University of Brasília

Exact Sciences Institute
Computer Science Department

A method for defining customer spending behavior based on unsupervised machine learning

Gabriel Porto Oliveira

Monograph submitted in partial fullfilment of
the requirements to Bachelor Degree in Computer Engineering

Advisor
Mrs. Dra. Roberta Barbosa Oliveira

Brasília
2023

University of Brasília

Exact Sciences Institute
Computer Science Department

A method for defining customer spending behavior based on unsupervised machine learning

Gabriel Porto Oliveira

Monograph submitted in partial fullfilment of
the requirements to Bachelor Degree in Computer Engineering

Mrs. Dra. Roberta Barbosa Oliveira (Advisor)
CIC/UnB

Mr. Dr. Thiago Paulo Faleiros
CIC/UnB

Mr. Dr. Luís Paulo Faina Garcia
CIC/UnB

Mr. Dr. João Luiz Azevedo de Carvalho
Coordinator of Bachelor Degree in Computer Engineering

Brasília, February 06, 2023

Dedication

I dedicate this study to all my family, friends and other special people that helped me in anyway, directly or indirectly finish this long journey. I would like to give a special thanks to my parents that helped and supported me throughout my life and gave me the possibility to one day achieve something meaningful, the day has come. Without each one of you none of this would be possible.

Acknowledgements

First, I would like to thank my parents and friends for their continued help, support and time. I would also like to thank my advisor Roberta Barbosa Oliveira for her continued guidance and being available at all times.

Summary

With more financial information being generated each year, a necessity is created to use such information to develop financial products tailored to the experience of users. Here, a method to define possible spending patterns using categorized financial transactions is proposed. This study compares different clustering and outlier detecting algorithms with common metrics for internal validation of clusters, along with an empirical analysis of cluster balancing. The clustering algorithms compared are k-Means, Bisecting k-Means and Mean-Shift; besides, the outlier detecting algorithms used in this study are Local Outlier Factor and Isolation Forest. Lastly, the performance metrics used, namely, Silhouette Index, Calinski-Harabasz Index and Davies-Bouldin Index. Along with the method, a variant of the k-Means clustering algorithm, the Ok-Means, is proposed, pursuing the decrease of anomalies in clusters by removing outliers during the training process. The clustering and outlier removal algorithms usually were found to have better results when in use together. The proposed Ok-Means algorithm has found to give better results, based on internal validation metrics, when compared to the k-Means and k-Means + Isolation Forest combination in most of the tests; exhibiting a Silhouette Index score of 0.7920, Calinski-Harabasz Index of 37.1286 and Davies-Bouldin Index of 0.1404. Still, the Ok-Means does not solve the issue of unbalanced clusters. A visualization of spending patterns is created using the proposed method and validated by an expert in the area to help extract more information based on user behavior.

Keywords: unsupervised machine-learning, outlier removal, spending patterns

Resumo

À medida em que mais dados financeiros são gerados a cada ano, se faz necessário o uso desses para desenvolver produtos financeiros personalizados conforme a experiência do usuário. Neste trabalho é proposto um método para definir possíveis padrões de gastos a partir de transações financeiras categorizadas. São comparados diferentes algoritmos de clusterização e de detecção de outliers com métricas usuais para validação interna de grupos, em conjunto com análises empíricas do nível de balanceamento dos clusters. Os algoritmos de clusterização comparados são k-Means, Bisecting k-Means e Mean-Shift; ademais, os algoritmos de detecção de outliers usados neste trabalho são Local Outlier Factor e Isolation Forest. Por fim, as métricas de desempenho usadas, a saber, Silhoutte Index, Calinski-Harabasz Score e Davies-Bouldin Index. Juntamente com o método, uma variação do algoritmo de clusterização k-Means, o Ok-Means, é proposto com o objetivo de reduzir as anomalias nos clusters através da detecção de outliers durante o processo de treinamento. Os algoritmos de clusterização e detecção de outliers geralmente mostraram melhores resultados quando usados em conjunto. O algoritmo proposto Ok-Means demonstrou melhores resultados, baseados nas métricas de desempenho, quando comparado com o k-Means e com a combinação de k-Means + Isolation Forest, na maioria dos testes; exibindo um índice no Sillhouette Index de 0,7920, no Calinski-Harabasz Score apresenta 37,1286 e no Davies-Bouldin Index um valor de 0,1404. Ainda assim, o Ok-Means não resolve o problema do desbalanceamento dos clusters. Uma visualização dos padrões de gastos é criada usando um método proposto e validado por um especialista na área para auxiliar na extração de informações baseadas no comportamento do usuário.

Palavras-chave: aprendizado de máquina não supervisionado, remoção de outliers, padrões de gastos

Contents

List of Figures

List of Tables

Acronyms

ALPSO Adaptive Learning Particle Swarm Optimization.

BGSS Between Groups Sum of Squares.

CH Calinski-Harabasz.

DB Davies-Bouldin.

DBSCAN Density-based Spatial Clustering of Applications with Noise.

GMM Gaussian Mixture Model.

LOF Local Outlier Factor.

ML Machine Learning.

PCA Principal Component Analysis.

RFM Recency, Frequency and Monetary value.

RFMC Receny, Frequency, Monetary and Category.

SSE Squared Sum of Errors.

WCSS Within Cluster Sum of Squares.

WGSS Within Groups Sum of Squares.

WSS Within Sum of Squares.

Chapter 1

Introduction

The task of grouping customers based on financial transaction history has been extensively addressed in the literature [20][23]. The common Recency, Frequency and Monetary value (RFM) model is an example of a possible way to help in segmentation, divide into different groups. The RFM model utilizes the date of the last transaction (Recency), the number of transactions (Frequency) and the amount of money spent (Monetary Value) to help create values to be used in customer segmentation. This type of data can be useful for targeting specific groups of users with marketing strategies or for providing specific financial services to such groups [32]. Furthermore, the definition of spending pattern profiles could help optimize customer segmentation.

Brazil has become a resourceful ground for financial data since the introduction of OpenFinance[1], an initiative to share financial information between financial institutions. Furthermore, Pix[2] a fast payment mean that generates transaction data when compared to cash transactions has also been introduced. These new technologies make Brazil an excellent place to gather financial information and to use such new information to create new financial products.

Machine Learning (ML) can be defined as a field of which computers have been given the ability to learn without being explicitly programmed. This field is divided into several areas, one of them is unsupervised learning. In the unsupervised learning field algorithms are tasked to learn and present information based solely on the structure of the data [26]. One of the areas found within unsupervised ML is Clustering. Clustering algorithms are used to identify in data, groups of points that are similar to each other and group them together, along with grouping points that do not share similarities in different clusters. A commonly used clustering algorithm is the k-Means algorithm. This iteration based algorithm utilizes distances between data points to group together those close to each

[1]https://www.bcb.gov.br/en/financialstability/open_finance.
[2]https://www.bcb.gov.br/estabilidadefinanceira/pix.

other and form clusters. In order to assess the quality of results generated with clustering, internal validation metrics, that only analyze the characteristics of clusters, are used [12].

Majority of the studies that utilize clustering algorithms for customer segmentation only resort to information regarding total spending value and frequency of purchases or RFM model based input [15][32][35]. The problem of this approach is that it does not take into consideration the type of spending of a specific transaction, leaving out possibly important information that could give more insight into the way customers spend their money. For this study, only categorized transaction data, the origin of expense is known, is used along with clustering algorithms to create spending patterns.

The generation of spending pattern profiles alone is not enough to contribute insightful information to experts. To solve this issue a graphical visualization of generated profiles could help when analyzing results.

Some of the major issues when dealing with clustering algorithms is the presence of outliers that can negatively affect the results obtained from clusters by grouping together data that otherwise would not be with one another. The outlier points are generally data patterns that do not follow the characteristics found within normal data points [25]. In order to solve such issues outlier detection algorithms are used to detect such data points. In these specific domains, outliers could be related to users with abnormal spending or data that was erroneously added to databases.

1.1 Objectives

This work aims to detect and present spending profiles using categorized transaction data with clustering algorithms. A method is proposed for generating clusters from categorized transaction data. Specifically, the objectives can be defined as:

- Investigate performance of different clustering algorithms with and without outliers;
- Analyze the performance of outlier detectors combining with clustering algorithms;
- Compare the use of outlier detector and clustering algorithms to find the best combinations;
- Investigate potential improvements to the k-Means algorithm;
- Analyze the impact of internal validation metrics when generating spending patterns;
- Adapt an already existing graph view for the visualization of spending pattern profiles.

1.2 Structure of the document

This document is formatted as follows: Chapter 2 introduces important concepts for understanding this work. Chapter 3 presents an overview of literature found in the area of customer segmentation using clustering algorithms. Chapter 4 describes the steps and information necessary to create results in the next chapter. Chapter 5 presents the experiments performed and analyzes the obtained results. Finally, Chapter 6 gives an overview of the results found in this work and proposes possible next steps.

Chapter 2

Theory

This section presents an overview of the theory behind techniques and algorithms needed to make this study a reality. An overview of unsupervised machine-learning is explored, along with popular outlier detection and clustering algorithms. Clustering validation metrics to measure quality of clustering tasks are explained. Finally, some data pre-processing techniques are presented.

2.1 Unsupervised Learning

Unsupervised learning is a realm in the machine learning area that focuses on algorithms and techniques that do not have a target variable. Typically these algorithms focus on two separate fields: clustering and dimensionality reduction. The first one focuses on finding groups of data points that could be related to one another. The other, dimensionality reduction, takes a dataset with a high number of features and decreases this number while keeping the relevant relationship between them intact.

Figure 2.1[1] is an example of supervised and unsupervised learning, note that data points on the left image have symbols attached to them, these symbols represent the different labels they have. On the other hand, the image on right does not have such information; only their features can be taken into consideration. For this study only unsupervised machine learning algorithms will be used.

2.2 Outlier Detection Algorithms

Outlier detection algorithms are tasked to remove specific data points within a dataset that have a relationship with other points in a manner that does not respect the pattern

[1] https://analystprep.com/study-notes/cfa-level-2/quantitative-method/supervised-machine-learning-unsupervised-machine-learning-deep-learning.

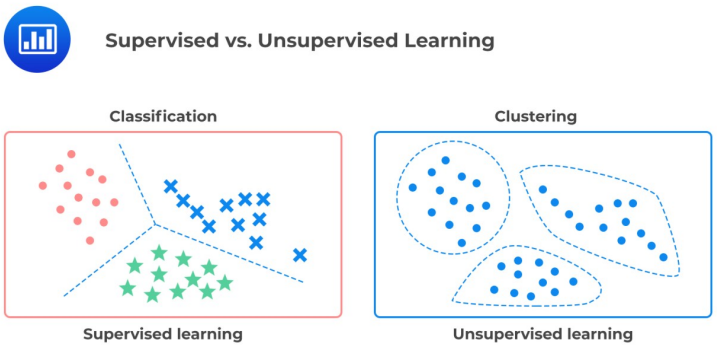

Figure 2.1: Example of supervised and unsupervised algorithms.

found within the data. Such anomalies could negatively affect the results when using machine learning training algorithms. Figures 2.2 and 2.3 show examples of a clustering process with and without outliers, respectively, note that the clusters created are different because of the outliers present.

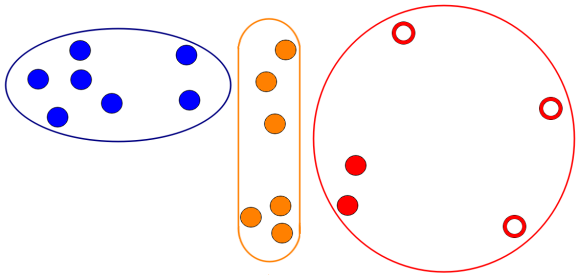

Figure 2.2: Example of clustering with outliers. Hollow points are the outliers.

2.2.1 Local Outlier Factor

The Local Outlier Factor (LOF) metric was first proposed by Breunig et al. [4]. It uses the density of points and distances between them to determine possible outliers. LOF is a value that indicates the degree of how likely a data point is to be considered an outlier. This algorithm has only one parameter that is a value k. The parameter indicates the number of neighbors to be taken into consideration during the execution.

