Incorporating K-means, Hierarchical Clustering and PCA in Customer Segmentation

Article · February 2021

DOI: 10.12691/jcd-3-1-3

Azad Abdulhafedh
University of Missouri

Journal of City and Development, 2021, Vol. 3, No. 1, 12-30
Available online at http://pubs.sciepub.com/jcd/3/1/3
Published by Science and Education Publishing
DOI:10.12691/jcd-3-1-3

Incorporating K-means, Hierarchical Clustering and PCA in Customer Segmentation

Azad Abdulhafedh[*]

University of Missouri, USA
*Corresponding author: asa8cd@mail.missouri.edu

Received January 04, 2021; Revised January 25, 2021; Accepted February 02, 2021

Abstract This paper addresses the use of clustering algorithms in the customer segmentation to define a marketing strategy of a credit card company. Customer segmentation divides customers into groups based on common characteristics, which is useful for banks, businesses, and companies to improve their products or service opportunities. The analysis explores the applications of the K-means, the Hierarchical clustering, and the Principal Component Analysis (PCA) in identifying the customer segments of a company based on their credit card transaction history. The dataset used in the project summarizes the usage behavior of 8950 active credit card holders in the last 6 months, and our aim is to perform customer segmentation in the most accurate way using clustering techniques. The project uses two approaches for customer segmentation: first, by considering all variables in the clustering algorithms using the Hierarchical clustering and the K-means. Second, by applying the dimensionality reduction through Principal Component Analysis (PCA) to the dataset, then identifying the optimal number of clusters, and repeating the clustering analysis with the updated number of clusters. Results show that the PCA can effectively be employed in the clustering process as a check tool for the K-means and Hierarchical clustering.

Keywords: *K-means, Hierarchical Clustering, Principal Component Analysis, Agglomerative hierarchical clustering, scree plot, Silhouette average width, Davies-Bouldin Index, Dunn index, customer segmentation*

Cite This Article: Azad Abdulhafedh, "Incorporating K-means, Hierarchical Clustering and PCA in Customer Segmentation." *Journal of City and Development*, vol. 3, no. 1 (2021): 12-30. doi: 10.12691/jcd-3-1-3.

1. Introduction

Unsupervised learning is a process for gaining meaningful insights by summarizing data in innovative ways. As opposed to supervised learning methods that predict a target of interest; in an unsupervised learning, no single feature is more important than any other. Thus, with unsupervised learning, there are inputs but no supervising output [1,2,3]. Customer Segmentation (also called market segmentation) is one the most important applications of the unsupervised learning methods in data science and machine learning. Customer Segmentation is the process of dividing customers into several groups that share common characteristics relevant to marketing such as gender, age, interests, and miscellaneous spending habits. Segmentation process can help businesses and companies understand their customer groups, target the right groups, and develop effective marketing strategies for different targeted groups. Clustering techniques are the most appropriate methods that enable businesses and companies to identify segments or groups of customers in order to target the potential user base. Customer segmentation can be performed using a variety of different customer characteristics. The most common types are customer's geographical regions, customer's demographics (e.g., age, gender, marital status, income), customer's psychographics (e.g., values, interests, lifestyle, group affiliations), and purchase behavior (e.g., previous purchases, shipping preferences). Customers data usually contain observations from millions of customers; however, these customers may only belong to a few segments: customers are similar within each segment but different across segments. Grouping data with similar characteristics into clusters is called cluster analysis. This is similar to classification; except we do not have a labelled dataset to use for training. Data points are simply grouped based on how similar they are to each other. Since there is not a response variable, this is an unsupervised method, which implies that it seeks to find relationships between the observations without being trained by a response variable [1,4,5,6].

2. Data

A credit card company has collected over the time data about their customers' accounts. The data has various facts related to the customers, such as their balances, purchases, cash advances, credit scores, etc. The management team was willing to make meaningful insights from the data, and then develop strategies to target segments of customers in order to increase credit card sales, and in turn to increase the revenue. The dataset used in this project is publicly available at the Kaggle website (http://www.Kaggle.com/data) and is created in 2018. This dataset consists of behavioral and non-labeled data related to credit cards transactions. The main goal is to perform customer segments that is best fitted to the data by implementing the clustering analysis. The dataset has high-dimensionality and possesses correlated variables. The data consists of **8950** observations (rows) related to the credit card holders as registered users of a European credit card company, and **18** behavioral variables (columns). The description of these variables is shown below:

1. **CUSTID**: Identification of Credit Card holder (Categorical).
2. **BALANCE**: Balance amount left in their account to make purchases.
3. **BALANCE_FREQUENCY**: How frequently the Balance is updated, score between 0 and 1 (1 = frequently updated, 0 = not frequently updated).
4. **PURCHASES**: Amount of purchases made from account.
5. **ONE-OFF_PURCHASES**: Maximum purchase amount done in one-go.
6. **INSTALLMENTSPURCHASES**: Amount of purchase done in installments.
7. **CASHADVANCE**: Cash in advance given by the user.
8. **PURCHASES_FREQUENCY**: How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased).
9. **ONE-OFFPURCHASES_FREQUENCY**: How frequently Purchases are happening in one-go (1 = frequently purchased, 0 = not frequently purchased).
10. **PURCHASE_SINSTALLMENTS_FREQUENCY**: How frequently purchases in installments are being done (1 = frequently done, 0 = not frequently done).
11. **CASHADVANCE_FREQUENCY**: How frequently the cash in advance being paid.
12. **CASHADVANCE_TRX**: Number of Transactions made with "Cash in Advanced".
13. **PURCHASES_TRX**: Number of purchase transactions made.
14. **CREDITLIMIT**: Limit of Credit Card for user.
15. **PAYMENTS**: Amount of Payment done by user.
16. **MINIMUM_PAYMENTS**: Minimum amount of payments made by user.
17. **PRCFULLPAYMENT**: Percent of full payment paid by user.
18. **TENURE**: Tenure of credit card service for user.

3. Exploratory Data Analysis (EDA)

To better understand the data, it is important to explore the dataset by conducting an exploratory data analysis (EDA) using R, and Python software's. First, a summary of statistical description of the dataset is obtained. Then null values are identified and replaced with the mean of the column. Next, correlation matrix is produced [1,2,3,6,7].

