysis of identifying different segments in the existing customers based on their spending patterns as well as past interaction with the bank. This was to help resolve queries such as identifying the number of different segments of customers , how are these segments different from each other.This study illustrated how clustering algorithms employed by data mining techniques which include DBSCAN,K-Means and Hierarchial Clustering.Based on the Silhouette coefficients which is the predominant metric for evaluation DBSCAN outperforms the other two clustering algorithms

It can be said that income, education, and fiscal discipline indicate consumers' socio-economic status, which is closely related to their credit worthiness.They can be relied upon to provide valuable insights about consumer behavior and characteristics.We can use large dataset to analysis more of it and we can increase accuracy of it.

8 References

- C. C. Aggarwal, "A Framework for Clustering Evolving Data Streams," in IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 8-17, Jan. 2009, DOI: 10.1109/TNN.2008.2006198.
- P. Chakraborty, S. Das, S. Mitra, and C. A. Murthy, "A Comprehensive Review on Ensemble-Based Techniques for Credit Scoring in Financial Domain," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 10, pp. 4174-4193, Oct. 2021, DOI: 10.1109/TKDE.2021.3083696.
- S. Akram, M. Shafique, A. Khan, and S. Iqbal, "An Intelligent Credit Scoring Model for Small and Medium-Sized Enterprises Using Ensemble Learning," in IEEE Access, vol. 9, pp. 108438-108449, 2021, DOI: 10.1109/ACCESS.2021.3093896.
- W. H. Organization, "Hierarchical Clustering," IEEE Xplore, 2015. [Online]. Available: https://ieeexplore.ieee.org/d April 21, 2024]
- GeeksforGeeks, "Hierarchical Clustering," GeeksforGeeks, Available: https://www.geeksforgeeks.org/hierarchical-clustering/. [Accessed: April 20, 2024].
- Wikipedia. "Hierarchical Clustering." Wikipedia. https://en.wikipedia.org/wiki/Hierarchicalclustering [accessed April 21, 2024]

2021A7PS0040H-2021A7PS0372H-2021AAPS0639H (1).pdf

ORIGINALITY REPORT

21%	18%	11%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	github.com Internet Source	3%
2	fastercapital.com Internet Source	3%
3	origin.geeksforgeeks.org Internet Source	2%
4	Saikat Raj, Santanu Roy, Surajit Jana, Soumyadip Roy, Takaaki Goto, Soumya Sen. "Customer Segmentation Using Credit Card Data Analysis", 2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA), 2023 Publication	2%
5	Farzan Madadizadeh, Sajjad Bahariniya. "Tutorial on Statistical Data Reduction Methods for Exploring Dietary Patterns", Clinical Nutrition ESPEN, 2023 Publication	1%
6	www.inforly.io Internet Source	1%

7	repositori.tecnocampus.cat Internet Source	1 %
8	www.mdpi.com Internet Source	1 %
9	dspace.spbu.ru Internet Source	1 %
10	Nora Oikonomakou. "A Review of Web Document Clustering Approaches", Data Mining and Knowledge Discovery Handbook, 2005 Publication	1 %
11	pdfprof.com Internet Source	1 %
12	Qing He, Hai Xia Gu, Qin Wei, Xu Wang. "A Novel DBSCAN Based on Binary Local Sensitive Hashing and Binary-KNN Representation", Advances in Multimedia, 2017 Publication	<1 %
13	medium.com Internet Source	<1 %
14	www.analyticsvidhya.com Internet Source	<1 %
15	Kumar, Vipin, Pang-Ning Tan, and Michael Steinbach. "Data Mining", Chapman &	<1 %

Hall/CRC Computer & Information Science Series, 2004.
Publication

16	scholarworks.lib.csusb.edu Internet Source	<1 %
17	Ali Ebadi Torkayesh, Sepehr Hendiani, Grit Walther, Sandra Venghaus. "Fueling the future: Overcoming the barriers to market development of renewable fuels in Germany using a novel analytical approach", European Journal of Operational Research, 2024 Publication	<1 %
18	assets.researchsquare.com Internet Source	<1 %
19	srividyasrinivasulamth522.sites.umassd.edu Internet Source	<1 %
20	mediatum.ub.tum.de Internet Source	<1 %
21	www.frontiersin.org Internet Source	<1 %
22	www.geeksforgeeks.org Internet Source	<1 %
23	Yi Li. "Weighted Centroid Localization Algorithm Based on MEA-BP Neural Network and DBSCAN Clustering", Journal of Physics: Conference Series, 2022	<1 %

Publication

24	Manjarini Mallik, Sanchita Das, Chandreyee Chowdhury. "Rank Based Iterative Clustering (RBIC) for indoor localization", Engineering Applications of Artificial Intelligence, 2023 Publication	<1 %
25	www.digitalvidya.com Internet Source	<1 %
26	Marzena Kryszkiewicz, Piotr Lasek. "Chapter 8 TI-DBSCAN: Clustering with DBSCAN by Means of theTriangle Inequality", Springer Science and Business Media LLC, 2010 Publication	<1 %
27	Sonal Kumari, Poonam Goyal, Ankit Sood, Dhruv Kumar, Sundar Balasubramaniam, Navneet Goyal. "Exact, Fast and Scalable Parallel DBSCAN for Commodity Platforms", Proceedings of the 18th International Conference on Distributed Computing and Networking - ICDCN '17, 2017 Publication	<1 %
28	easychair.org Internet Source	<1 %
29	lhiteshmth522.sites.umassd.edu Internet Source	<1 %
30	vdoc.pub Internet Source	<1 %

31	www.displayr.com Internet Source	<1%
32	El Maadi, Amar, and Mohand Said Djouadi. "Using a Light DBSCAN Algorithm for Visual Surveillance of Crowded Traffic Scenes", IETE Journal of Research, 2015. Publication	<1%
33	Lidia Ghosh, Dipanjan Konar. "Efficient fuzzy-pruned high dimensional clustering with minimal distance measure", Expert Systems with Applications, 2024 Publication	<1%
34	"Transactions on Large-Scale Data- and Knowledge-Centered Systems XXXIX", Springer Science and Business Media LLC, 2018 Publication	<1%
35	Yewang Chen, Lida Zhou, Songwen Pei, Zhiwen Yu, Yi Chen, Xin Liu, Jixiang Du, Naixue Xiong. "KNN-BLOCK DBSCAN: Fast Clustering for Large-Scale Data", IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019 Publication	<1%

Exclude quotes	Off	Exclude matches	Off

Exclude bibliography On