The first step for this algorithm is to calculate a boundary that has a distance d from an origin point o to a given point p, such that the distance of the boundary is distant enough to have k data points within the calculated boundary and $D(o, p)$ is distance between points o and p. After setting the initial boundary, a reach-ability distance is calculated as the maximum value between the distance d and the distance between the points o and p. Furthermore, a local reach-ability density is calculated as the inverse of the

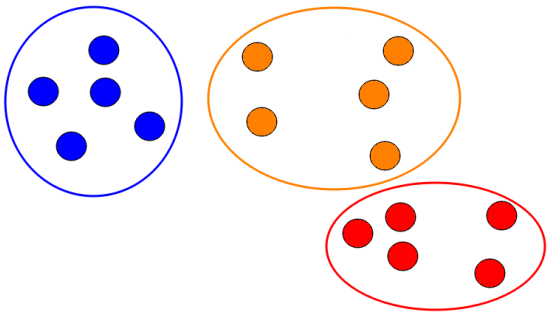

Figure 2.3: Example of clustering without outliers.

average reach-ability distance of the number of neighbors defined k and the reach-ability distance is defined in equation 2.1.

$$rd(p, o) = \max d, D(o, p) \tag{2.1}$$

Finally, the LOF value can be calculated as shown in Equation 2.2, where lrd is the local reach-ability density value and $Nk(p)$ are all the points that have a distance less than or equal to d. The parameter $N_{MinPts(p)}$ represents the points in the neighborhood of p that have at least k points.

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts(p)}} \frac{lrd_{MinPts(o)}}{lrd_{MinPts(p)}}}{|N_{MinPts(p)}|} \tag{2.2}$$

2.2.2 Isolation Forest

The Isolation Forest is an ensemble algorithm to detect outliers that relies on isolation trees to determine if a data point in a dataset can be considered an outlier. It was first proposed by Liu et al. [25].

This algorithm works by using the depth of trees to create an anomaly score to determine if a point is an anomaly or not. It creates trees by selecting attributes of a given instance and creating new nodes using a randomly selected split value. After creating the isolation trees, the path length is used to generate final anomaly scores. The path length $h(x)$ is defined as the number of edge points x traverses between the root node and the final node of a tree.

The anomaly score $s(x, n)$ is defined in Equation 2.3 where n is the number of data points. The parameter $c(n)$ is defined in Equation 2.4, note that $H(i)$, that is harmonic number, can be estimated using $ln(i) + e$ and that $E(x)$ is the average value of $h(x)$ from a group of trees, note that $e = 0.5772156649$ is the constant of Euler.

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{2.3}$$

$$c(n) = 2H(n-1) - (2(n-1)/n) \tag{2.4}$$

The value from $s(x,n)$ is used to differ anomalies from normal entries in the dataset. If s is close to 1 the entries are an anomaly, if it is a lot less than 0.5 the values are probably not anomalies and if all entries have values around 0.5 then there is no clear anomaly in the group. Figure 2.4 shows what anomaly scores are for a given dataset.

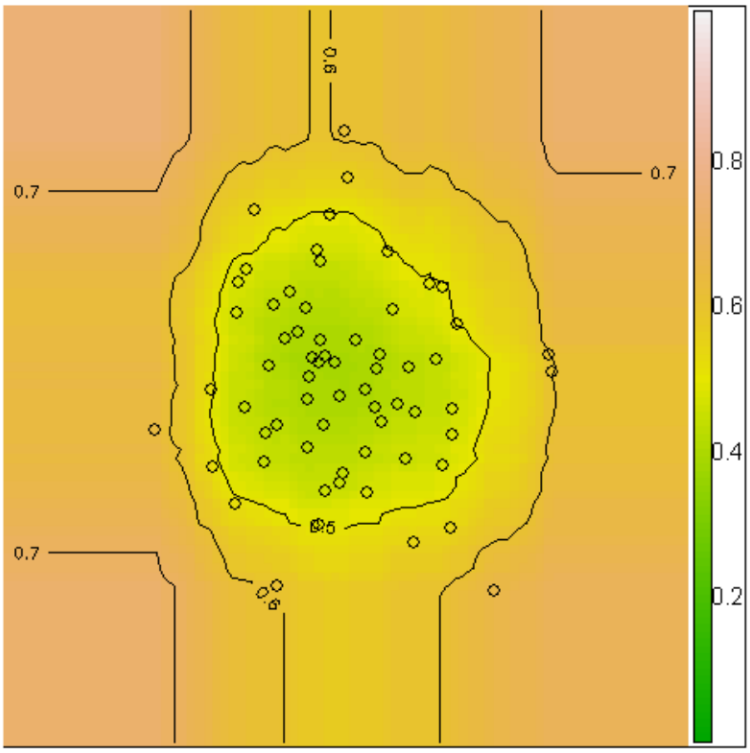

Figure 2.4: Contour with anomaly score shown, values on the right indicate the score by color (Source: [25]).

Isolation trees are created according to a parameter t, which the default value is 100. Moreover, the data is recursively partitioned and only a set number of samples n is used from the whole dataset, typically 256 data points is a good parameter for the number of samples [25].

2.3 Clustering Algorithms

The Clustering task is one of the areas found in the unsupervised machine-learning domain. These types of algorithms analyze the structure found within data presented to

them to find possible entries that share some commonality with other data points. An advantage of using clustering algorithms is that no target variable is needed. This means that an algorithm can be run to generate clusters and increase the possible knowledge gained when analyzing the data together with the structures created. As noted by Jain et al. [18] there are several different clustering algorithms in literature, in this study the algorithms Bisecting k-Means, k-Means and Mean-Shift algorithms will be used based on their presence in the literature review Section 3.

2.3.1 k-Means

The k-Means algorithm is one of the most famous clustering algorithms to be used, it's history and origin have been extensively studied by Bock [3]. In order to use this algorithm, it needs only the number of clusters to be created k and the initial centroids of each cluster.

The first step in this algorithm is to select the location of the initial centroids, this number of centroids has to be the same as k. Figure 2.5 shows an example of randomly selected centroids for a clustering process with $k = 3$. These initial points can be generated randomly, pre-defined or can be generated using some other algorithm.

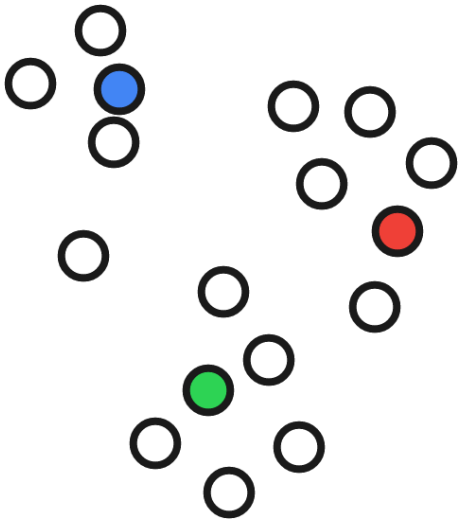

Figure 2.5: Initial centroids are defined.

Because of the effect initial centroids have on the final clustering results some work has been done to optimize the definition of the initial points. The k-Means++ algorithm is an example of an approach to optimize initial centroids introduced by Arthur and Vassilvitskii [2]. It utilizes probability measures between data points and random centroids to generate new center points to be used.

After the initial centroids are set, the clusters are generated by setting the closest points to each centroid to be in the selected cluster. Figure 2.6 shows the updating of each point according to the closest centroid. The steps of calculating the new centroids and updating each data point to be of the same cluster as the nearest centroid, are shown in Figures 2.7 and 2.8. The steps of calculating new centroids and new clusters are executed until some specific criterion is reached. The stopping criteria can be until the new centroids are the same as the last ones or a set specific iteration number is arrived at. The centroids are updated by calculating the mean of all points in a cluster.

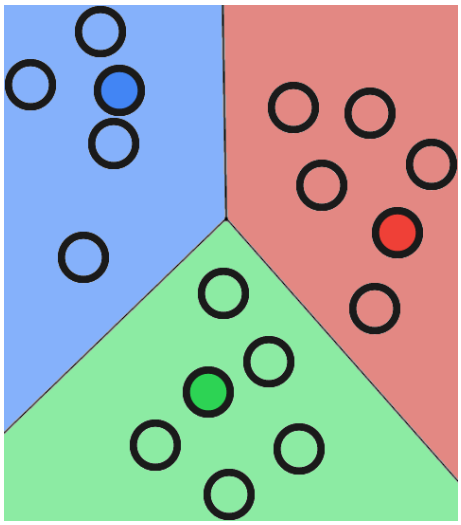

Figure 2.6: Data points near centroids are considered inside the same cluster.

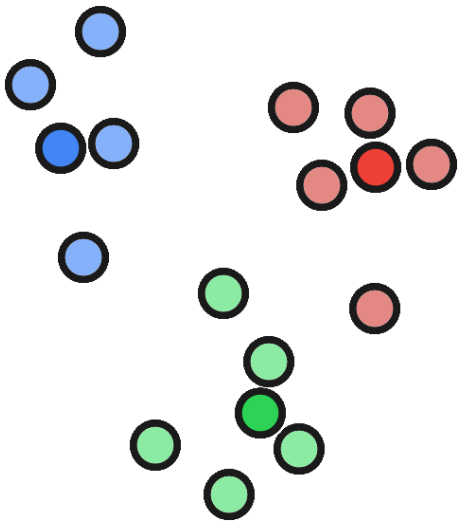

Figure 2.7: New centroids in each cluster are calculated as the mean value of all points in the cluster.

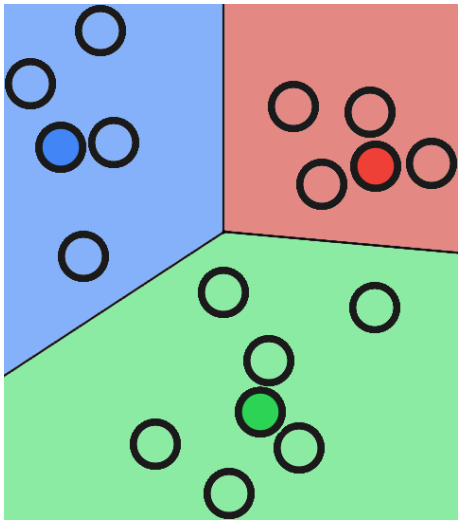

Figure 2.8: New cluster is generated.

2.3.2 Bisecting k-Means

The Bisecting k-Means algorithm extends the famous k-Means algorithm by utilizing a different method of setting clusters, it was proposed by Di and Gou [9]. The Bk-Means algorithm has the advantage of selecting initial centroids. In order to properly function, it still needs the number of clusters k but works differently to define clusters.

This algorithm functions by dividing the initial cluster of points into two each and calculating the Squared Sum of Errors (SSE) of the cluster. Equation 2.5 shows how to calculate the SSE value, n is the number of points, x is an ordinary point in the cluster and y the centroid. Then, it selects the cluster with the highest SSE value and divides it again. It executes these steps until it reaches the number of clusters previously determined. Moreover, Figure 2.9 shows the flow of execution of the Bisecting k-Means algorithm.

$$SSE = \sum_{i=1}^{n}(x_i - y)^2 \tag{2.5}$$

2.3.3 Mean-Shift

The Mean-Shift algorithm was first introduced by Fukunaga and Hostetler [11] and has been used in several applications, such as image segmentation and discontinuity preserving smoothing [7]. The algorithm utilizes data density to find local maxima, to function it needs as input a value h as bandwidth/radius and can automatically detect existing clusters differently than the k-Means algorithm, which needs the number of clusters as input.