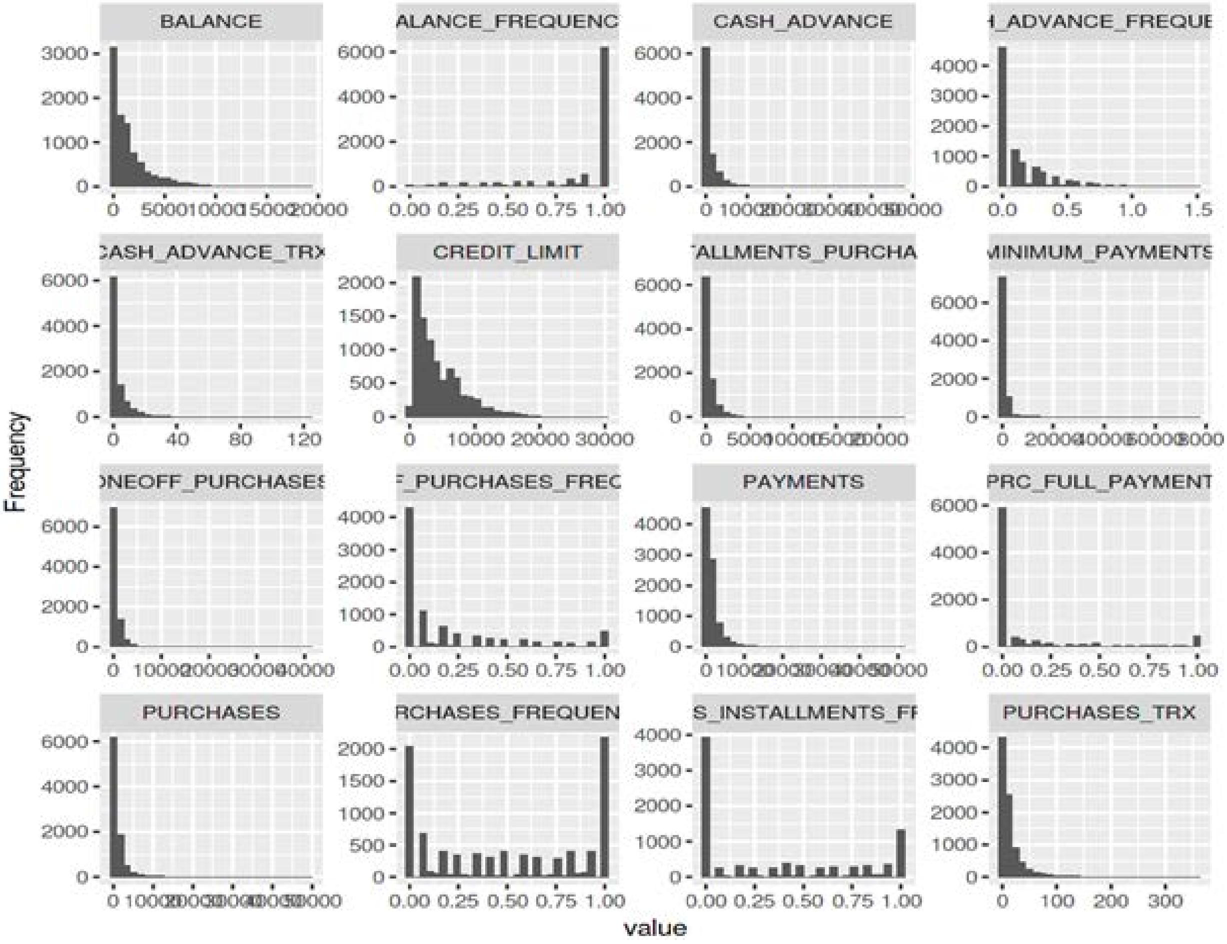


Figure 1. frequency histograms of some main variables in the dataset

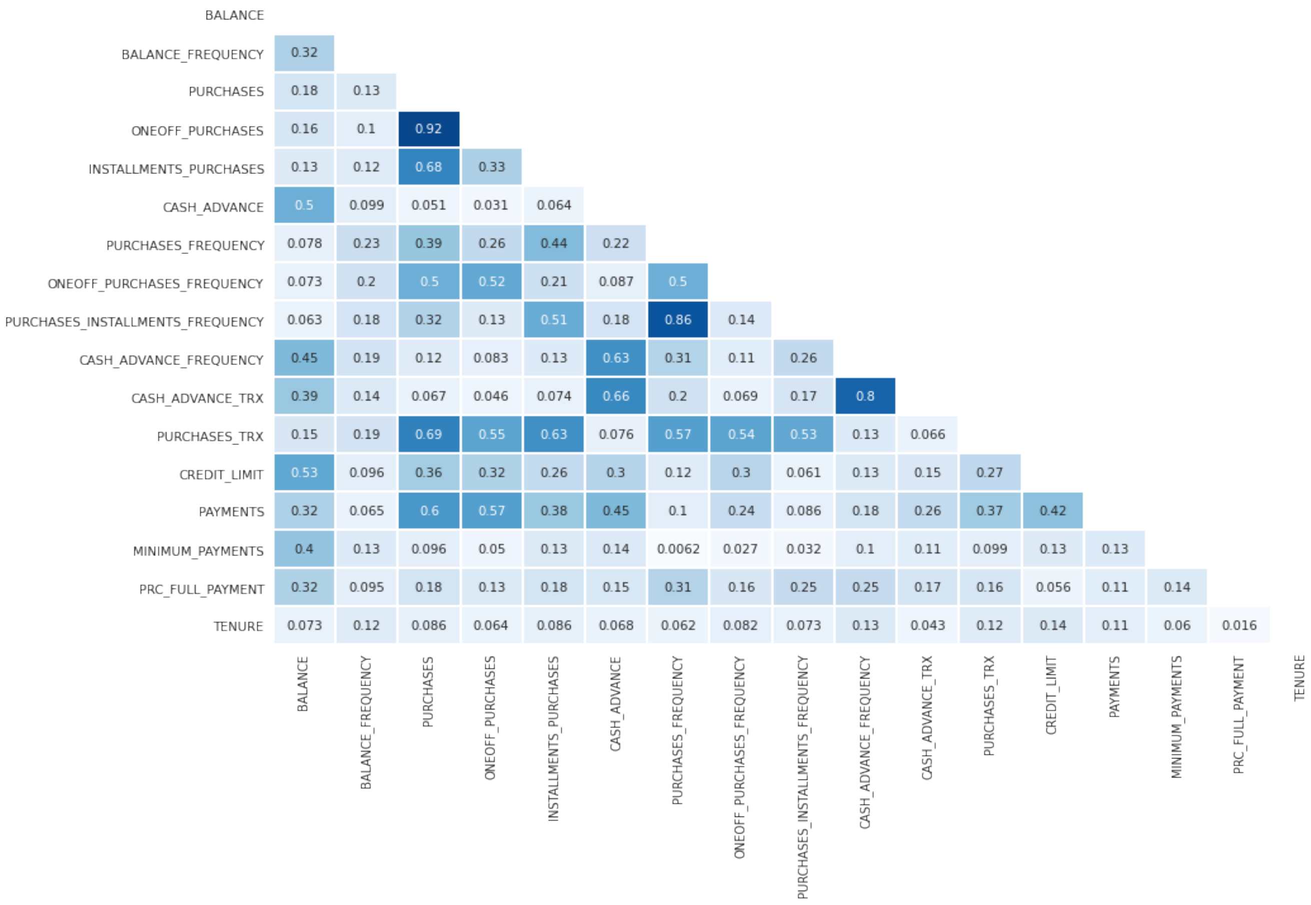


Figure 2. the correlation matrix of the variables in the dataset

The following steps are implemented in the EDA using R, and Python software's:

- **Summary Statistics:** Main statistical information of the data is obtained: minimum, maximum values as well as standard deviations are found.
- **Checking for Null Values**: The data contains (314) null values that are replaced with mean values from their respective columns. The variable "MIMIMUM_PAYMENTS" has 313 null values. The Null values are imputed with the mean (864.21). The variable "CREDIT_LIMIT" has only (1) null value and the null value is imputed with the mean (4494.45).
- There are different options to work with **null values**. One would be to drop them; another option would be to replace them with the mean or median values. In our case, we filled them with the mean value of the column, as both (minimum payments and credit limit) are continuous variables.
- **Frequency Histograms**: These plots help to understand how the values on different variables are distributed, as shown in Figure 1.
- **Dropping the CUSTID**: This variable is categorical presenting the customer's ID and has very little effect on the analysis, so it is dropped from the analysis.
- **Correlation Matrix:** This matrix helps visualizing how the different variables are correlated as shown in Figure 2.

Inspecting the relationship between variables is very important because highly correlated or collinear variables may disrupt the algorithm and eventually affect distinguished clusters. High correlation between two variables means that they have similar trends and are likely to carry similar information. If two variables are perfectly correlated, the concept represented by both variables is represented twice in the data. From the correlation matrix, we can see that variables (PURCHASES, ONE-OFF _PURCHASES) seem to be highly correlated (correlation of 0.92). The variables (PURCHASE_FREQUENCY, PURCHASE_INSTALLMENTS_FREQUENCY) are highly correlated (0.86). Moreover, high correlation occurs in case of variables CASH_ADV, CASH_ADV_FREQ and CASH_ADV_TRX and also PURCH_INST_FREQ and PURCH_FREQ. The high correlation between these variables will be reduced later by applying the Principal Component Analysis (PCA). The key findings from the above correlation matrix are:

- The more purchases a customer makes, the more likely they will have had a larger one off purchase.
- Customers with higher credit balances are more likely to have a higher credit limit and also have more cash advances.
- Customers who make more purchases also make more payments.

4. Methods

Clustering is an unsupervised learning method that divides the feature space into clusters or groups of similar objects. In general, clustering is used for pattern recognition. In this paper, we will use two approaches for customer segmentation: the first approach is by considering all variables for clustering algorithms using the Hierarchical clustering and the K-means. The second approach is by using the dimensionality reduction through Principal Component Analysis (PCA), then identifying the optimal number of clusters, and repeating the clustering analysis with the updated number of clusters.

5. First Approach: Considering All Variables for Clustering Algorithms

5.1. Hierarchical Clustering

Hierarchical clustering performs a series of successive mergers to group n objects based on some distance. Unlike K-means, this method does not need clusters to be specified in advance, but rather chooses its clusters by using dendrograms [1,3,5,7,8]. A dendrogram is a tree representation plot that shows how clusters are distributed. The dendrogram consists of clades (branches) that possess at least one or more leaves. The clades are arranged according to how similar (or dissimilar) they are. Clades that are close to the same height are similar to each other; clades with different heights are dissimilar, the greater the difference in height, the more dissimilarity. A hierarchical clustering begins with each data point in its cluster and goes on combining the clusters until a single cluster is reached. A dendrogram criterion is generally used to represent the clusters, which takes the longest edge that does not cross a horizontal line as the minimum distance criterion. Any cluster that crosses this line will be chosen in the final model. The height of the cut to the dendrogram serves as the K value in K-means clustering: it controls the number of clusters obtained. Hierarchical Clustering tries to reduce the variance in the clusters [7,8,9,10,11,12]. Therefore, it provides adequate and sharp clusters. There are two types of hierarchical clustering algorithms:

1. Agglomerative hierarchical clustering (bottom up), which works as follows:

- Start with n clusters where each object is in its own cluster,
- The most similar objects are merged together,
- Repeat step 2 until all objects are in the same cluster.

2. Divisive hierarchical clustering (top down), which works as follows:

- Start with one cluster where all objects are together in the cluster,
- The most dissimilar objects are splitted,
- Repeat step 2 until all objects are in their own cluster.

A distance measure defines similarity (or dissimilarity) between objects. There are different methods for calculating this distance, such as [1,3,13,14,15,16]:

- The Euclidean distance (represents the shortest distance between two points),
- The Manhattan distance (the sum of absolute differences between points across all the dimensions),
- The Minkowski distance, which is a generalization form of the Euclidean distance and the Manhattan distance.
- The Pearson sample correlation distance (represents the converted Pearson correlation coefficient with values between -1 and 1 to a score between 0 and 1).

There are different cluster agglomeration methods (i.e., linkage methods) to calculate the distance between clusters. The most common methods are [1,2,3,15,17,18,19,20,21,22]:

- Maximum or complete linkage clustering: Computes all pairwise dissimilarities between the elements in cluster 1 and the elements in cluster 2 and considers the largest value of these dissimilarities as the distance between the two clusters. It tends to produce more compact clusters.
- Minimum or single linkage clustering: Computes all pairwise dissimilarities between the elements in cluster 1 and the elements in cluster 2 and considers the smallest of these dissimilarities as a linkage criterion. It tends to produce long, “loose” clusters.
- Mean or average linkage clustering: Computes all pairwise dissimilarities between the elements in cluster 1 and the elements in cluster 2 and considers the average of these dissimilarities as the distance between the two clusters. Can vary in the compactness of the clusters it creates.
- Centroid linkage clustering: Computes the dissimilarity between the centroid for cluster 1 (a mean vector of length p, one element for each variable) and the centroid for cluster 2.
- Ward’s minimum variance method: Minimizes the total within-cluster variance. At each step, the pair of clusters with the smallest between-cluster distance are merged. Tends to produce more compact clusters.

The choice of linkage method affects the results. Typically, either complete, average, and Ward’s linkage performs better compared to single linkage which tend to yield trailing clusters [1,2,3].

In addition, we might also consider the standardization (sometimes called data normalization or feature scaling) process of rescaling the values of the variables in our dataset so they share a common scale. Standardization may be important if we are working with data where each variable has a different unit, or where the scales of each of our variables are very different from one another (e.g., 0-1 vs 0-1000). This is particularly important in cluster analysis, because groups are defined based on the distance between points in the space [1,2,3].

In order to implement the Hierarchical clustering in our project, the distance matrix is computed based on the Euclidean distance. We use the Agglomerative hierarchical clustering (bottom up).

We produced dendrograms for the single, complete, and average linkage methods to examine the differences between these methods, and how they affect the results. We also standardized or scaled the variables in the dataset, so they have a common scale. The dendrogram for the single linkage method is shown in Figure 3 below:

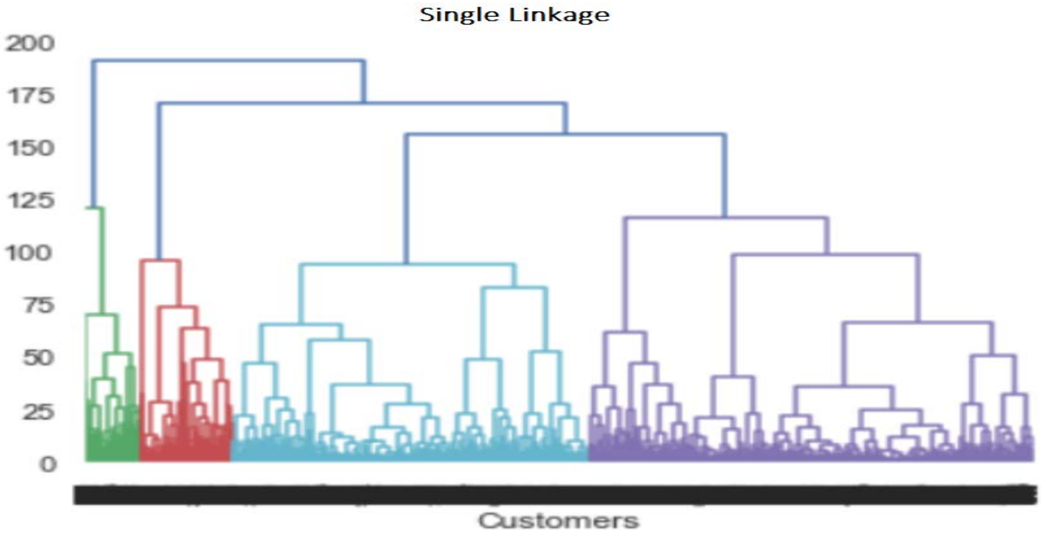


Figure 3. dendrogram of the single linkage method

The dendrogram for the complete linkage method is shown in Figure 4 below:

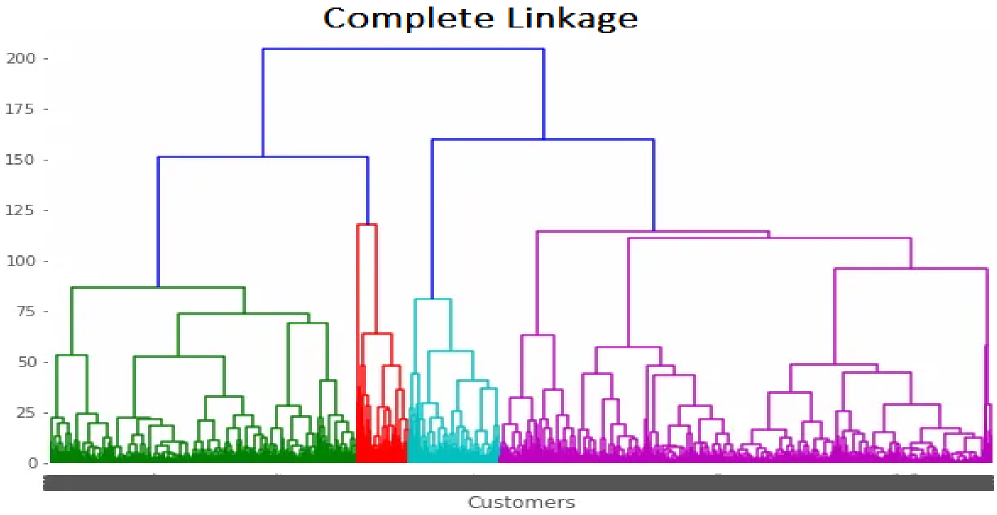


Figure 4. dendrogram of the complete linkage method

The dendrogram for the average linkage method is shown in Figure 5 below:

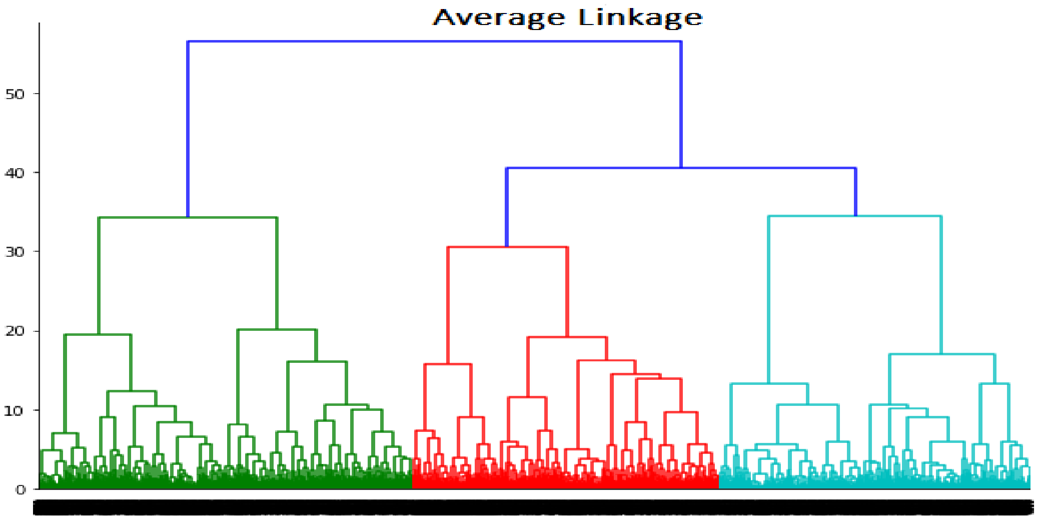


Figure 5. dendrogram of the average linkage method

Next, we cut the dendrogram at the height that will give us an optimal number of clusters, say three, as shown in Figure 6 below:

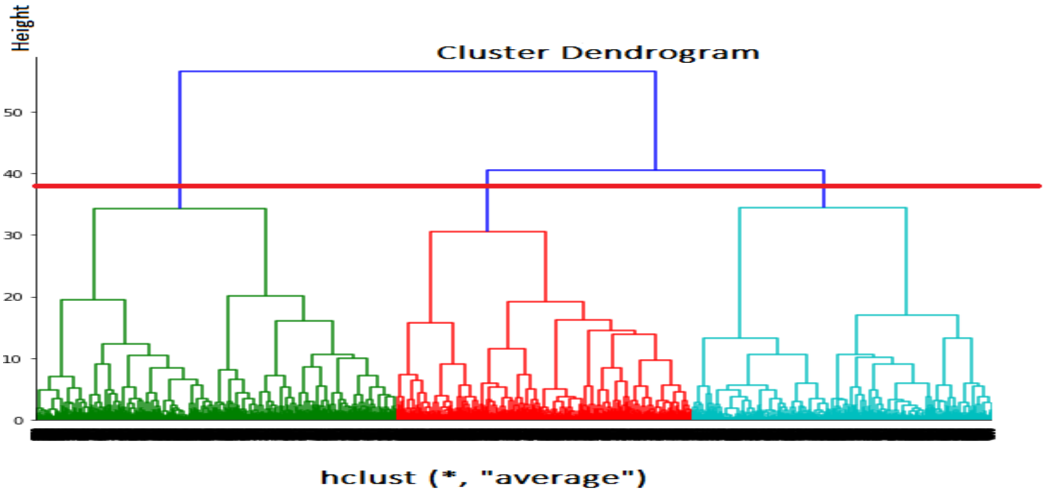


Figure 6. cutting the dendrogram

There are three clusters presented by cutting the dendrogram. By fitting the Hierarchical clustering into our dataset, we can further present these clusters by the plots shown in Figure 7 below using the variables “BALANCE”, “PURCHASES”, “CASH_ADVANCE”, and “CREDIT_LIMIT”:

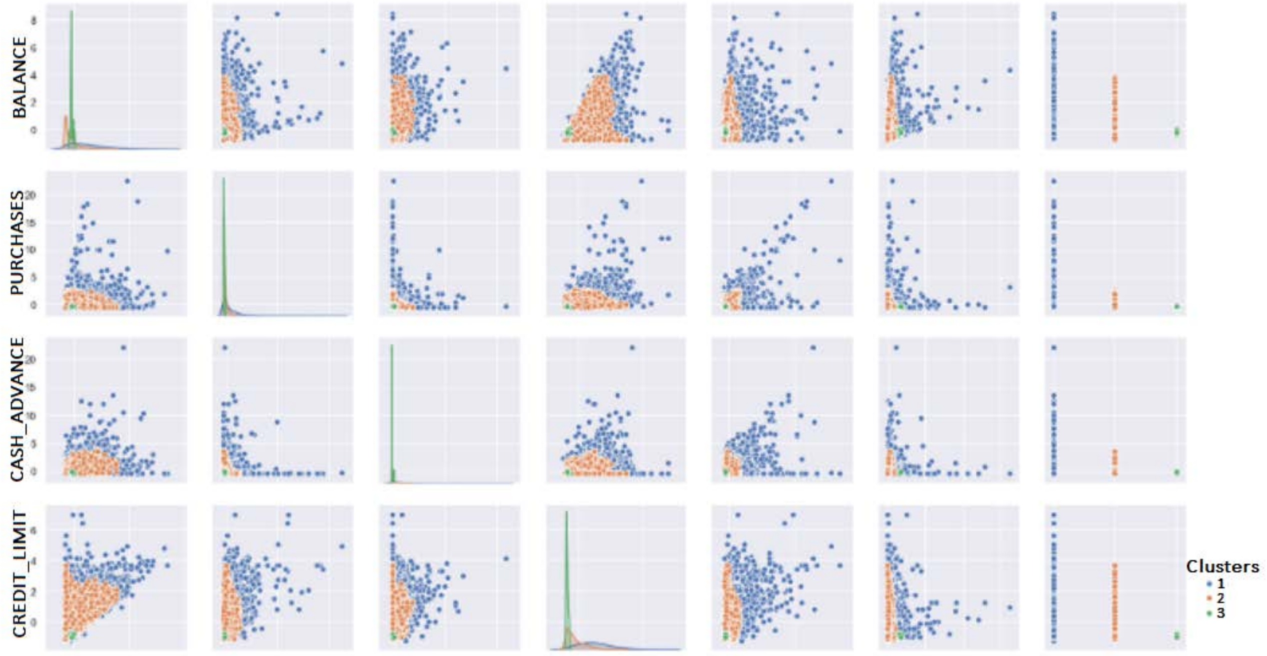


Figure 7. Clusters resulted from fitting the Hierarchical clustering into the dataset

Using the R function (clm_euclidean$clustering_stats), the basic statistics of our analysis regarding the average linkage method using the Euclidean distance for clusters (1, 2, 3) are shown in Table 1 below. There is information about number of observations in each cluster, maximum dissimilarity, average dissimilarity, and isolation.

Table 1. basic statistics of the clusters resulted from the Hierarchical clustering

Clusters		number_obs	max_dissimilarity	average_dissimilarity	isolation
1	2	4591	0.9497022	0.07161773	2.618806
2	3	4359	1.0135705	0.18034851	2.794923

5.2. K-means Clustering

K-means tries to classify observations into mutually exclusive groups (or clusters), such that observations within the same cluster are as similar as possible, whereas observations from different clusters are as dissimilar as possible. In K-means clustering, each cluster is represented by its center (i.e., centroid) which corresponds to the mean of the observation values assigned to the cluster [1,3,6].

5.3. Outline of the K-means Algorithm

Assuming we have input data points x1, x2, x3, …,xn and value of K (the number of clusters needed). We follow the below procedure [1,9,17,22,23,24,25,26]:

1. Pick K points as the initial centroids from the dataset, either randomly or the first K.
2. Find the Euclidean distance of each point in the dataset with the identified K points (cluster centroids).
3. Assign each data point to the closest centroid using the distance found in the previous step.
4. Find the new centroid by taking the average of the points in each cluster group.
5. Repeat 2 to 4 for a fixed number of iteration or till the centroids do not change.

The basic idea behind k-means clustering is constructing clusters so that the total within-cluster variation is minimized. There are several K-means algorithms available for doing this. The standard algorithm is the Hartigan-Wong algorithm, which defines the total within-cluster variation as the sum of the Euclidean distances between observation i's feature values and the corresponding centroid [1,3,23,24,25,27]:

$$W\left(C_k\right)=\sum_{x_i \in C_k}\left(x_i-\mu_k\right)^2$$

where

x_i is an observation belonging to the cluster C_k

μ_k is the mean value of the points assigned to the cluster C_k

Each observation (xi) is assigned to a given cluster such that the sum of squared (SS) distances of each observation to their assigned cluster centers (μk) is minimized. We define the total within-cluster variation as follows:

$$SS_{within}=\sum_{k=1}^{k} W\left(C_k\right)=\sum_{k=1}^{k} \sum_{x_i \in C_k}\left(x_i-\mu_k\right)^2$$

The (SS_{within}) measures the compactness (i.e., goodness) of the resulting clusters and we want it to be as small as possible.

In order to find the optimal number of clusters in K-means, it is recommended to choose it based on:

- The context of the problem at hand if we know that there is a specific number of groups in our data (this option is however subjective), or with any of the following three approaches:
- Elbow method (which uses the within cluster sums of squares by looking at the total within-cluster sum of square as a function of the number of clusters. The location of a knee or elbow in the plot is usually considered as an indicator of the appropriate number of clusters),
- Average silhouette method (measures the quality of a clustering. That is, it determines how well each object lies within its cluster. A high average silhouette width indicates a good clustering),
- Gap statistic method (compares the total intra-cluster variation for different values of K with their expected values under null reference distribution of the data).

In order to implement the K-means clustering in our project, first we choose the optimal number of clusters (K) using the elbow, the average silhouette, and the Gap statistic methods.