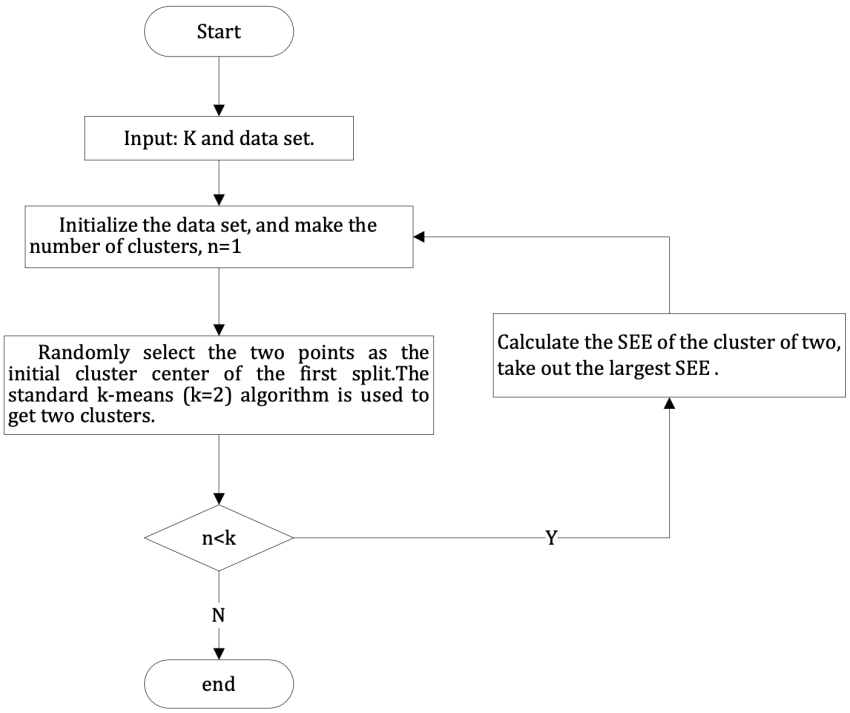

Figure 2.9: Bisecting k-Means flow (Source [9]).

In order to effectively function the Mean-Shift algorithm works as described in the steps below:

1. Initialize each starting point randomly;
2. Compute the mean point based on all points inside the set radius h;
3. Move the position of each data point to newly calculated center;
4. Repeat steps 2 and 3 until a specific convergence criteria is met; and
5. Data points are assigned to the clusters they moved to.

Figure 2.10 shows an example of the path made by executing the Mean-Shift algorithm.

2.4 Cluster Validation Metrics

Cluster validation metrics are used to measure the quality of results generated by machine learning algorithms. When using clustering algorithms, there are typically only two types of metrics, external or internal validation [34]. External validation is when outside knowledge is used to classify if the clustering process was successful or not. This is done by checking if data points that share the same category are in the same cluster. On the other hand, internal validation is when only characteristics of the generated clusters are taken

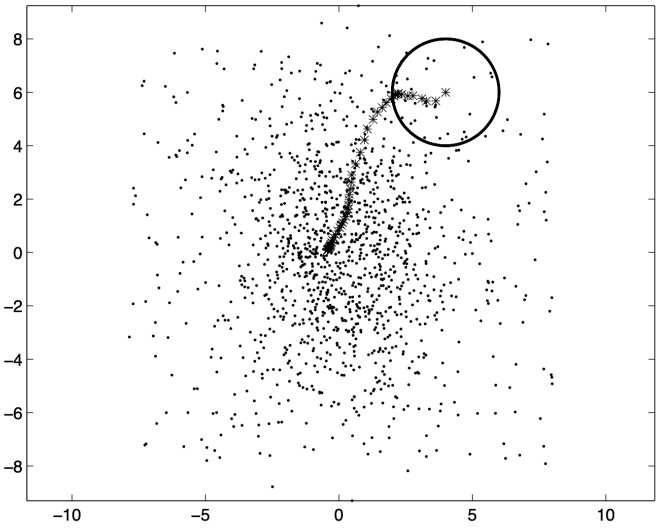

Figure 2.10: An example Path made by calculating the mean point then shifting (Source [6]).

into consideration therefore, no prior label information of different data points is needed. For this study, the internal validation metrics Calinski-Harabasz (CH), Davies-Bouldin (DB) and Silhouette Index will be used.

2.4.1 Calinski-Harabasz Index

The Calinski-Harabasz (CH) index was created by Caliński and Harabasz [5] as a validity index [12]. The index can be calculated using Equation 2.6, where BGSS is the between groups sum of squares, WGSS is the within groups sum of squares, n is the number of samples in the dataset and k is the number of clusters.

From Equation 2.6, it is possible to notice that clustering results with larger distances between clusters and clusters that have a tighter grouping of data points will have a higher CH index. There is no standard interval to indicate better clustering results and the final index will depend on the dataset, when comparing different results the higher the score the better.

$$CH = \frac{\frac{BGSS}{k-1}}{\frac{WGSS}{n-k}} \tag{2.6}$$

2.4.2 Silhouette Index

The Silhouette index, which was first introduced by Rousseeuw [31] and is a metric for measuring clustering quality. This index has a result in the interval $[-1, 1]$, the closer to 1 the more compact are the clusters and the higher is the distance between clusters thus

indicating better clustering processes. Equation 2.7 shows how to calculate the Silhouette Index $s(i)$ for an specific point i in the dataset. The index $a(i)$ represents the mean distance of entry i between all other points in the same cluster. The value $b(i)$ represents the minimum distance of the mean distance of entry i and all points in a different cluster.

$$s(i) = \frac{b(i) - a(i)}{max\{b(i)\, a(i)\}} \tag{2.7}$$

When analyzing the result of this metric, the closer together points are in a cluster and the furthest away it is to nearest cluster, the closer to 1 the index is. For measuring the whole clustering process, the mean of all Silhouette indexes calculated for all entries in the dataset is used.

2.4.3 Davies-Bouldin Index

The Davies-Bouldin index is a clustering validation metric introduced by Davies and Bouldin [8] and utilizes measures of dispersion and dissimilarity to identify better clustering. The Davies-Bouldin index DB can be calculated using Equation 2.8, where R_i is from Equation 2.9 and stands as the maximum set of values R_{ij} as seen in Equation 2.10. Note that k is the number of clusters and the closer the score is to 0 the better.

$$DB = \frac{1}{k} \sum_{i=1}^{k} R_i \tag{2.8}$$

$$R_i = \max_{i \neq j} R_{ij} \tag{2.9}$$

$$R_{ij} = \frac{S_i + S_j}{D_{ij}} \tag{2.10}$$

Each value of R_{ij} depends on dispersion value S from Equation 2.11 and dissimilarity of clusters centroids D from Equation 2.12. In this context C_i is a cluster, c is the centroid of a cluster C_i, a is the total number of points in a cluster C_i and $d(x, c)$ is the distance between data points x and c.

$$S_i = \left(\frac{1}{|a|} \sum_{x \in C_i} d^2(x, c) \right)^{\frac{1}{2}} \tag{2.11}$$

$$D_{ij} = \left(\sum_{l=1}^{d} |c_{il} - c_{jl}|^2 \right)^{\frac{1}{2}} \tag{2.12}$$

2.5 Data Pre-processing Methods

This section presents two methods of scaling data in order to change the interval but still keep the underlying structure of it.

2.5.1 Minimum-Maximum Normalization

The Minimum-Maximum Normalization is used to change the data features into an interval $[-1, 1]$, the method to calculate it is found in Equation 2.13 where x is the original data and x_1 is the normalized result.

$$x_1 = \frac{x - \min x}{\max x - \min x} \tag{2.13}$$

2.5.2 L2 Norm Normalization

L2 Norm Normalization is the process of changing data by turning the features to be in a $[-1, 1]$ range, the method to calculate this change is found in Equation 2.14 where x_2 is the normalized value, x is the original data and $\|x\|$ is the length of the vector in the Euclidean space.

$$x_2 = \frac{x}{\|x\|} \tag{2.14}$$

Chapter 3

Literature Survey

The current chapter presents an overview of literature done in the clustering of customers area. Some of the work apply the RFM model, while others use different types of input data for clustering. For this work, the terms customer clustering and customer segmentation are used interchangeably.

3.1 RFM based segmentation

The RFM model has been widely used in the literature for the task of customer segmentation [10]. It is defined as a combination of three variables that can be represented in different ways, such ranges of values or a single value determined from the previous three. These variables are: (1) Recency, time since the last purchase; (2) Frequency, typically the number of purchases made; and (3) Monetary Value, the amount of money spent. Usually, these measurements are taken within periods of time. The studies addressed in this section utilize the RFM model or use data heavily influenced by it along with some clustering algorithms to better extract information.

Lefait and Kechadi [22] propose an architecture to cluster customers using the RFM model and k-Means algorithm for clustering. Their work suggests a way of using RFM data to create several clusters and gain knowledge from visual representation of selected clusters. The proposed architecture defines the following steps: (1) data input, (2) cluster generation, (3) cluster selection, and (4) cluster representation utilizing a graphical view. For the experimental results, a dataset from the SLDS09 challenge[1] was used and consisted of purchase logs of 10 000 customers over a period of 62 weeks for 6 different brands. For generating clusters the k-Means algorithm was trained with values K 2-10, 15, 20, 25. The metrics F-Measure, harmonic mean of the precision and recall were used to measure the homogeneity of the clusters. Overall the architecture is able to support the definition

[1]Symposium Apprentissage et Science des Donnees 2009.

of the best clustering results for expert analysis, using the aforementioned steps and is able to create a visual representation to show customer segmentation.

Parikh and Abdelfattah [28] focused on comparing several clustering algorithms with RFM data. The algorithms density-based spatial clustering of applications with noise (DBSCAN), Mean-Shift, k-Means and Agglomerative Clustering were used to segment customer of an online retailer. The dataset consisted of 541 909 customers with 8 features, the final dataset used for clustering only contained the RFM values. Furthermore, the value $K = 3$ was found to have the best score using the metric Within Cluster Sum of Squares (WCSS) and elbow method of analysis.

Maryani et al. [27] have also the k-Means algorithm for customer segmentation. In order to segment a dataset of 82 648 transactions the RFM model was used to generate a final dataset of 102 customers. After the clustering process was done, two clusters with 63 and 39 customers were found.

Hu et al. [15] added a new parameter T to the RFM model along with using clustering algorithms to define users into five possible groups. The new parameter was added to measure the time between the first and last transaction of a client as a way to measure loyalty. The minimum-maximum normalization procedure was done before the clustering phase to map features to the [0,1] interval, this is important to eliminate dimension and data value range. The dataset used consisted of 1796 customers of a restaurant. During the clustering phase the algorithm k-Means++ was used to generate five clusters, this is an enhanced version of the normal version that optimizes the starting points. The final number of customers in each cluster is visible in Table 3.1.

Table 3.1: Number of customers in each cluster.

Cluster	**Number of customers**
1	431
2	92
3	215
4	556
5	502

Allegue et al. [1] expanded the RFM model by adding the component C to create a new model, named Receny, Frequency, Monetary and Category (RFMC), proposed an architecture that uses user feedback to enhance the model performance. The new component is related to the category of each transaction, this new feature reflects better the spending behavior of users. The new model was tested together with the k-Means clustering algorithm. The dataset used for training consisted of 120 222 customers. The metric Within Sum of Squares (WSS) was used to select the value 7 for the parameter

K. The final test resulted in validation metrics Silhouette Score and Davies-Bouldin were used to evaluate the final results with scores 0.77 and 0.37, respectively.

Huang et al. [16] have expanded the RFM model by adding a parameter C. This parameter is different than the one found in the work by Allegue et al. [1] and is related to a community factor. This study used the k-Means algorithm for the clustering task and found the parameter $K = 5$ to be the best by using the elbow method with metric SSE of approximately 800.

Pondel and Korczak [30] analyzed the methods employed in an already existing customer recommendation system. The system utilizes a process of collective clustering, this means that many different clustering algorithms are used together to generate richer segmentation. Furthermore, the data used for training involves transaction data, geolocation and social network information. A process is needed to unify clustering results and involves the use of both internal and external validation and expert analysis. A process of unification then uses the available information of the clustering results to then unify selected clusters. The final experiment utilizes RFM inspired data and extends it with the number of user orders, the final number of customers consist of 56 237 that made at least 2 purchases. The experiment focuses on segmenting users in 6 clusters and utilizes algorithms k-Means, bisecting k-Means, GMM and an extended version of DBSCAN. Clusters generated by the k-Means algorithm were found to be the most impacted by the recency and frequency features, as there was a greater variation in these features, and segments created were generally balanced yet in some cases clients with values far from each other were found in the same cluster. The bisecting k-Means was able to create balanced clusters as the k-Means algorithm, was able to have a greater variety but still encountered the same problem of clients with values far from one another was present. After the process o clustering unification took place, a larger number of different segments was identified when compared to only using one algorithm with better interpretation.

3.2 Not-RFM based segmentation

Other studies have used clustering algorithms to segment customers utilizing different types of input data.

Li et al. [24] proposed a new k-Means algorithm that uses Adaptive Learning Particle Swarm Optimization (ALPSO) to help optimize the initial points for the k-Means algorithm. A comparison between the standard k-Means algorithm, using particle swarm optimization with k-Means and using the adaptive learning particle swarm algorithm was done using five different datasets. The initial testing found the new algorithm to have bet-

ter results than the others. The newly proposed algorithm KM-ALPSO was then tested using a dataset from a Chinese grape market.

Kansal et al. [21] analyzed a comparison between clustering algorithms k-Means, Mean-Shift and Agglomerative Clustering algorithms. The dataset used consisted of 200 customers of a local retail shop with 2 features, number of visits and total shopping. The parameter $K = 5$ was used for k-Means after using the elbow method with metric SSE. Two internal validation metrics were used for the comparison of the clusters; namely, Silhouette Score and Calinski-Harabasz. For the Silhouette Score the k-Means and Agglomerative Clustering algorithms scored 0.55 and the Mean-Shift algorithm scored 0.53.

Umuhoza et al. [32] used a dataset of credit card transactions from an Egyptian financial institution. The k-Means algorithm was applied along with validation metrics Silhouette Score and Calinski-Harabasz Score. The number of four clusters was found to be the best. Different types of customer behavior were determined based on the results. Using the generated clusters information, marketing strategies were defined for each of the spending profiles.

Wu and Lin [33] applied the k-Means algorithm along with the customer value matrix model, instead of the common RFM model, for generating clusters of customers. The clustering algorithm was run several times in order to find clusters within clusters and a final number of nine different clusters was found to be optimal for general customer expending. A clustering process was also made for taking a look at fluctuations of expending compared to prior months. Finally, a combination of 54 types of consumption profiles was the final result, each one with different characteristics.

Zakrzewska and Murlewski [35] compared parameters of scalability, effectiveness and outlier detection using k-Means, DBSCAN and Two-Phase clustering algorithms. The study made comparisons using two and multi dimensional and analyzed the effect of standardization on the data. The k-Means algorithm was found to be the fastest compared to the others while generating balanced clusters with a good amount of data points in each but it has no way of removing outliers. DBSCAN had the second best execution time however identified too many data points as outliers and generated a single large cluster and many smaller ones. The Two-Phase algorithm managed to detect the outliers with more accuracy but also created one large cluster and several smaller ones while taking the longest to execute.

Holm [14] used categorized transactions to create clusters. The data originated from banking transactions and contained several features that were divided into two categories. The first one is called category statistics which contains information regarding the nature of the transactions. The second category is general statistics which has general information such as amount spent and day of the week for the expense. The Variance

Threshold function was used to remove features from the dataset along with Principal Component Analysis (PCA). For the clustering step, the algorithms k-Means and Hierarchical Clustering were used with the validation metrics Calinski-Harabasz, Silhouette Score and Dunn index. The credit scores of some customers were also used to check the best number of clusters. The algorithms k-Means and Hierarchical Clustering using Ward linkage generated similar clustering when using a low number of clusters.

3.3 Final considerations

As mentioned before, both RFM and non-RFM based studies have used clustering algorithms to generate possible spending patterns. Some studies have established the RFM to not be enough, as it limits the amount of data used to only three values, and have extended the model in some capacity [1][15][16]. Others have used totally different approaches because of shortcomings the RFM model has shown [33].

To the best of our knowledge, techniques that used categorized transaction data were not widely used and were only seen in some studies [1][14]. Categorized data has shown to give more insightful information compared to the normal RFM model approach [1].

The k-Means algorithm has been widely used by various studies found in the literature review, some other clustering algorithms Agglomerative Clustering, Hierarchical Clustering, Mean-Shift and DBSCAN have also been used.

Regarding techniques that were found other than clustering algorithms. For feature detection, Variance Threshold has been used along with PCA for dimensionality reduction [14]. Scaling of features is another technique that is present in the literature [15][35]. Outlier detection techniques were not as explored and the topic of outliers only taken into consideration by Zakrzewska and Murlewski [35] when using clustering algorithm but not with specific techniques to remove such data points.

In this work, the usage of outlier detectors along with clustering algorithms are tested together to find the best combination. Furthermore, a profile visualization method is presented along with a new k-Means variant with outlier detection capability.

Table 3.2: Overview of literature studies.

Reference	**Year**	**Clustering Algorithms**	**Validation Metrics**
Allegue et al. [1]	2020	k-Means	Silhouette Score and Davies-Bouldin
Holm [14]	2018	k-Means and Hierarchical Clustering	Dunn Index, Silhouette Score and Calinski-Harabasz
Hu et al. [15]	2020	k-Means++	-
Huang et al. [16]	2020	k-Means	SSE
Kansal et al. [21]	2018	k-Means, Mean-Shift and Agglomerative Clustering	Silhouette Score and Calinski-Harabasz
Lefait and Kechadi [22]	2010	k-Means	Execution Time
Li et al. [24]	2021	KM-ALPSO	Silhouette Score, Davies-Bouldin and Calinski-Harabasz
Maryani et al. [27]	2018	k-Means	-
Parikh and Abdelfattah [28]	2020	DBSCAN, Mean-Shift, k-Means and Agglomerative Clustering	WCSS
Pondel and Korczak [30]	2018	k-Means, GMM, DBSCAN and Bisecting k-Means	Several internal and external metrics including: Silhouette Index, DB, F-Score
Umuhoza et al. [32]	2020	k-Means	Silhouette Score and Calinski-Harabasz
Wu and Lin [33]	2005	k-Means	-
Zakrzewska and Murlewski [35]	2005	k-Means, DBSCAN and Two-Phase	-

Chapter 4

Methodology

The main objective of this study is to propose a method for generating spending patterns using categorized transaction data. A process is needed to find the best combination of outlier detection and clustering algorithms. Moreover, a new variation of the k-Means algorithm with some outlier detection capabilities is also proposed; the new algorithm is called Ok-Means.

This chapter presents all proposed steps to create the final results along with a discussion of each one of them. Section 4.1 will describe the proposed method to properly create spending patterns based on categorized transaction data and explain each step present. Section 4.2 presents the steps needed to create datasets from the database available. Section 4.3 describes the optimization process to decide best parameters for the algorithms. The following Section 4.4 presents the spending pattern visualization to help visualize the generated results. The last Section 4.5 presents and explains the new k-Means variant algorithm.

4.1 Spending Pattern Generation Method

The method for creating the spending patterns consists of four steps, as shown in Figure 4.1. It contains (1) data input, (2) data pre-processing, (3) outlier detection and (4) clustering phases. The first step consists of raw data analyses and removal of non-important data. During the second step, the raw data is formatted into a training friendly format and transaction category mapping occurs. In the third step anomalies are removed from the dataset in order to remove data that could negatively effect the final results. The last step is the Clustering step, here the clustering algorithms use the data formatted from the last steps to generate clusters, before the data is used for clustering, a process of feature standardization is applied.

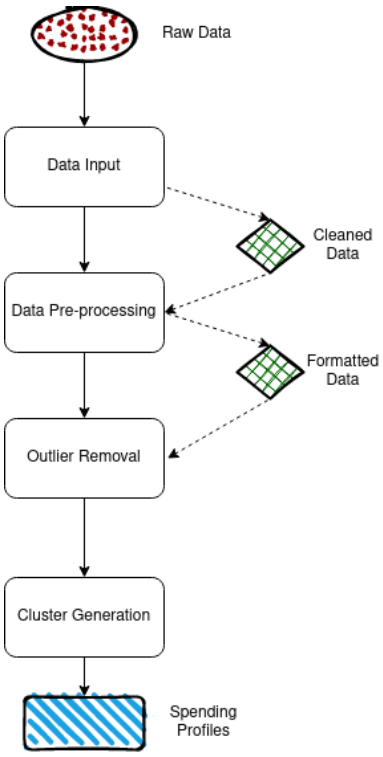

Figure 4.1: Illustration of the proposed method, note the different types of input and output data for different steps.

The algorithms Isolation Forest and Local Outlier Factor will be used as outlier detectors and the clustering algorithms k-Means, Mean-Shift and Bisecting k-Means will be used and compared. Note that the k-Means algorithm implemented in the SKLearn library utilizes the k-Means++ variant [2].

4.2 Data Pre-Processing

A total of two databases were made available to test and analyze the training models, both of them are from a Peruvian financial technology company that operates two different but very similar mobile applications, the main objective of the company is to serve the underbanked or unbanked population, this population consists of people with no or little access to banking services. The database consists of financial information of users from both applications with incomes, expenses, investments and some more information related to financial habits. Database Peru originates from users in Spanish-speaking countries in Latin America, mostly Peru and contains 5281 users. Database Brazil originates from users in Brazil with 433 users. In this work, the name database is used when specifically related to the original data extracted from the company, after the data is pre-processed it is then called a dataset.

The original databases were structured in a way that could not be used for clustering tasks, as shown in Figure 4.2 and explained in Table 4.1. In the Data-Input step, (1) the raw data is analyzed and (2) personal and not-needed information is removed. In the Data Pre-processing step, (1) original categories are mapped to new ones in order to remove redundant information, (2) features are created using transaction data available and (3) users with less than five transactions are filtered out. Motivated by the many RFM based studies discussed in Section 3.1, the parameters Frequency (number of transactions) and Monetary Value (sum value of transactions) inspired the features used in this study. The final dataset features consist of the number of expenses and the sum of all expenses of each new mapped category along with the total amount of expenses and the sum of expenses of each user. Here, the Recency value is not employed, since several users stopped inserting data into the used mobile application in very different intervals, some users stopped using the application in a few days, and others used it for some more time. Therefore, all users that have less than five transactions are removed so only those that had some amount of meaningful information were kept.


```
"bills": [
    {
        "category": "household",
        "id": "79604000-57b9-11ec-e673-d580ba8d6d38",
        "date": "2021-12-07",
        "value": 10.0,
        "name": "Mantenimiento",
        "role": "RoleType.PERSONAL",
        "periodicity": "PeriodicityType.MONTHLY"
    },
    {
        "date": "2021-12-07",
        "role": "RoleType.PERSONAL",
        "id": "79604000-57b9-11ec-e924-65fe322086f8",
        "value": 1020.0,
        "periodicity": "PeriodicityType.MONTHLY",
        "category": "household",
        "name": "Alquiler"
    },
    {
        "category": "transport",
        "id": "79606710-57b9-11ec-9861-1756fcec3d4e",
        "value": 15.0,
        "name": "Bus",
        "periodicity": "PeriodicityType.NONE",
        "date": "2021-12-07",
        "role": "RoleType.PERSONAL"
    },
```


Figure 4.2: One snippet of data present in the original database Peru.

Since the two databases used distinct categories, a process to group different transaction categories was needed and the number of features changed from Dataset Peru to

Table 4.1: Explanation of each key found in the original database structure.

Key	Explanation
Bills	List of transactions of an specific user
Category	Category of the transaction
Id	Identification number of the transaction
Date	Date of when transaction was registered
Name	Name given to the transaction
Role	Indicator of whether the transaction is of a business of personal account
Periodicity	Information if the registered transaction can repeat in a specific time interval

Dataset Brazil. Each dataset is derived from Databases Peru and Brazil respectively.

Tables 4.2 and 4.3 show the category mapping done for each of databases Peru and Brazil. Each category on column Original Category was mapped to a category on column New Category. This process reduces the number of features in the dataset and removes redundant ones while still keeping relevant categories.

After the data pre-processing step was done Dataset Brazil contained 109 users with 28 features each and Dataset Peru with 871 users and 26 features.

4.3 Optimization and Performance Evaluation Process

In order to find the best combination of Outlier detection and Clustering algorithms a process of testing several hyper-parameters was needed to find the best possible scenario with all possible combinations. The internal validation performance metrics Davies-Bouldin, Silhouette Index and Calinski-Harabasz Indexes, discussed in Section 2.4, will be used to better determine what the best combinations are.

Table 4.4 has the hyper-parameters used for the clustering algorithms and Table 4.5 has the hyper-parameters used for the outlier detection algorithms. Because Dataset Peru differs from Dataset Brazil for the Mean-Shift algorithm the lower bound was set as 0.15 bandwidth. It is also important to note that the Isolation Forest algorithm used is from the ML library Sci-Kit Learn[1], in this version the trees used are called Extreme Randomized Trees introduced by Geurts et al. [13].