The plot for the elbow method is shown Figure 8 below, indicating the optimal number of clusters = 3:

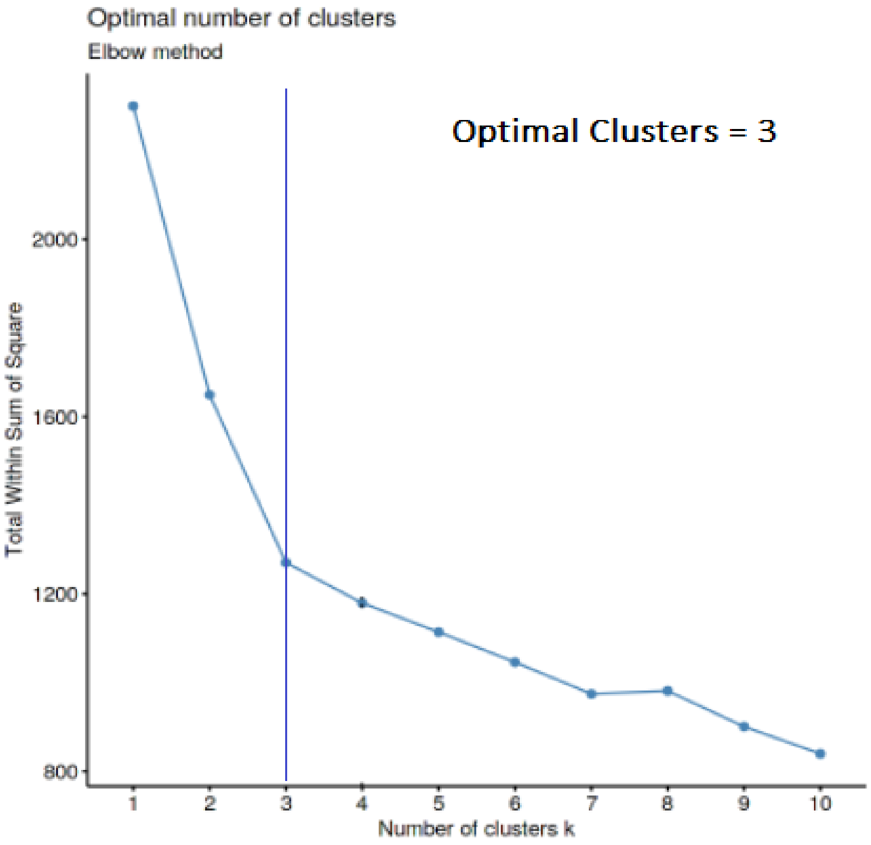


Figure 8. finding optimal number of clusters using the elbow method

The plot for the average silhouette method is shown in Figure 9 below, indicating the optimal number of clusters = 3:

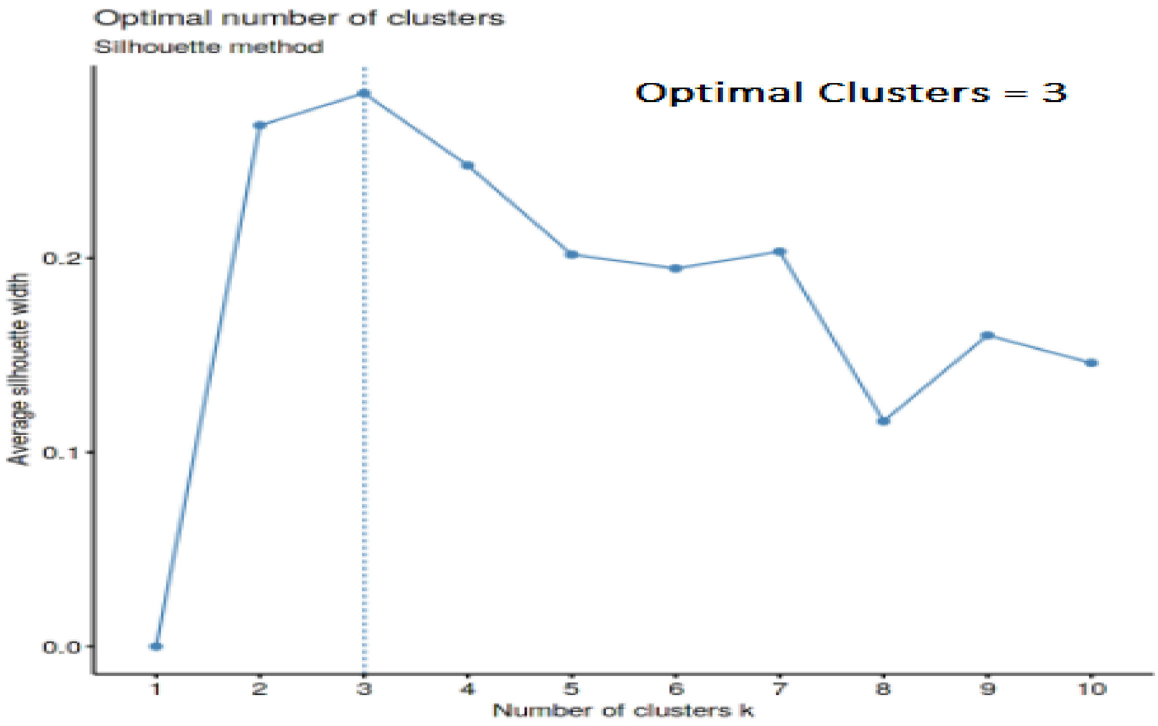


Figure 9. Silhouette method for optimal number of clusters

The plot for the Gap stat. method is shown Figure 10 below, indicating optimal number of clusters = 3:

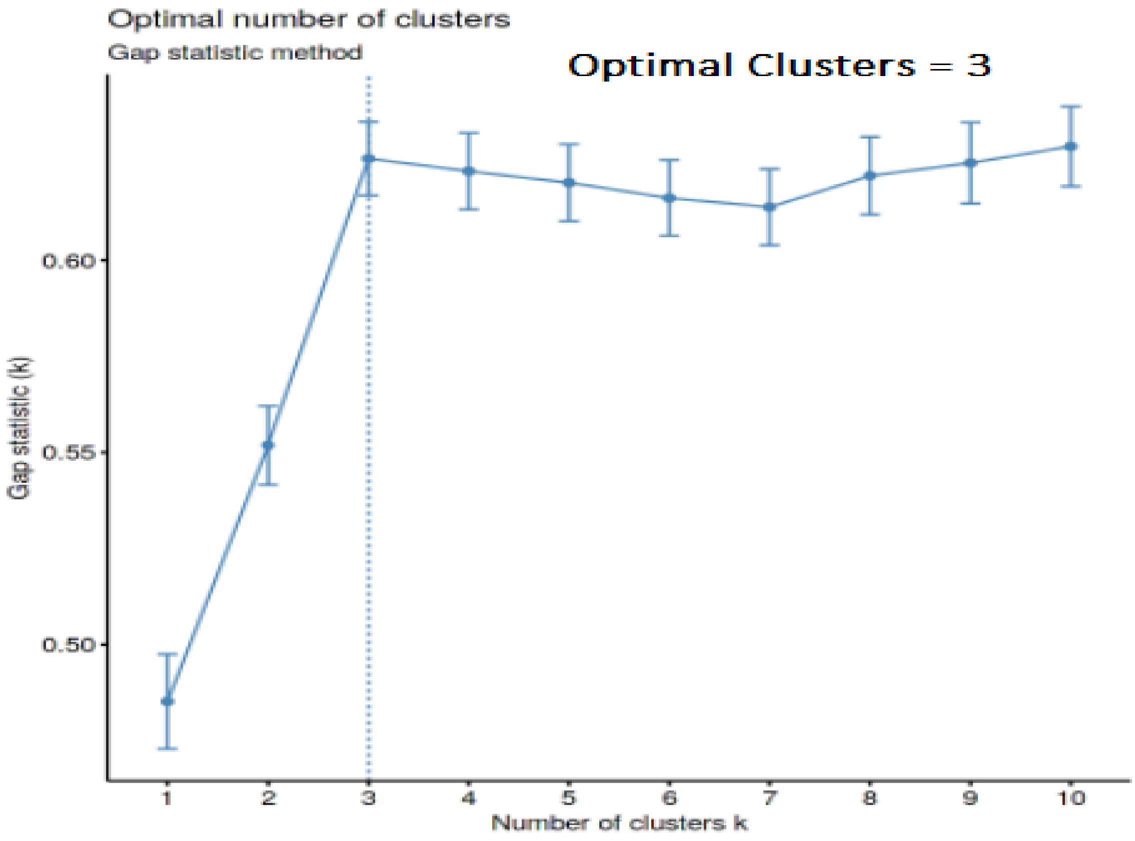


Figure 10. Gap statistic method for optimal number of clusters

Based on the above three methods, we choose the optimal number of clusters K = 3 in our analysis.

By fitting the K-means clustering into our dataset, we can further present these clusters by the plots shown in Figure 11 below using the variables “BALANCE”, “PURCHASES”, “CASH_ADVANCE”, and “CREDIT_LIMIT”:

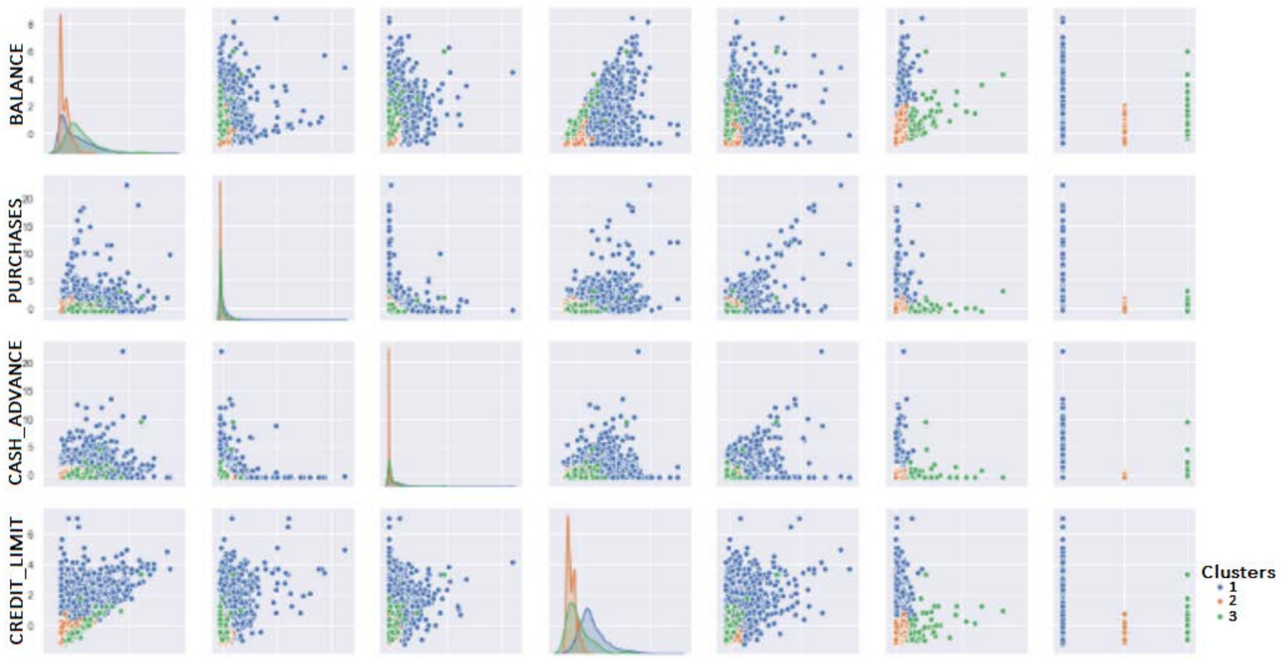


Figure 11. Clusters resulted by fitting K-means into the credit card dataset

The basic statistics of our K-means regarding clusters (1, 2, 3) are shown in Table 2 below. There is information about number of observations in each cluster, maximum dissimilarity, average dissimilarity, and isolation.

Table 2. basic statistics of the clusters resulted from applying the K-means

Clusters		number_obs	max_dissimilarity	average_dissimilarity	isolation
1	2	5597	0.9185662	0.06354467	2.540923
2	3	3353	1.1908471	0.19772944	2.960464

5.4. The Validation of Hierarchical Clustering and K-means Clustering

The evaluation of the clustering outputs is very important for the data modeling. The validation can be based on either external criterion (evaluate the results with respect to a pre-specified structure), or internal criteria (evaluate the results with respect to information related to the data alone). There are different measures for the internal validation of clustering, such as [15,19,20,23,27,29,30]:

- Davies-Bouldin Index: considers the dispersion and separation of all clusters. The score is defined as the average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances. Thus, clusters which are farther apart and less dispersed will result in a better score.
- Silhouette coefficient: it lies between (-1) (e.g. poor clustering) to (+1) (e.g. good clustering). It should be maximized.
- Dunn index: It is the ratio between the smallest distance between observations not in the same cluster to the largest intra-cluster distance. It has a value between 0 and infinity and should be maximized. The higher the Dunn index value, the better is the clustering.

5.5. Silhouette Coefficient

The Silhouette Coefficient is measured as follows:

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where a(i) is the average distance of point i from all other points in its cluster and b(i) is the smallest average distance of i to all points in any other cluster. To clarify, b(i) is found by measuring the average distance of i from every point in cluster A, the average distance of i from every point in cluster B and taking the smallest resulting value.