[1]Library [29] available at https://scikit-learn.org/stable/preface.html.

Table 4.2: Category mapping for Dataset Peru.

Original Category	**New Category**
Household Rent	Household
Entertainment	Entertainment
Transport Car Fuel Parking	Transport
Personal Care Beauty Fitness	Personal Care
Education	Education
Feeding Dining Groceries Market	Feeding
Taxes	Taxes
Health Health Care	Health
Finances	Finances
Travel	Travel
Shopping Clothing	Shopping
Others	Others

4.4 Cluster Visualization

A cluster visualization method, adapted from a website[2], is proposed to help experts in personal finance easily identify spending patterns. In order to verify the possible usability of the cluster visualization a graph is generated using the proposed method steps with one of the datasets as input. The results of cluster visualization are sent to an expert in the field for analysis of the spending pattern profiles created. An analysis is asked to better understand and classify the meaningfulness of the generated graph for finding spending patterns. The expert is the owner of the company the databases originated from, has a bachelor's degree of Applied Sciences with a focus in Economic and International Development.

The cluster visualization contains several graphs, one for each feature, the x axis for each one contains the number of the cluster and the y axis a box and whisker chart is

[2]https://towardsdatascience.com/best-practices-for-visualizing-your-cluster-results-20a3baac7426.

Table 4.3: Category mapping for Dataset Brazil.

Original Category	**New Category**
Household Rent	Household
Entertainment	Entertainment
Transport Car Fuel Parking Transportation	Transport
Telephone	Telephone
Personal Care Beauty Fitness	Personal Care
Education	Education
Feeding Dining Groceries Market	Feeding
Taxes	Taxes
Health Health Care	Health
Bonuses Transfers Loans and Financing	Finances
Travel	Travel
Shopping Clothing	Shopping
Others	Others

used to show the behavior of clusters for each feature.

4.5 Ok-Means algorithm

Knowing the issues the standard k-Means algorithm has dealing with outliers [19], a variant of it is proposed called Outlier k-Means (Ok-Means). The main difference is that some level capability to deal with outliers is added during the clustering process and it takes into consideration the structure of already defined data points in a cluster. The idea to introduce this variant originates from preliminary results found when using the

Table 4.4: Parameters for clustering algorithms.

Algorithm	**Parameter**	**Start Value**	**End Value**	**Step**
MeanShift	Bandwidth	0.8 (0.15)	2.2	0.05
k-Means	K	2	10	1
Bk-Means	K	2	10	1

Values between parentheses are for dataset Brazil

Table 4.5: Parameters for outlier detection algorithms.

Algorithm	**Parameter**	**Start Value**	**End Value**	**Step**
Isolation Forest	Number of Trees	50	100	2
Local Outlier Factor	Number of Neighbors	10	50	1

k-Means algorithm. Furthermore, the Ok-Means variant is proposed as replacement of the normal algorithm with some outlier removal capability.

The new capability is added by observing the stopping criteria of the standard k-Means. Normally the algorithm stops it's execution when either a specific iteration number is reached or the centroids of each cluster stop updating. The former stopping criteria will be used to add a manner of removing possible anomalies.

The Ok-Means algorithm utilizes three main parameters to function properly. The first parameter α is the maximum number of iterations that should be executed. The second parameter β is the iteration number when the removing of anomalies should begin. The last parameter θ is the l2-norm scaled feature threshold to determine an anomaly within a cluster and remove it.

The Ok-Means algorithm works as follows in this order:

1. Initial centroids are set;
2. Data points near the closest centroid are set to be of the same cluster;
3. New cluster centroids are calculated;
4. Data points of the closest centroids are updated to be of the same clusters;
5. If the current iteration number is more than or equal to β continue to the next step, otherwise, go to the last step;
6. Scale features by l2-norm of each cluster, remove data points that have any scaled feature over θ;

7. Continue from Step 3 if the current iteration number is higher than α, otherwise, increase the current iteration number by one.

When removing values over a specific standard deviation threshold θ it helps make sure that anomalies within a specific cluster are removed and the algorithm can continue updating centroids and update points accordingly. This process could help removing outliers during the training phase and make the necessity of techniques to remove outliers less needed.

Chapter 5

Experimental Results

This chapter presents the results obtained with the best combinations of outlier detection and clustering algorithms. These results are compared with each other and with Ok-means, the newly proposed algorithm. Furthermore, the spending profile visualization is presented.

5.1 Set Up

To generate all final results the data was processed on a desktop computer with the operating system Kubuntu version 22.04, CPU Ryzen 5 1600 and 32 Gb of RAM.

The Python programming language was used to pre-process the data and train the models; the version used is 3.11.0. The implementations of outlier detection and clustering algorithms were used from the library SKLearn[1] with version 1.1.2. The code needed to generate the results presented here can be found on a GitHub public repository[2].

5.2 Dataset

As seen in Section 4.2 two datasets were obtained from a financial technology company. After the pre-processing step, Dataset Peru contained 871 users with 26 features each and Dataset Brazil contained 109 entries with 28 features. A visualization of an entry in Dataset Brazil showing each feature is shown in Figure 5.1.

[1]Library [29] available at https://scikit-learn.org/stable/preface.html.
[2]https://github.com/patonoide/spending-patterns-clustering-final-paper.

```
"n_household": 6,
"household_total": 970.0,
"n_entertainment": 0,
"entertainment_total": 0,
"n_transport": 0,
"transport_total": 0,
"n_personalCare": 1,
"personalCare_total": 50.0,
"n_education": 0,
"education_total": 0,
"n_feeding": 1,
"feeding_total": 450.0,
"n_taxes": 0,
"taxes_total": 0,
"n_others": 0,
"others_total": 0,
"n_health": 2,
"health_total": 208.0,
"n_finances": 0,
"finances_total": 0,
"n_travel": 0,
"travel_total": 0,
"n_shopping": 0,
"shopping_total": 0,
"n_telephone": 0,
"telephone_total": 0,
"n_bills": 10,
"total_bills": 1678.0
```


Figure 5.1: Visualization of an entry in Dataset Brazil.

5.3 Outlier Detection

Outlier Detection algorithms help make sure the results obtained during training phases are not influenced by anomalies, this necessity makes these algorithms highly important. The algorithms parameters are visible in Table 4.5. The algorithms Isolation Forest and LOF, mentioned in Section 2.2, are used to detect anomalies in Datasets Brazil and Peru, an analysis of the number of outliers detected will be made based on the parameters used. Since the LOF algorithm returns a value that indicates how likely a data point is considered an outlier, thresholds 10, 20 and 40 were used to determine if a point is considered an outlier.

5.3.1 Local Outlier Factor

The LOF algorithm utilizes the number of neighbors parameter to determine the final LOF score of data points. When using Dataset Peru it is possible to detect that the higher the number of neighbors the more outliers are detected; Figure 5.2 shows such behavior. Taking into consideration the inner workings of this outlier detector, this result could mean that some of the data is tightly packed. That could also mean that some other parts are more sparse, as increasing the number of neighbors also increases the number of outliers detected.

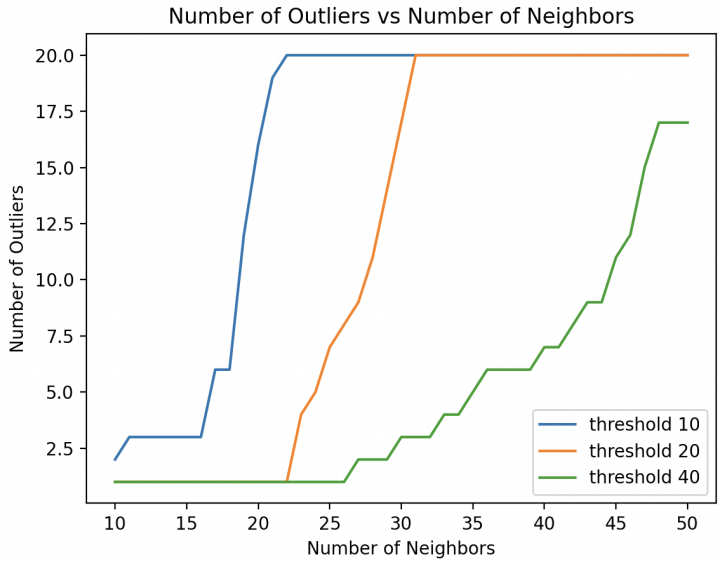

Figure 5.2: Number of outliers detected in Dataset Peru when changing parameter number of neighbors for the LOF algorithm with LOF thresholds of 10, 20 and 40.

The number of outliers detected when changing the parameter number of neighbors for Dataset Brazil is shown in Figure 5.3. It is possible to note that with thresholds 20 and 40 no outliers were detected.

5.3.2 Isolation Forest

Anomaly detection with the Isolation Forest algorithm involves setting the parameter number of trees as noted in Table 4.5. Using this algorithm with Datasets Peru and Brazil generates Figure 5.4. As shown in the generated graphs there is no clear relation between the number of trees and the number of outliers detected, this could be related to the different implementation used in the Sci-Kit Learn Library. Overall the Isolation Forest outlier detector is able to detect more outliers when comparing it to the LOF, setting lower LOF thresholds could result in different results.

5.4 Clustering without outlier detection

The following test results were executed without removing any outliers from Datasets Peru and Brazil.

5.4.1 Dataset Peru

Results with the k-Means algorithm and Dataset Peru are located in Table 5.2. Note that the best Silhouette and DB index scores are the same in some instances. Taking a look at

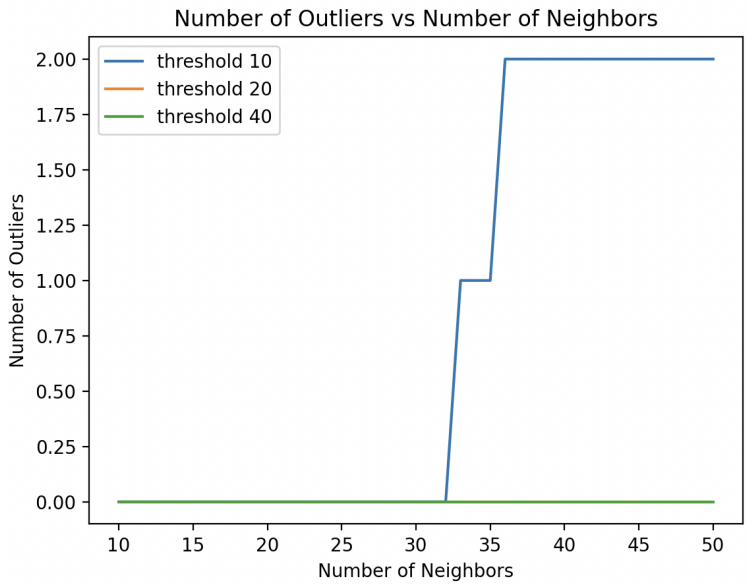

Figure 5.3: Number of outliers detected in Dataset Brazil with LOF thresholds of 10, 20 and 40. Lines for thresholds 20 and 40 are always zero.

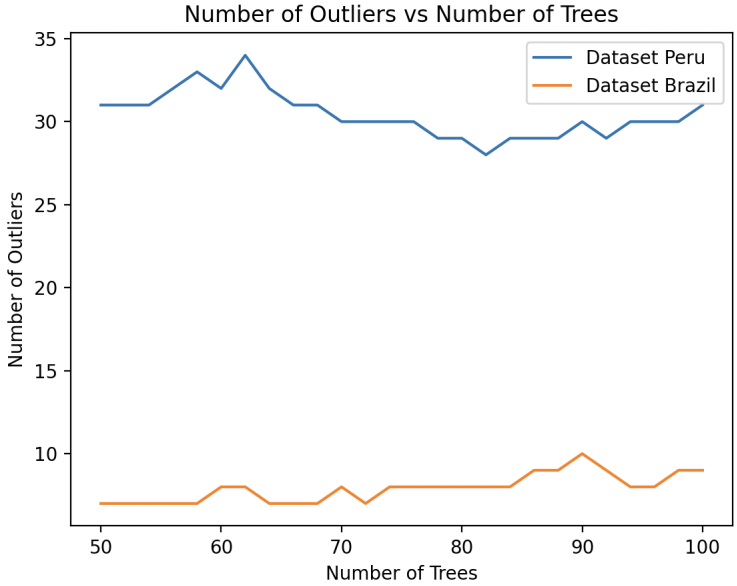

Figure 5.4: Number of outliers detected when changing parameter number of trees for the Isolation Forest algorithm.

number of data points when using $K = 2$, one cluster has 870 entries and the other only one. This is interesting since most performance metrics are very good but they do not take into consideration the imbalance of the clusters, this imbalance results in a cluster analysis that does not give any meaningful information since almost all data points are grouped together.

Table 5.3 has the result when using the Bisecting k-Means algorithm with Dataset Peru. All the best scores are the same with number of clusters K = 2, overall worse than the k-Means algorithm. However, this result is more balanced, consisting of one cluster containing 803 entries and the other 68, when compared to the k-Means result.

Results obtained with the Mean-Shift algorithm and Dataset Peru are shown in Table 5.4. Observing the number of entries with Bandwidth 0.95 and 1.00, a similar result is found with the k-Means algorithm, one cluster has the majority of data points and the others only 1 entry.

Overall the results with the best CH score tend to have more clusters with more balanced clustering. The k-Means result with $k = 10$ is visible in Table 5.1 and shows such behavior.

Table 5.1: Results of Dataset Peru with k-Means algorithm without any outlier detection algorithm and $k = 10$.

Cluster Number	**Number of Data Points**
0	127
1	287
2	107
3	77
4	2
5	7
6	1
7	195
8	38
9	30

5.4.2 Dataset Brazil

Dataset Brazil results with the k-Means and Bk-Means algorithms are located in Table 5.5; both of these algorithms scored the same best results. When analyzing the number of entries with $K = 2$ the same balancing issue mentioned earlier arises; one cluster has 108 data points and the other one data point.

Table 5.2: Results of Dataset Peru with k-Means algorithm without any outlier detection algorithm.

Number of Clusters	Silhouette	CH	DB
2	**0.9004**	82.7399	**0.0662**
3	0.8862	78.0749	0.0720
4	0.8599	91.7117	0.4726
5	0.3787	90.3491	1.0096
6	0.8264	96.0995	0.3182
7	0.4248	112.3695	1.1568
8	0.3939	105.0298	0.7113
9	0.4109	112.6364	0.8512
10	0.4191	**120.0695**	0.9635

Note: best score in **bold**.

Table 5.3: Results of Dataset Peru with Bk-Means algorithm without any outlier detection algorithm.

Number of Clusters	Silhouette	CH	DB
2	**0.4870**	**58.4129**	**1.7541**
3	0.1641	39.4979	2.7071
4	0.0272	30.4135	2.6266
5	0.0889	43.5991	2.3491
6	0.0897	53.5237	2.0997
7	0.0754	46.0899	2.0954
8	0.0457	40.3647	2.2302
9	0.04605	43.5568	2.0363
10	0.0545	43.1185	2.0174

Note: best score in **bold**.

Table 5.4: Results of Dataset Peru with Mean-Shift algorithm without any outlier detection algorithm.

Number of Clusters	Silhouette	CH	DB	Bandwidth
3	**0.9011**	91.5934	**0.0651**	1.00 - 1.40
5	0.8649	85.5543	0.0757	0.95
6	0.8653	85.7551	0.0763	0.85 - 0.90
7	0.8661	**93.5256**	0.4796	0.80

Note: best score in **bold**.

Table 5.7 shows the same results found with the other clustering algorithms; the same issues of balancing are also present with the same exact number of data points in each cluster.

Table 5.5: Results of Dataset Brazil with k-Means algorithm without any outlier detection algorithm.

Number of Clusters	Silhouette	CH	DB
2	**0.7871**	**34.8118**	**0.1404**
3	0.48327	29.6952	1.2999
4	0.4808	25.7390	1.1582
5	0.3338	25.7316	1.3407
6	0.3146	25.2268	1.2221
7	0.2936	23.9865	1.0373
8	0.2049	24.0376	1.1164
9	0.2664	24.0156	1.0540
10	0.2204	24.3161	0.8609

Note: best score in **bold**.

Table 5.6: Results of Dataset Brazil with Bk-Means algorithm without any outlier detection algorithm.

Number of Clusters	Silhouette	CH	DB
2	**0.7871**	**34.8118**	**0.1404**
3	0.20636	25.5402	1.6090
4	0.2490	26.4268	1.5413
5	0.2517	21.7940	1.8125
6	0.2558	19.7783	1.8370
7	-0.0021	17.0498	2.0701
8	0.0081	16.8146	1.9505
9	0.0123	15.6649	1.9921
10	0.0357	14.7224	1.9394

Note: best score in **bold**.

5.5 Combination of Outlier Detectors and Clustering algorithms

In this section, the outlier detector and clustering algorithms will be combined and an analysis of the effect the removal of outliers have on the clustering validation metrics will

Table 5.7: Results of Dataset Brazil with Mean-Shift algorithm without any outlier detection algorithm.

Number of Clusters	Silhouette	CH	DB	Bandwidth
2	**0.7871**	**34.8118**	**0.1404**	1.00 - 2.10
3	0.5808	22.5971	0.2416	0.95
4	0.5709	19.0024	0.2486	0.9
5	0.5510	17.3689	0.2549	0.8 - 0.85

Note: best score in **bold**.

be done.

5.5.1 Local Outlier Factor

This section displays the results obtained utilizing the LOF anomaly detector algorithm together with different clustering techniques. These algorithms are applied in the two available datasets.

Dataset Peru

Table 5.10, Table 5.11 and Table 5.12 have the results with the k-Means, Bk-Means and Mean-Shift algorithm, respectively. After removing outliers detected with LOF algorithm. The LOF threshold used for this test was set at 40. All the best scores were generated using parameter values of 48 to 50 for the number of neighbors, as proposed in Table 4.5.

Analyzing how balanced were the clusters generated with the 48 number of neighbors parameter, the highest Silhouette and DB scores were results that one cluster had several of the data points and the other less than four points. This happened with all clustering algorithms, the number of data points in each cluster is visible in Table 5.9. When taking into consideration the best CH score the clusters were more balanced (except for the Bk-Means result), with the k-Means algorithm obtaining the values in Table 5.8.

Table 5.8: Number of data points in each cluster using k-Means with K = 5 and number of neighbors 48 and Dataset Peru.

Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
111	3	738	1	1

Table 5.9: Number of data points in each cluster according to each clustering algorithm with a number of neighbors 48 and highest Silhouette score with Dataset Peru.

Algorithm	**Cluster 0**	**Cluster 1**
k-Means	853	1
Bk-Means	851	3
Mean-Shift	853	1

Table 5.10: Results of Dataset Peru with the LOF and k-Means algorithms.

Number of Clusters	**Silhouette**	**CH**	**DB**	**Number of Neighbors**
2	**0.9144**	153.9543	**0.0579**	48 - 50
3	0.5281	78.5206	1.4425	40 - 41
4	0.40611	95.6107	1.7438	48 - 50
5	0.4459	**194.6502**	0.9494	48 - 50
6	0.1421	53.5531	1.8756	45
7	0.1385	47.9470	2.0144	45
8	0.1278	59.3182	1.8451	48 - 50
9	0.1326	56.1703	1.8189	48 - 50
10	0.1407	92.8888	1.5871	48 - 50

Note: best score in **bold**.

Table 5.11: Results of Dataset Peru with the LOF and Bk-Means algorithms.

Number of Clusters	**Silhouette**	**CH**	**DB**	**Number of Neighbors**
2	**0.8431**	**118.7449**	**0.5298**	48 - 50
3	0.5281	78.5206	1.4425	40 - 41
4	0.40611	95.6107	1.7438	48 - 50
5	0.1700	69.0409	2.1537	47
6	0.1421	53.5531	1.8756	45
7	0.1385	47.9470	2.0144	45
8	0.1278	59.3182	1.8451	48 - 50
9	0.1326	56.1703	1.8189	48 - 50
10	0.1407	92.8888	1.5871	48 - 50

Note: best score in **bold**.

Table 5.12: Results of Dataset Peru with the LOF and Mean-Shift algorithms.

N Clusters	Silhouette	CH	DB	Bandwidth	N Neighbors
2	**0.9021**	91.3850	**0.0651**	1.00 - 1.40	10 - 26
3	0.9020	**155.5529**	0.0629	0.80 - 0.95	48 - 50
4	0.8794	127.4863	0.06714	0.80 - 0.95	47
5	0.8704	92.1759	0.0768	0.85 - 0.90	33 - 34
6	0.8694	94.2228	0.5038	0.80	30 - 32

Note: best score in **bold**.

Dataset Brazil

When analyzing the number of anomalies detected utilizing the LOF algorithm on Dataset Brazil the LOF threshold of 40 and 20 were not enough to identify outliers, because of this the threshold used set at 10. Tables 5.13, 5.14 and 5.15 show the best results obtained when combining the outlier detector with clustering algorithm k-Means, Bk-Means and Mean-Shift.

Similar to results found with Dataset Peru the best scores, Silhouette and DB, are situations where here are only two clusters with one cluster containing several data points and the other only one. All clustering algorithms scored the same exact best values for all performance metrics. Only when lowering the LOF threshold to four does the value of the performance metrics change. The clusters created are more balanced with the k-Means and Bk-Means algorithms.

Table 5.13: Results of Dataset Brazil with the LOF algorithm and the k-Means algorithm.

Number of Clusters	Silhouette	CH	DB	Number of Neighbors
2	**0.7932**	**37.0483**	**0.1362**	33 - 50
3	0.5394	29.5566	0.9825	36 - 50
4	0.4808	25.7390	1.1582	10 - 32
5	0.3376	25.1029	1.1920	33 - 35
6	0.3260	23.9474	1.1853	33 - 35
7	0.3222	22.4708	1.2451	33 - 35
8	0.3098	25.1396	0.8512	36 - 50
9	0.2664	24.0156	1.0540	10 - 32
10	0.2204	24.3161	0.8609	10 - 32

Note: best score in **bold**.

Table 5.14: Results of Dataset Brazil with the LOF algorithm and the Bk-Means algorithm.

Number of Clusters	Silhouette	CH	DB	Number of Neighbors
2	**0.7932**	**37.0483**	**0.1362**	33 - 50
3	0.4683	29.1224	1.1953	33 - 35
4	0.316	26.8106	1.447	33 - 35
5	0.2795	25.9542	1.4013	36 - 50
6	0.2558	9.7783	1.8370	10 - 32
7	0.2301	20.5576	1.6589	36 - 50
8	0.1750	17.8832	1.5451	33 - 50
9	0.1441	19.0889	1.4688	36 - 50
10	0.0836	18.0642	1.5359	36 - 50

Note: best score in **bold**.

Table 5.15: Results of Dataset Brazil with the LOF algorithm and the Mean-Shift algorithm.

N Clusters	Silhouette	CH	DB	Bandwidth	N Neighbors
2	**0.7932**	**37.0483**	**0.1362**	1.05 - 2.10	36 - 50
3	0.5923	24.2207	0.2335	1.00	36 - 50
4	0.5878	20.5665	0.2397	0.90 - 0.95	36 - 50
5	0.5680	18.9368	0.2459	0.75 - 0.85	36 - 50
6	0.5137	17.3262	0.2658	0.70	36 - 50
7	0.4980	19.6185	0.3683	0.60 - 0.65	36 - 50
8	0.4651	21.5736	0.5139	0.50 - 0.55	36 - 50
9	0.4698	20.9285	0.5095	0.50 - 0.55	10 - 32

Note: best score in **bold**.

5.5.2 Isolation Forest

This section presents results combining the Isolation Forest outlier detector with the different clustering algorithms and the Datasets Peru and Brazil.

Dataset Peru

Results with Isolation Forest anomaly detector and Dataset Peru are visible in Tables 5.16, 5.17 and 5.18 for clustering algorithms k-Means, Bk-Means and Mean-Shift, respectively. Similar to the scores obtained with the LOF algorithm, k-Means and Mean-Shift scored the best values for the performance metrics.