5.6. Davies-Bouldin (DB) Index

The DB Index is calculated by the following formula:

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

where n is the number of clusters and σ_i is the average distance of all points in cluster i from the cluster centroid c_i.

The DB index captures the intuition that clusters that are: (1) well-spaced from each other and (2) themselves very dense are likely a ‘good’ clustering. This is because the measure’s ‘max’ statement repeatedly selects the values where the average point is farthest away from its centroid, and where the centroids are closest together. As the DB index shrinks, the clustering is considered ‘better’.

5.7. Dunn Index

The formula for the Dunn Index is as follows:

$$D = \frac{\min_{1\le i<j\le n} d(i,j)}{\max_{1\le k\le n} d'(k)}$$

where i, j and k are indices for clusters, d measures the inter-cluster distance and d’ measures the intra-cluster difference.

The Dunn Index captures the same idea as the DB Index: it gets better when clusters are well-spaced and dense. But the Dunn Index increases as performance improves.

In order to compare the results obtained from both the Hierarchical clustering and K-means, we applied these measures to the outputs of the methods, and get the results shown in Table 3 below:

Table 3. different validation metrics for the Hierarchical and K-means clustering

Algorithm	Davis-Bouldin Index	Silhouette Coefficient	Dunn Index
Hierarchal Clustering	1.834418	0.289862	0.57328051
K-means Clustering	2.017857	0.311893	0.64092362

We can see from the table that the K-means clustering has better scores than the Hierarchical clustering in terms of Davis-Bouldin, Silhouette, and Dunn index. Therefore, K-means clustering algorithm is more suitable for customer segmentation than Hierarchical clustering in this dataset.

6. Second Approach: Dimensionality Reduction Using PCA

The Principal Component Analysis (PCA) is an unsupervised statistical technique that can be used for dimension reduction, feature extraction, and data visualization. PCA can analyze the data to identify patterns in order to reduce the dimensions of the dataset with minimal loss of information. High dimensionality means that the dataset has a large number of features, which could produce overfitting. By reducing the dimensions of datasets, PCA provides an effective and efficient method for data description and visualization. PCA produces a low-dimensional representation of the feature space by finding a sequence of linear combinations of the features that have maximal variance and are mutually uncorrelated. The proportion of variance explained by each principal component is a measure of the strength of each component. In this project, we use PCA as an approach to group features that explain the maximum variability in the feature space. Dimensionality reduction aims to reduce the number of variables in our dataset and also reduce multicollinearity among variables. In general, a $n \times p$ data matrix (X) has min $(n - 1, p)$ distinct principal components. However, we usually like to use just the first few principal components in order to visualize or interpret the data. We decide on the number of principal components required to visualize the data by examining a scree plot to find a point at which the proportion of variance explained by each subsequent principal component drops off. This is often referred to as an elbow in the scree plot. Each principal component is a linear combination of the initial variables. Also, each principal component has an orthogonal relationship with each other. This means they are uncorrelated. The first principal component (PC1) captures most variability within the data. The second principal component (PC2) captures the second most. The third principal components (PC3) captures the third most, and so on. PCA works by the following steps [1,2,3,25,26,27,29,30]:

1. Normalize the Data: This is often necessary if the features in the dataset are measured in different units,
2. Calculate the covariance matrix,
3. Compute the eigen values and eigen vectors,
4. Re-orient the data,
5. Plot the data by biplots (PC1 against PC2).

6.1. Normalization of the Data

Before PCA is performed, the variables should be centered to have mean zero. We typically scale each variable to have standard deviation one before we perform PCA. The high variabilities in the features could result in an arbitrarily large variance. Normalization addresses this issue. Therefore, we have to check the standard deviation and variance of the features as we normalize them before running the PCA.

6.2. Examining the Importance of the Principal Components

Next, we apply the PCA to our dataset after normalization. The initial number of the principal components would be the same as the number of variables in the dataset (17 in our dataset). The summary table of the initial PCs is shown in Table 4 below.

Table 4. The importance of initial principal components Importance of components

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
Standard deviation:	2.154	1.8586	1.22482	1.1276	1.02707	0.98720	0.91101	0.85733	0.80175
Proportion of Variance:	0.273	0.2032	0.08825	0.0748	0.06205	0.05733	0.04882	0.04324	0.03781
Cumulative Proportion:	0.273	0.4762	0.56444	0.6392	0.70129	0.75862	0.80744	0.85068	0.88849
	PC10	PC11	PC12	PC13	PC14	PC15	PC16	PC17	
Standard deviation:	0.7236	0.63509	0.54907	0.49285	0.45478	0.41489	0.21307	0.003413	
Proportion of Variance:	0.0308	0.02373	0.01773	0.01429	0.01217	0.01013	0.00267	0.000000	
Cumulative Proportion:	0.9193	0.94301	0.96075	0.97504	0.98720	0.99733	1.00000	1.000000	

The result of the summary table shows three statistics for all components: standard deviation, proportion of variance and cumulative variance. From the output we can see that PC1 explains about 27% of variance, PC2 explains about 20% of variance, PC3 explains about 8% of variance, PC4 explains about 7% of variance, PC5 explains about 6% of variance and so on.

6.3. Reducing the Correlation between Highly Correlated Variables

One advantage of PCA is that it would reduce the high correlation between correlated variables. The original dataset contains high correlations between some variables. For example, a correlation of (0.92) exists between "PURCHASE and ONE-OFF PURCHASE", and a correlation of (0.86) exists between "PURCHASE FREQUENCY and PURCHASE_ INSTS_FREQUENCY". After applying PCA, the correlation between all variables becomes less than (0.61), as shown in Figure 12.

6.4. Choosing the Optimal Number of Principal Components to Retain

In order to select the optimal number of Principal Components (PCs) to retain, we can use either Kaiser's criterion or the scree plot [1,3,25,29,30]. The Kaiser's criterion implies that one should retain the principal components that have eigenvalues greater than one. We need to compute the eigenvalues of the PCs by using get_eigenvalue() from factoextra package in R, and they are shown in Table 5 below.

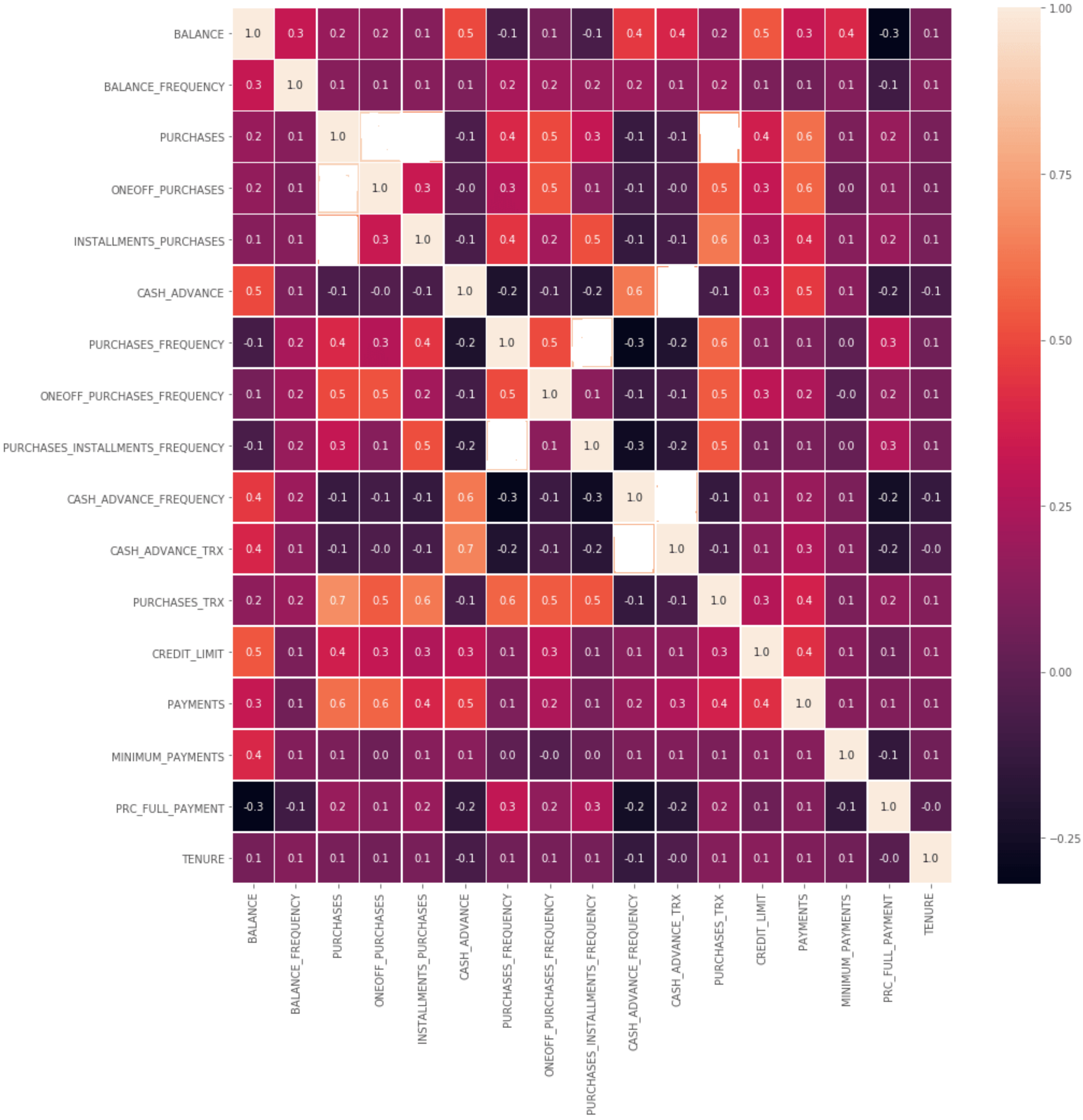


Figure 12. correlation matrix of the variables after applying the PCA

Table 5. the eigen values and variance% of the initial PCs (dimensions)

	eigenvalue	Variance percent	Cumulative Variance percent
Dim.1	4.640880e+00	2.729930e+01	27.29930
Dim.2	3.454466e+00	2.032039e+01	47.61968
Dim.3	1.500177e+00	8.824573e+00	56.44426
Dim.4	1.271589e+00	7.479937e+00	63.92419
Dim.5	1.054863e+00	6.205076e+00	70.12927
Dim.6	9.745631e-01	5.732724e+00	75.86199
Dim.7	8.299380e-01	4.881988e+00	80.74398
Dim.8	7.350087e-01	4.323581e+00	85.06756
Dim.9	6.428031e-01	3.781195e+00	88.84876
Dim.10	5.236154e-01	3.080091e+00	91.92885
Dim.11	4.033368e-01	2.372569e+00	94.30142
Dim.12	3.014818e-01	1.773422e+00	96.07484
Dim.13	2.429048e-01	1.428852e+00	97.50369
Dim.14	2.068286e-01	1.216639e+00	98.72033
Dim.15	1.721326e-01	1.012545e+00	99.73287
Dim.16	4.539988e-02	2.670581e-01	99.99993
Dim.17	1.164913e-05	6.852429e-05	100.00000