Analyzing how balanced the clusters generated were, the highest Silhouette and DB scores the lower the number of clusters and generally, there is one cluster with the great majority of data points and the rest of the clusters have one data point, this happened with all clustering algorithms. On the other hand, the Bk-Means algorithm typically generates clusters that are more balanced but scores lower values for the performance metrics. A similar result was found when analyzing how balanced the clusters were with the best Calinski-Harabasz scores without any anomaly detector and the k-Means algorithm, more balanced were the clusters.

Table 5.16: Results of Dataset Peru with Isolation Forest and k-Means algorithms.

N Clusters	Silhouette	CH	DB	N Trees
2	0.9189	154.0773	0.0546	62
3	**0.9194**	188.1034	**0.0535**	62
4	0.9075	204.4792	0.0570	56
5	0.8893	210.2034	0.0621	64
6	0.8808	192.7234	0.0655	60
7	0.3434	201.6246	0.7024	60
8	0.330	214.5511	0.6723	82
9	0.2591	**221.8445**	0.7950	62
10	0.2548	216.9797	0.7013	60

Note: best score in **bold**.

Dataset Brazil

Results with Isolation Forest anomaly detector and Dataset Brazil are visible in Tables 5.19, 5.20 and 5.21 have the results for clustering algorithms k-Means, Bk-Means and Mean-Shift, respectively.

Table 5.17: Results of Dataset Peru with Isolation Forest and Bk-Means algorithms.

N Clusters	Silhouette	CH	DB	N Trees
2	**0.9189**	**154.0773**	**0.0546**	62
3	0.3325	45.1831	2.3366	58
4	0.3302	40.7073	2.0827	70 - 98
5	0.2558	66.4102	1.8837	64
6	0.2606	58.4166	1.9691	64
7	0.2615	93.2923	1.6426	64
8	0.1577	38.8338	1.9328	60
9	0.1670	66.3088	1.7188	60
10	0.1744	104.3715	1.7985	70 - 98

Note: best score in **bold**.

Table 5.18: Results of Dataset Peru with Isolation Forest and Mean-Shift algorithms.

N Clusters	Silhouette	CH	DB	Bandwidth	N Trees
2	0.9188	169.6362	0.0547	1.0 - 1.4	50 - 54
3	**0.9194**	188.1034	**0.0535**	1.00 - 1.40	62
4	0.8885	170.3483	0.0657	0.75 - 0.95	50 - 54
5	0.8898	**212.4307**	0.06189	0.75 - 0.95	62
6	0.8476	198.4412	0.07312	0.3 - 0.7	62
7	0.8465	185.9190	0.07379	0.3 - 0.7	60
8	0.7049	170.9883	0.15430	0.2 - 0.25	60
9	0.5133	144.1741	0.2933	0.15	62
10	0.5125	143.0178	0.2737	0.15	60

Note: best score in **bold**.

Analyzing how balanced the generated clusters were, the k-Means with $K = 2$ created one cluster containing 23 data points and the others 79.

Table 5.19: Results of Dataset Brazil with Isolation Forest and k-Means algorithms.

N Clusters	Silhouette	CH	DB	N Trees
2	**0.5608**	14.0976	**0.4520**	90
3	0.3307	16.4206	1.4289	86 - 100
4	0.3431	**17.6677**	1.3458	86 - 100
5	0.3404	15.7453	0.9813	70 - 96
6	0.3168	16.7392	1.0744	60 - 62
7	0.3539	15.3244	0.8043	70 - 96
8	0.2288	15.9375	1.1633	70 - 96
9	0.1919	15.9827	1.3044	60 - 62
10	0.1916	16.1465	1.0671	70 - 96

Note: best score in **bold**.

Table 5.20: Results of Dataset Brazil with Isolation Forest and Bk-Means algorithms.

N Clusters	Silhouette	CH	DB	N Trees
2	0.4440	4.1475	**0.4187**	50 - 72
3	**0.4638**	10.7888	1.2955	50 - 72
4	0.2982	13.7102	1.4957	50 - 72
5	0.1779	**13.8029**	1.6733	60 - 62
6	0.1894	13.1310	1.7721	60 - 62
7	0.1262	13.3328	1.6553	90
8	0.1325	12.6428	1.6667	90
9	0.1461	12.4098	1.6382	90
10	0.1549	12.3689	1.5393	90

Note: best score in **bold**.

5.5.3 Final Remarks

Overall results when testing with Datasets Peru and Brazil differ in significant ways, since the former generally produces higher scores but the clusters created are highly unbalanced and the latter typically generates clusters with lower performance scores but are more balanced.

Both Isolation Forest and LOF algorithms were able to find most of the outliers and when paired with the clustering algorithms generate similar results. The Isolation Forest

Table 5.21: Results of Dataset Brazil with Isolation Forest and Mean-Shift algorithms.

N Clusters	Silhouette	CH	DB	Bandwidth	N Trees
2	**0.5608**	7.5188	**0.3151**	0.65 - 0.70	90
3	0.5157	6.7960	0.3369	0.65 - 0.70	70 - 96
4	0.5087	**14.0716**	0.6111	0.50 - 0.55	90
5	0.5068	13.8733	0.6200	0.50 - 0.55	60 - 62
6	0.4689	10.8949	0.5079	0.45	90
7	0.4577	10.1616	0.4833	0.45	70 - 96
8	0.4599	10.0878	0.4604	0.45	60 - 62
9	0.3969	10.8608	0.6827	0.40	86 - 100
10	0.4007	10.6727	0.5001	0.40	60 - 62

Note: best score in **bold**.

algorithm was able to score higher in some tests, this could be related to how aggressive it is when detecting what is an outlier. This is inline of what was found in Section 5.3, which shows that Isolation Forest was able to detect a higher number of outliers. A possible way to improve Local Outlier Factor results, compared to Isolation Forest, would be to use lower Local Outlier Factor thresholds.

5.6 Profile Visualization

For generating the customer behavior clustering results, those with good Silhouette and DB scores generally do not produce balanced clusters, therefore do not generate customer behavior graphs with meaningful information. In order to generate better graphs a mix of highly balanced clusters with good performance metrics generate better visualizations.

The group of graphs in the visualization show how different clusters behave according to different features in the dataset. Figure 5.5 shows a graph generated when utilizing outlier detector Isolation Forest with the number of trees parameter 86 and clustering algorithm k-Means with $K = 4$ using Dataset Brazil. This combination generated metrics of CH of 17.6677, DB of 1.3458 and Silhouette Index of 0.3431. Table 5.22 presents the number of data points in each cluster.

Each graph in the visualization contains a title, the ones that start with the prefix *n_* are related to the number of transactions of that features and the ones that end in the suffix *_total* are related to the total amount spent in that feature. The features *n_bills* and *bills_total* are from the total number of transactions and total amount spent.

The expert analyzed Visualization 5.5 and gave an overview of each graph generated and how people in different clusters behave compared to each other, his insight took into

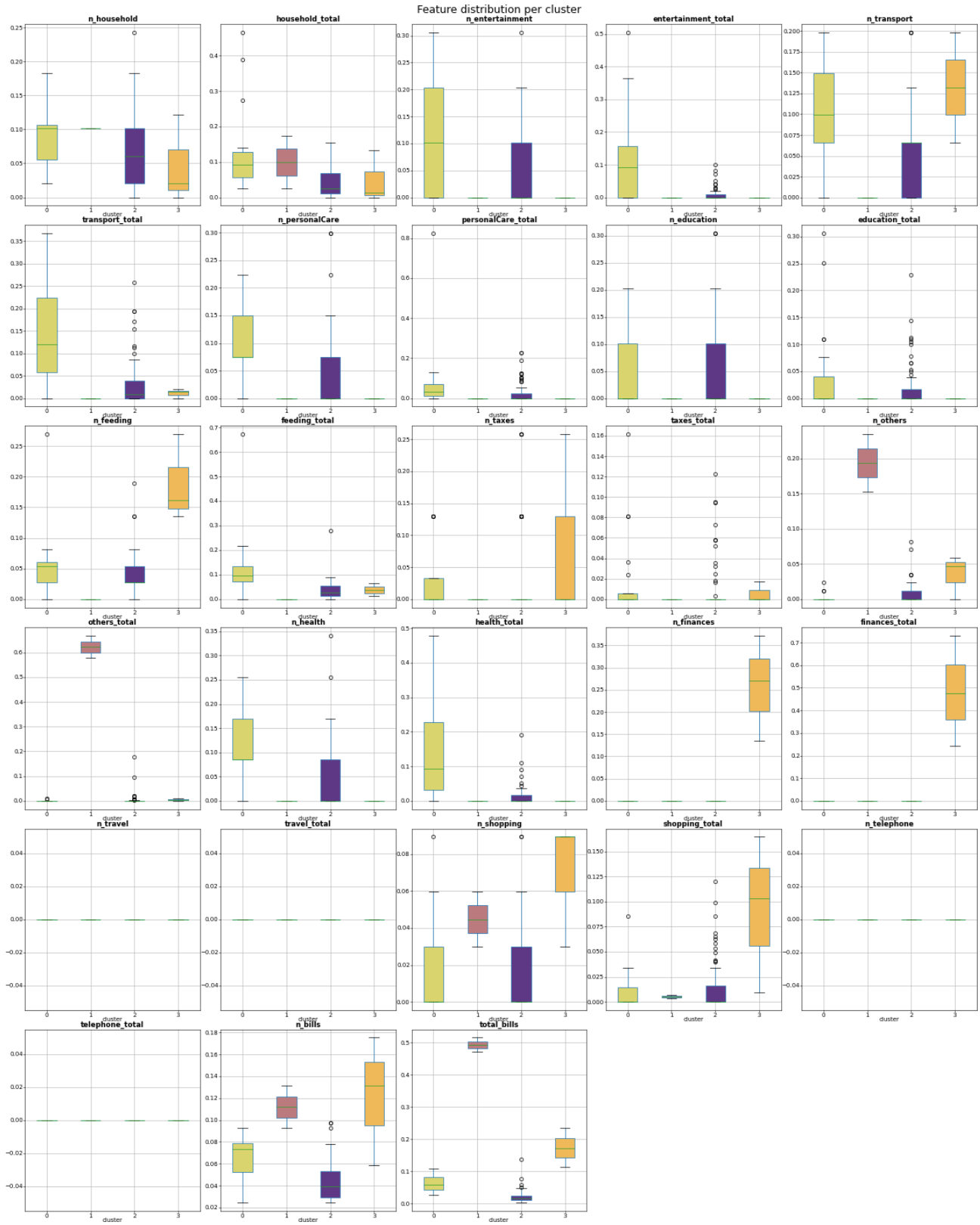

Figure 5.5: Group of graphs that show cluster behavior for each feature.

Table 5.22: Number data points in each cluster.

Cluster	Number of Data Points
0	20
1	2
2	75
3	5

consideration the background information available for the current Brazilian financial landscape for underbanked and unbanked people.

Regarding the quality of the visualization generated, the expert found the y axis scale changing from one graph to another to be a problem. Furthermore the initial idea was to have an analysis of a cluster by cluster basis, the expert found that doing an analysis graph by graph would be a better approach.

5.7 Ok-Means

The Ok-Means algorithm was compared to the k-Means algorithm with and without using the Isolation Forest anomaly detector along with Dataset Brazil as input data. The same initial points and maximum iteration number were used for each clustering algorithm, the K parameter was varied from 2 to 8 and the iteration number 500. The k-Means implementation used is the one found in the SK-Learn Python library[3]. Regarding the anomaly detector, the parameter number of trees is set to 90.

Table 5.25 has the comparison of values when using the k-Means algorithm without outlier detection and when using Ok-Means. For this test the threshold is set at 0.9 and the starting iteration is 200. Overall the Ok-Means algorithm showed a slight advantage in almost all metrics and in only some instances loses to the standard k-Means algorithm. Table 5.23 shows the number users in each cluster when changing the paramenter K, it is possible to notice that the algorithm is able to remove values but unbalanced clusters are still being generated when comparing to the results obtained with the Isolation Tree + k-Means combination, the number of points in each cluster in visible in Table 5.24.

Table 5.26 has the results of the comparison between the Ok-Means and k-Means + Isolation Forest combination. In some results, the Ok-Means algorithm is able to score better but the k-Means combination generates more balanced clusters.

When analyzing the behavior of the Ok-Means algorithm, setting the initial iteration number has a small effect on the number of anomalies removed, changing only this value

[3]Library [29] available at https://scikit-learn.org/stable/preface.html.

Table 5.23: Number of users in each cluster with Dataset Brazil and Ok-Means.

N Clusters	Cluster information
2	0: 107, 1: 1
3	0: 1, 1: 106, 2: 1
4	0: 19, 1: 81, 2: 1, 3: 1
5	0: 1, 1: 2, 2: 1, 3: 99, 4: 1
6	0: 67, 1: 1, 2: 1, 3: 10, 4: 1, 5: 1
7	0: 1, 1: 1, 2: 60, 3: 26, 4: 2, 5: 1, 6: 1
8	0: 33, 1: 15, 2: 1, 3: 1, 4: 10, 5: 33, 6: 1, 7: 1

Table 5.24: Number of users in each cluster with Dataset Brazil, Isolation Tree and k-Means algorithms.

N Clusters	Cluster information
2	0: 97, 1: 2
3	0: 20, 1: 77, 2: 2
4	0: 72, 1: 3, 2: 22, 3: 2
5	0: 10, 1: 58, 2: 26, 3: 2, 4: 3
6	0: 30, 1: 2, 2: 17, 3: 3, 4: 1, 5: 46
7	0: 62, 1: 2, 2: 3, 3: 10, 4: 20, 5: 1, 6: 1
8	0: 14, 1: 5, 2: 52, 3: 14, 4: 3, 5: 2, 6: 8, 7: 1

Table 5.25: Results of Dataset Brazil with Ok-Means and k-Means algorithms.

Algorithm	N Clusters	Silhouette	CH	DB
k-Means	2	0.7871	34.8118	0.1404
Ok-Means	2	**0.7920**	**37.1286**	**0.1370**
k-Means	3	0.5755	22.5313	0.2424
Ok-Means	3	**0.5846**	**24.1824**	**0.2362**
k-Means	4	0.2434	18.3154	1.4278
Ok-Means	4	**0.2543**	**26.7572**	**1.0898**
k-Means	5	0.3926	15.6627	0.8430
Ok-Means	5	**0.4315**	**16.3335**	**0.7157**
k-Means	6	**0.4251**	19.2370	1.0359
Ok-Means	6	0.3240	**39.4288**	**0.6857**
k-Means	7	0.2151	23.4194	1.5064
Ok-Means	7	**0.2187**	**27.5843**	**0.9909**
k-Means	8	0.0485	17.4907	1.6469
Ok-Means	8	**0.0581**	**19.5680**	**1.3649**

Note: best score in **bold**.

Table 5.26: Results of Dataset Brazil with Ok-Means and k-Means + Isolation Forest algorithms.

Algorithm	**N Clusters**	**Silhouette**	**CH**	**DB**
k-Means and Isolation Forest	2	0.4642	4.5932	0.4004
Ok-Means	2	**0.7920**	**37.1286**	**0.1370**
k-Means and Isolation Forest	3	0.2673	14.6823	1.8446
Ok-Means	3	**0.5846**	**24.1824**	**0.2362**
k-Means and Isolation Forest	4	0.0624	8.1084	1.9988
Ok-Means	4	**0.2543**	**26.7572**	**1.0898**
k-Means and Isolation Forest	5	0.3256	12.0457	1.2183
Ok-Means	5	**0.4315**	**16.3335**	**0.7157**
k-Means and Isolation Forest	6	**0.2070**	12.3718	1.3396
Ok-Means	6	0.3240	**39.4288**	**0.6857**
k-Means and Isolation Forest	7	0.1779	13.4441	1.5797
Ok-Means	7	**0.2187**	**27.5843**	**0.9909**
k-Means and Isolation Forest	8	0.0224	8.4286	1.8039
Ok-Means	8	**0.0581**	**19.5680**	**1.3649**

Note: best score in **bold**.

to 1 and K to 4 the amount of anomalies changes from 3 to 8. Changing the threshold to 0.8 increases the number of anomalies removed from 3 to 4. To better visualize this, Figure 5.6 shows the behavior of the Ok-Means algorithm when changing the threshold and the number of anomalies removed. When nearing the value of 1.0 it converges to less outliers.

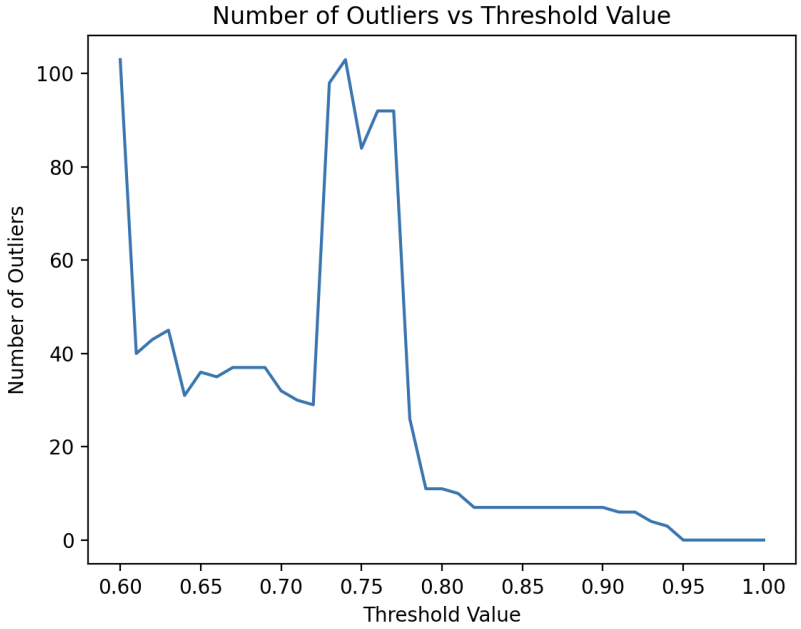

Figure 5.6: Number of outliers removed using the Ok-Means algorithm with $K = 4$, start iteration = 200 and maximum iteration = 500.

Chapter 6

Conclusion and Future Work

In this study, a proposed method using different outlier detection and clustering algorithms has shown promising results when using categorized transaction data. That may be a viable structure to cluster such information and generate spending pattern profiles.

Internal validation metrics along with balancing analysis were used to identify best combinations of outlier detection and clustering algorithms. During testing, better metrics alone did not result in the finest clustering results but their use along with an empirical balancing analysis were found to be a positive way of generating clusters. When comparing the metrics used, higher Calinski-Harabasz scores resulted in more balanced results when comparing to the other performance metrics in some instances. Overall the greater the performance metrics and the more balanced the clusters the more information was able to be extracted from each cluster.

A spending profile visualization was considered and verified with an expert in the area, the visualization was used to successfully identify spending behavior generated with the proposed method and give more insightful information. The guidance of more experts in the field along with the feedback received can be of service to optimize the existing visualization or propose new formats to be used.

The proposed Ok-Means algorithm has found to give better results when utilizing internal validation metrics and when compared to the k-Means and k-Means + Isolation Forest combination. On the other hand, this algorithm is not able to deal with creating balanced clusters. Tests with use of the k-Means++ centroid initialization algorithm could also possibly generate better results [2].

For future work, other datasets of different types of data could be used to better test the performance of the Ok-Means algorithm in order to understand better applications of it. Some mathematical work can be added to prove or disprove effectiveness along with a comparison of other variations of the k-Means algorithm [17].

Another important matter is the measuring of balanced clusters, a study of possible measurements or approaches of balanced clusters could help during the optimization step. The addition of some performance metric related balancing clusters is something to investigate.

References

[1] Allegue, S., Abdellatif, T., and Bannour, K. (2020). RFMC: a spending-category segmentation. In *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE. 16, 17, 19, 20

[2] Arthur, D. and Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA. Society for Industrial and Applied Mathematics. 8, 22, 49

[3] Bock, H.-H. (2008). Origins and extensions of the k-means algorithm in cluster analysis. *Electronic Journal for History of Probability and Statistics*, 4(2):1–18. 8

[4] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104. 5

[5] Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27. 12

[6] Comaniciu, D. and Meer, P. (1999). Mean shift analysis and applications. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1197–1203 vol.2. 12

[7] Comaniciu, D. and Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619. 10

[8] Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227. 13

[9] Di, J. and Gou, X. (2018). Bisecting k-means algorithm based on k-valued selfdetermining and clustering center optimization. *J. Comput.*, 13(6):588–595. 10, 11

[10] Ernawati, E., Baharin, S. S. K., and Kasmin, F. (2021). A review of data mining methods in RFM-based customer segmentation. *Journal of Physics: Conference Series*, 1869(1):012085. 15

[11] Fukunaga, K. and Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40. 10

[12] Gan, G., Ma, C., and Wu, J. (2007). In *Data Clustering: Theory, Algorithms, and Applications*, pages 299–320. Society for Industrial and Applied Mathematics. 2, 12

[13] Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1):3–42. 24

[14] Holm, M. (2018). Machine learning and spending patterns : A study on the possibility of identifying riskily spending behaviour. 18, 19, 20

[15] Hu, X., Shi, Z., Yang, Y., and Chen, L. (2020). Classification method of internet catering customer based on improved rfm model and cluster analysis. In *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, pages 28–31. IEEE. 2, 16, 19, 20

[16] Huang, Y., Zhang, M., and He, Y. (2020). Research on improved RFM customer segmentation model based on k-means algorithm. In *2020 5th International Conference on Computational Intelligence and Applications (ICCIA)*. IEEE. 17, 19, 20

[17] Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., and Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622:178–210. 49

[18] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323. 8

[19] Jie, C., Jiyue, Z., Junhui, W., Yusheng, W., Huiping, S., and Kaiyan, L. (2020). Review on the research of k-means clustering algorithm in big data. In *2020 IEEE 3rd International Conference on Electronics and Communication Engineering (ICECE)*. IEEE. 26

[20] Jo-Ting, W., Shih-Yen, L., and Hsin-Hung, W. (2010). A review of the application of rfm model. *African Journal of Business Management*, 4(19):4199–4206. 1

[21] Kansal, T., Bahuguna, S., Singh, V., and Choudhury, T. (2018). Customer segmentation using k-means clustering. In *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*. IEEE. 18, 20

[22] Lefait, G. and Kechadi, T. (2010). Customer segmentation architecture based on clustering techniques. In *2010 Fourth International Conference on Digital Society*. IEEE. 15, 20

[23] Li, H. and Wu, W. (2021). Construction of chinese national geography APP user operation strategy based on RFM model. In *2021 2nd International Conference on E-Commerce and Internet Technology (ECIT)*. IEEE. 1

[24] Li, Y., Chu, X., Tian, D., Feng, J., and Mu, W. (2021). Customer segmentation using k-means clustering and the adaptive particle swarm optimization algorithm. *Applied Soft Computing*, 113:107924. 17, 20

[25] Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE. 2, 6, 7

[26] Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9:381–386. 1

[27] Maryani, I., Riana, D., Astuti, R. D., Ishaq, A., Sutrisno, and Pratama, E. A. (2018). Customer segmentation based on RFM model and clustering techniques with k-means algorithm. In *2018 Third International Conference on Informatics and Computing (ICIC)*. IEEE. 16, 20

[28] Parikh, Y. and Abdelfattah, E. (2020). Clustering algorithms and rfm analysis performed on retail transactions. In *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 0506–0511. IEEE. 16, 20

[29] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. 24, 29, 45

[30] Pondel, M. and Korczak, J. (2018). Collective clustering of marketing data—recommendation system upsaily. In *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems*. IEEE. 17, 20

[31] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65. 12

[32] Umuhoza, E., Ntirushwamaboko, D., Awuah, J., and Birir, B. (2020). Using unsupervised machine learning techniques for behavioral-based credit card users segmentation in africa. *SAIEE Africa Research Journal*, 111(3):95–101. 1, 2, 18, 20

[33] Wu, J. and Lin, Z. (2005). Research on customer segmentation model by clustering. In *Proceedings of the 7th international conference on Electronic commerce - ICEC '05*. ACM Press. 18, 19, 20

[34] Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193. 11

[35] Zakrzewska, D. and Murlewski, J. (2005). Clustering algorithms for bank customer segmentation. In *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*. IEEE. 2, 18, 19, 20