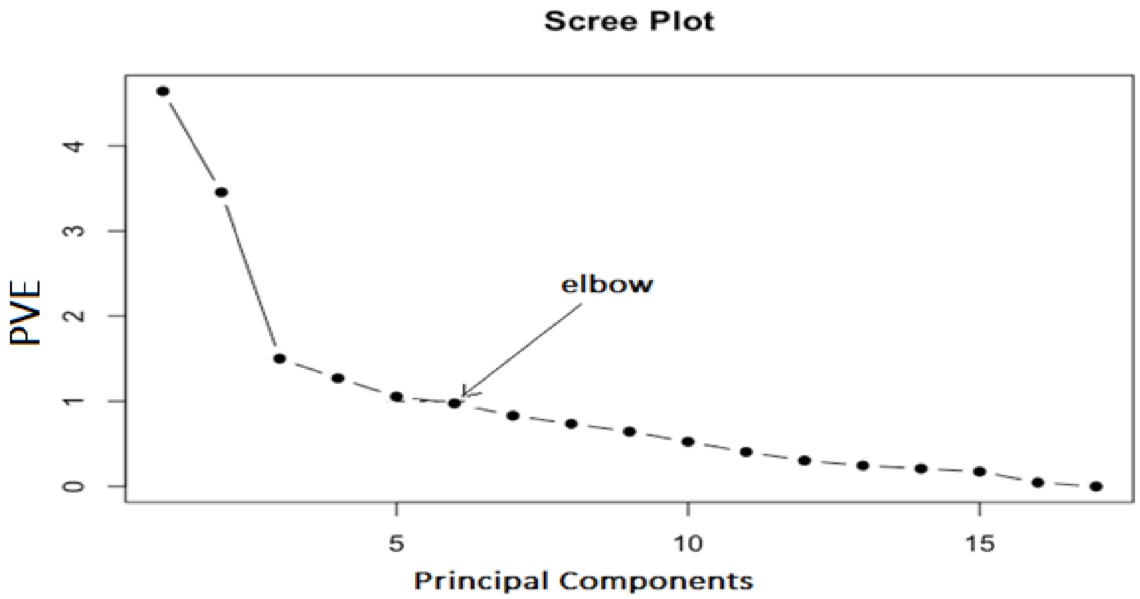


Figure 13. the scree plot for finding the optimal number of PCs

Since the first eigenvalue (4.64) is higher than 1, we have to retain the first component. The second eigenvalue (3.45) is also higher than 1, so it should also be retained. Eigenvalue higher than 1 is also for third (1.5), fourth (1.27) and fifth component (1.05). According to Kaiser’s criterion, the remaining components should not be used in further analysis.

Next, we can use the scree plot to select the optimal number of PCs, as shown in Figure 13.

The scree plot above can be used to determine the optimal number of PCs to represent the data. On the plot, we look for an elbow. We can see that there is an elbow in the plot after approximately the fifth principal component. Thus, 5 PCs seem to be the optimal number of PCs.

Since both Kaiser’s criterion and the scree plot indicate that there should be five principal components retained for further analysis, we use optimal number of PCs = 5.

From the plot of the cumulative percent of variance explained of the 5 optimal components shown in Figure 14 below, we can see that these 5 PCs explain about 87% variance in the data.

So, initially we had 17 variables in our dataset, now it is only 5. Thus, our variables get reduced by applying the PCA. The data frame for these 5 PCs is shown in Table 6 below.

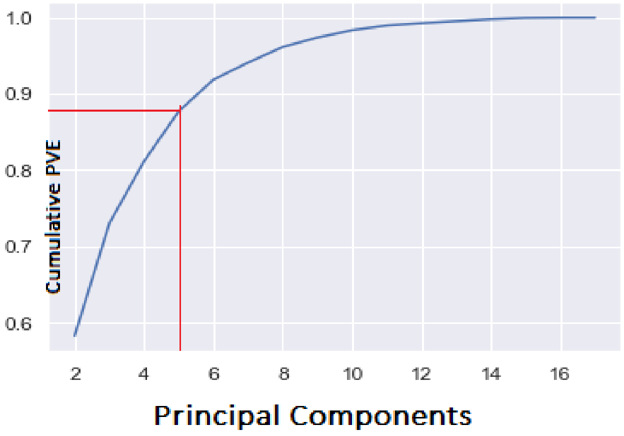
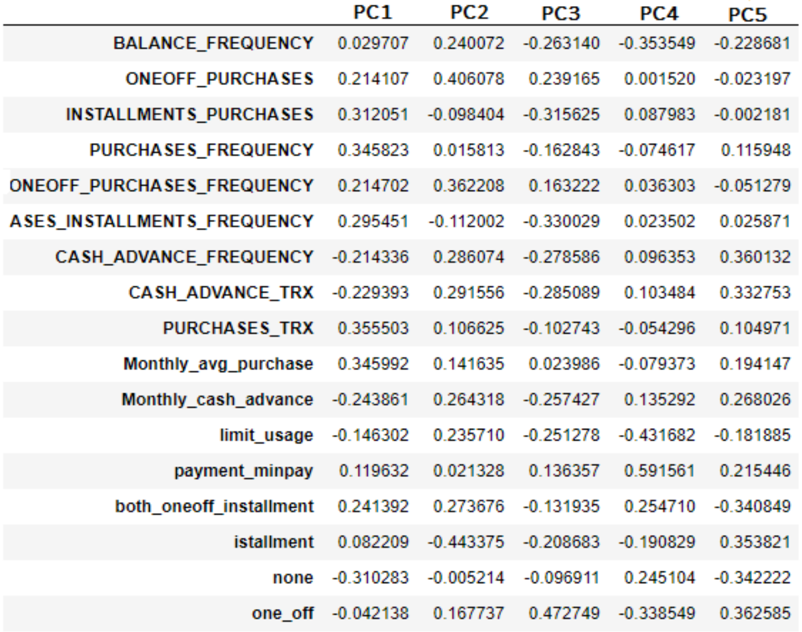


Figure 14. the cumulative percent of variance explained by the PCs

Table 6. data frame of the 5 optimal PCs



	PC1	PC2	PC3	PC4	PC5
BALANCE_FREQUENCY	0.029707	0.240072	-0.263140	-0.353549	-0.228681
ONEOFF_PURCHASES	0.214107	0.406078	0.239165	0.001520	-0.023197
INSTALLMENTS_PURCHASES	0.312051	-0.098404	-0.315625	0.087983	-0.002181
PURCHASES_FREQUENCY	0.345823	0.015813	-0.162843	-0.074617	0.115948
ONEOFF_PURCHASES_FREQUENCY	0.214702	0.362208	0.163222	0.036303	-0.051279
ASES_INSTALLMENTS_FREQUENCY	0.295451	-0.112002	-0.330029	0.023502	0.025871
CASH_ADVANCE_FREQUENCY	-0.214336	0.286074	-0.278586	0.096353	0.360132
CASH_ADVANCE_TRX	-0.229393	0.291556	-0.285089	0.103484	0.332753
PURCHASES_TRX	0.355503	0.106625	-0.102743	-0.054296	0.104971
Monthly_avg_purchase	0.345992	0.141635	0.023986	-0.079373	0.194147
Monthly_cash_advance	-0.243861	0.264318	-0.257427	0.135292	0.268026
limit_usage	-0.146302	0.235710	-0.251278	-0.431682	-0.181885
payment_minpay	0.119632	0.021328	0.136357	0.591561	0.215446
both_oneoff_installment	0.241392	0.273676	-0.131935	0.254710	-0.340849
istallment	0.082209	-0.443375	-0.208683	-0.190829	0.353821
none	-0.310283	-0.005214	-0.096911	0.245104	-0.342222
one_off	-0.042138	0.167737	0.472749	-0.338549	0.362585

The greater the variance explained by the PC, the more information that is summarized by that PC. The variance explained by each of the five principal components is shown in Table 7 below:

Table 7. the variance explained by the optimal PCs

PC	Variance Explained:
PC1	0.402058
PC2	0.180586
PC3	0.147294
PC4	0.081606
PC5	0.075511

We can see that PC1 explains about 40% of the variance in the data, PC2 explains about 18% of the variance in the data, PC3 explains about 14% of the variance in the data, PC4 explains about 8% of the variance in the data, and PC5 explains about 7% of the variance in the data.

6.5. Graphical Analysis using PCA

PCA is a powerful tool for graphical analysis and visualization of the data. It can show important insights of our data. To better understand the customer segmentation, we will use the visualization of variable's loadings as described below [5,8,28].

6.6. Visualization of Variable's Loadings using Biplots

We can construct what is known as biplots that display both the principal component scores and the principal component loadings. Loading describes the relationship between the original variables and the new principal component. Specifically, it describes the weight given to an original variable when calculating a new principal component. Score describes the relationship between the original data and the newly generated axis, so the score is the new value for a data row in the principal component space.

We generated the biplots of (PC1 against PC2), (PC2 against PC3), (PC3 against PC4), and (PC4 against PC5) as shown below. We can see from the biplots below that the correlation between original variables as well as the strength of each variable contribute to particular principal components. It is clear from the first plot (PC1, PC2) that e.g. cash-advance frequency and average amount per purchase transaction are positively correlated while cash-advance frequency and percent of months with full payment of the due statement balance seems to be negatively correlated. If variables are grouped together, it means they are positively correlated while if variables are positioned on opposite sides of the plot, it means that they are negatively correlated. The length of the vector tells how strong the contribution of particular variable to particular principal component is.

The biplot of PC1 and PC2 is shown in Figure 15 below:

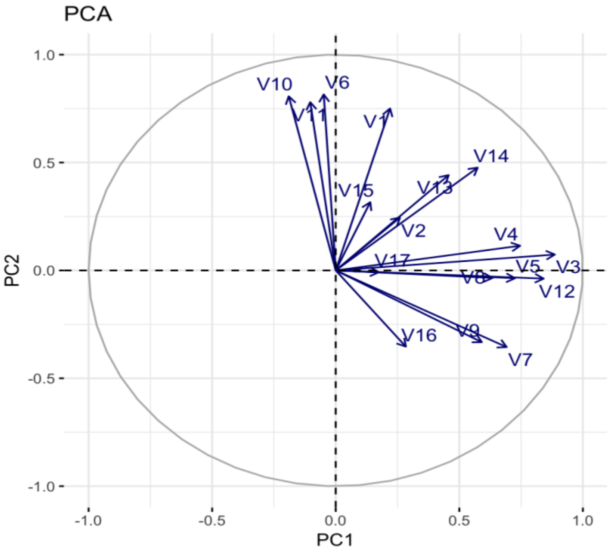


Figure 15. PC2 against PC2

The biplot of PC2 and PC3 is shown in Figure 16 below:

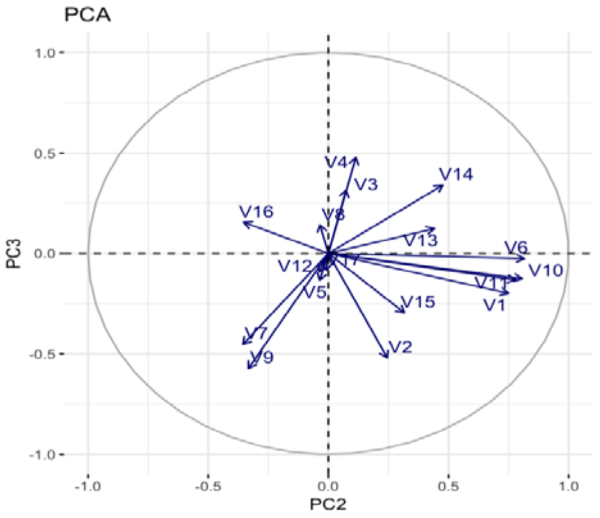


Figure 16. PC2 against PC3

The biplot of PC3 and PC4 is shown in Figure 17 below:

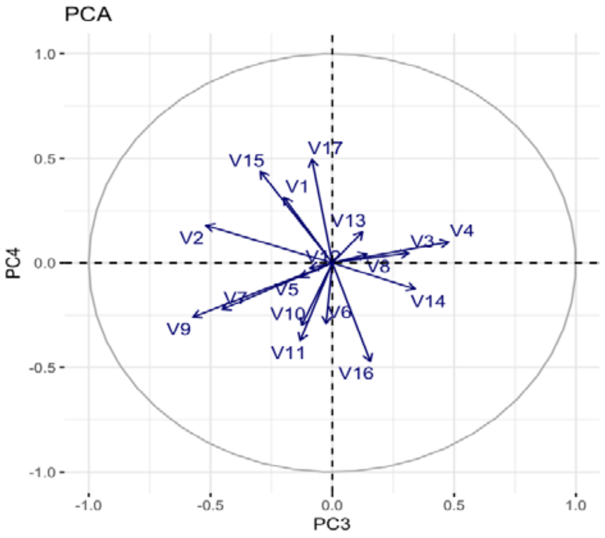


Figure 17. PC3 against PC4

The biplot of PC4 and PC5 is shown in Figure 18 below:

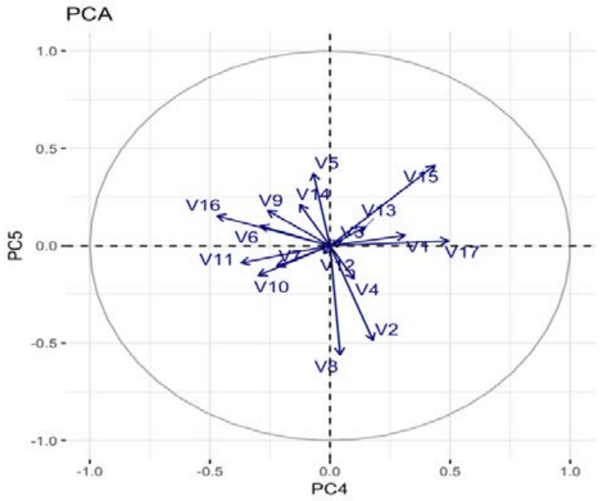


Figure 18. PC4 against PC5

Since the length of the vector tells how strong the contribution of particular variable to particular principal component is, this can also be confirmed by the plots in Figure 19 below. It presents the contribution of each variable to PC1, PC2, PC3, PC4 and PC5.

From the biplots and contribution plots, we can identify the components of each of the 5 PCs as follows:

Components of PC1:

- Total purchase amount spent during last 12 months
- Average amount per purchase transaction
- Total amount of one-off purchases
- Total amount of installment purchases
- Frequency of purchases (percentage of months with at least one purchase
- Frequency of one-off-purchases
- Frequency of installment purchases
- Total payments (due amount paid by the customer to decrease their statement balance).

Components of PC2:

- Total cash-advance amount
- Cash-Advance frequency
- Average amount per purchase transaction
- Monthly average balance (based on daily balance averages)
- Credit limit
- Total payments (due amount paid by the customer to decrease their statement balance).

Components of PC3:

- Frequency of installment purchases
- Ratio of last 12 months with balance
- Total amount of one-off purchases
- Frequency of purchases (percentage of months with at least one purchase)
- Total payments (due amount paid by the customer to decrease their statement balance) in the period
- Total purchase amount spent during last 12 months
- Total minimum payments due in the period.

Components of PC4:

- Number of months as a customer
- Percentage of months with full payment of the due statement balance
- Total minimum payments due in the period
- Average amount per cash-advance transaction
- Monthly average balance (based on daily balance averages)
- Total cash-advance amount
- Cash-Advance frequency.

Components of PC5:

- Frequency of one-off-purchases
- Ratio of last 12 months with balance
- Total minimum payments due in the period
- Total amount of installment purchases.

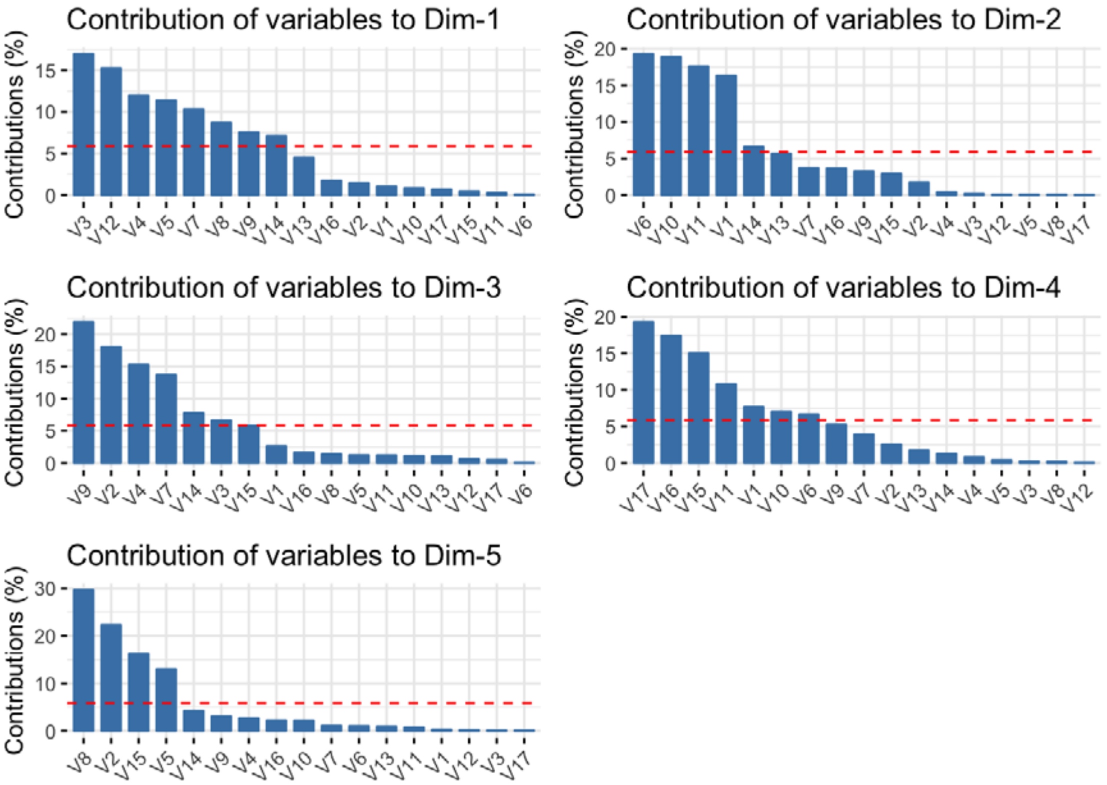


Figure 19. contributions of variables to the 5 optimal PCs

6.7. Employing the PCA in the Clustering Analysis

Although PCA is mainly used for visualization and dimensionality reduction, however, it can also be employed in the clustering analysis, customer segmentation, and pattern recognition. Unlike K-means, PCA is not a direct solution for clustering, but it can be used to improve the results of the K-means clustering and the Hierarchical clustering by detecting additional clusters as compared to the optimal number of clusters in the K-means or Hierarchical clustering. Hence, we use the PCA in this paper to detect more clusters or groups and compare the results to the K-means and Hierarchical clustering outputs. The PCA clustering procedure is based on the 5 PCs instead of 17 variables. Applying PCA to our data frame resulted in producing 4 clusters, as shown in Figure 20 below.

Since the procedure of PCA clustering produced 4 clusters, this implies that the optimal K=3 that was used in the Hierarchical and K-means has to be changed to K=4. The PCA has identified one more cluster or group of customers that was not discovered by the K-means and Hierarchical clustering. Therefore, we will update the optimal K value, and repeat the K-means clustering again to see if the results would improve with the new optimal K value.

6.8. Updating the K-means Clustering with K=4 instead of K=3

Applying the K-means algorithm with K=4 to our data, we get the K-means clustering results shown in Figure 21 below.

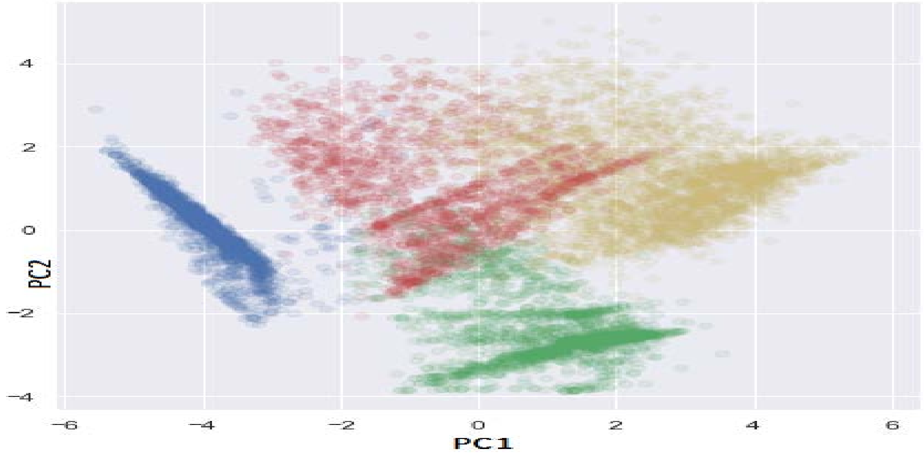


Figure 20. clusters resulted from applying the PCA

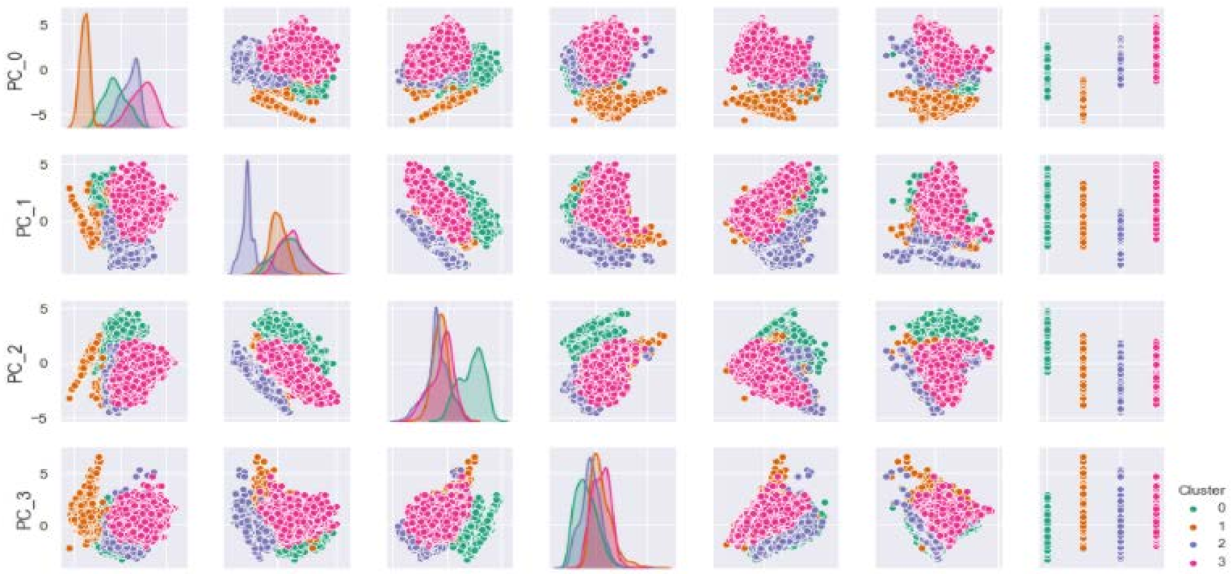


Figure 21. clusters resulted from applying the PCA to the dataset

The new updated K-means clustering resulted in producing four principal components (or clusters or groups). The basic statistics of our new K-means analysis regarding clusters (1, 2, 3, 4) are shown in Table 8 below. There is information about number of observations in each cluster, maximum dissimilarity, average dissimilarity, and isolation.

Table 8. basic statistics for the clusters resulted from the PCA

Clusters		number_obs	max_dissimilarity	average_dissimilarity	isolation
1	2	3507	1.9498201	0.070997583	2.3397045
2	3	2826	0.9168076	0.187769458	2.7205668
3	4	2617	0.8850341	0.144098962	2.6720819

In order to evaluate the results of the new K-means clustering, we applied the same evaluation measures that were applied previously to K-means and Hierarchical clustering, and we get the results shown in Table 9 below:

Table 9. validation metrics for the updated K-means clustering

Algorithm	Davis-Bouldin Index	Silhouette Coefficient	Dunn Index
New K-means Clustering	2.209894	0.340901	0.71098841

We can see from the table that the new K-means clustering has better scores than the original K-means and Hierarchical clustering in terms of Davis-Bouldin, Silhouette, and Dunn index. This implies that updating the optimal K value from K=3 to K=4 has improved the outcome of K-means clustering.

The average Silhouette score of the four clusters is shown in Figure 22 below:

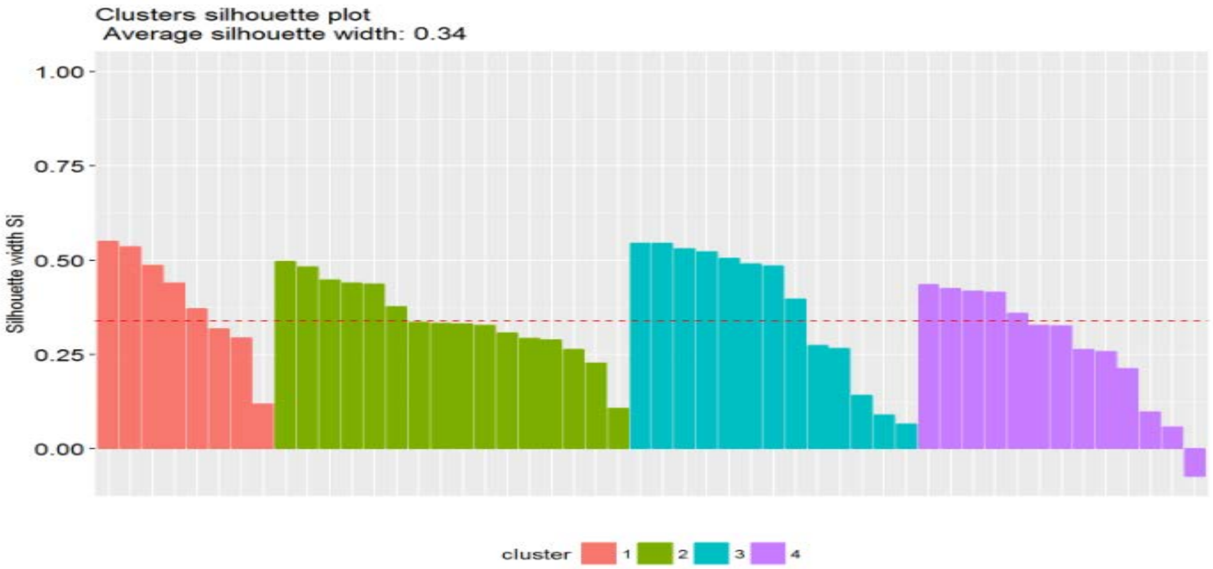


Figure 22. AVERAGE Silhouette width score for the updated K-means clustering

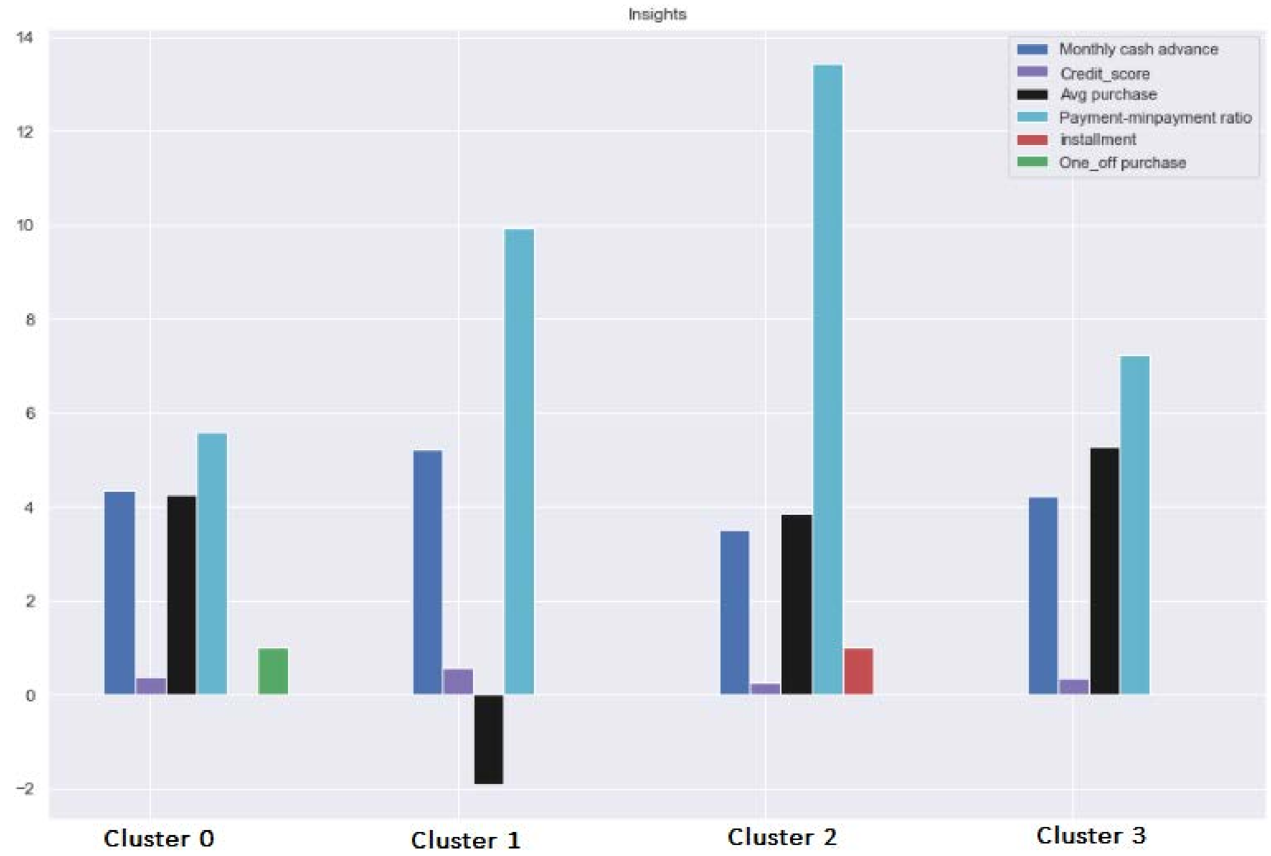


Figure 23. histograms of the four clusters or groups

6.9. Characteristics of the 4 Clusters or Customer's Groups

In order to show the characteristics of each cluster or customer's group with respect to monthly cash advance, credit scores, average purchase, payment ratio, installments, and one-off purchases, we present the histograms of the 4 clusters as shown in Figure 23.

By inspecting the histograms, we can interpret the characteristics of the customers in each cluster as follows:

1. The customers in the first cluster (cluster 0) do only one-off transaction and have least payment ratio. This group is about 21% of the total customers in the data.
2. The customers in the second cluster (cluster 1) take the maximum advance cash and pay less minimum payment and have poor credit scores. This group is about 23% of the total customers in the data.
3. The customers in the third cluster (cluster 2) have the highest monthly average purchases and they do both installments as well as one-off purchase. Besides, they generally have good credit scores. This group is about 31% of the total customers in the data.
4. The customers in the fourth cluster (cluster 3) have maximum credit score and they pay dues and maximum installment purchases. This group is about 25% of the total customers in the data.

Hence, K-means with 4 clusters is an effective tool that can show distinguished characteristics of each cluster or customer's group.

7. Conclusions

In this paper, we investigated the applications of clustering analysis in the market segmentation and customer segmentation. We identified active customers in order to apply proper marketing strategy towards them. We segmented the customers into four groups: the active users, the frequent users, the mid users, and the rare users.

As we saw from the results of the K-means and Hierarchical clustering, the K-means better fitted with our dataset. The K-means clustering scored better than the Hierarchical clustering in terms of Davis-Bouldin, Silhouette, and Dunn index. Therefore, we conclude that the K-means clustering algorithm is more suitable for customer segmentation than Hierarchical clustering in this dataset.

In addition, we showed that the Principal Component Analysis (PCA) as an unsupervised statistical technique can be used for dimension reduction, and data visualization. We proved that PCA can analyze the data to identify patterns in order to reduce the dimensions of the dataset with minimal loss of information. We found that high dimensionality could produce overfitting, and PCA was effective to reduce the number of variables (dimensions) in our dataset from 17 to only 5. Also, we proved that PCA can effectively reduce the multicollinearity between high correlated variables in our dataset.

Moreover, we showed that PCA can be employed in the clustering analysis, customer segmentation, and group detection. We showed that, unlike K-means, PCA is not a direct solution for clustering, but it can improve the results of the K-means clustering and Hierarchical clustering by detecting more clusters or patterns in the data as compared to the K-means clusters. Hence, we used the PCA in this paper to detect additional clusters or groups and compared the results to the K-means and Hierarchical clustering outputs.

Interestingly, we saw that the procedure of PCA clustering produced 4 clusters compared to the optimal K=3 that was used in the Hierarchical clustering and K-means. The PCA identified one more cluster or group of customers that was not discovered by the K-means and Hierarchical clustering. Therefore, we updated the optimal K value, and repeated the K-means clustering procedure again to see if the results would improve with the new optimal K value. We showed that the new updated K-means clustering scored better than the original K-means and Hierarchical clustering in terms of Davis-Bouldin, Silhouette, and Dunn index. This implied that updating the optimal K value from K=3 to K=4 improved the outcome of K-means clustering.

Based on the results of clustering algorithms achieved in this paper, the Principal Component Analysis (PCA) cannot be understated in the applications of clustering analysis.

Interesting results, such as the updated optimal K value by PCA , demonstrate the importance of having good knowledge of the reasons behind why the use of PCA enhances the clustering process.

Properly preprocessing and scaling the data was very important step. Perfect inspection, and knowledge of the data is noted to be very important in terms of modeling it and using it for clustering. Normalization of data should be considered in clustering analysis if the features in the dataset are measured in different units.

It was also discovered that, in addition to exploring all available data insights, multiple different clustering algorithms should be attempted. Different properties of the data are likely best exploited through fitting with different clustering types. In the instance of the data used for this project, the K-means clustering seemed to indicate the best fit for this data.

However, more complex models could also be investigated for future works. Possible advanced clustering methods might include the Fuzzy C-means clustering, Density-Based clustering, and Distribution-Model Based clustering. Further exploiting the combination of PCA with K-means and Hierarchical clustering might be applicable with this further work.

7. Recommendations for Marketing Strategies

By carefully investigating the data, we can identify some inputs to recommend marketing strategies to the credit card company as follows:

- Customers with high balance, and high purchase: These people made expensive purchases, but they also had higher balances to support these purchases. They also made large payments and can be the target for market research.
- Customers with high Balance, and low purchase: These are the people who had higher balances but made lower purchases and have moderate credit limits and took out large cash advances.
- Customers with medium balance, and medium purchase: These people did not have low or high balances and they also did not make big or small purchases, but they did everything at a medium level.
- Customers with low balance, and low purchase: These people made the smallest purchases and since their credit limit was also low, this means that the customers did not make these purchases frequently.

Based on the results of our clustering analysis, we can recommend the following marketing strategies to the credit card company regarding their customer's segmentation:

- **Customers in Group 1**

Customers in this group have minimum paying ratio and use credit card for just one-off transaction (may be for utility bills only). Therefore, this group seems to be ineffective in targeting for any potential marketing strategy.

- **Customers in Group 2**

Customers in this group have poor credit score and take only cash in advance. The company can target them by providing less interest rate on purchase transactions.

- **Customers in Group 3**

Customers in this group are potential target customers who pay dues and make purchases and maintain comparatively good credit scores. The company can offer them an increase in the credit limit or can lower down their interest rates. They can be given premium cards or loyalty cards to increase their potential transactions.

- **Customers in Group 4**

Customers in this group are performing best among all groups, as they maintain good credit scores and pay dues on time. The company can offer them higher reward points, which will encourage them to make more purchases.

References

[1] James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning: With Applications in R*. Springer.

[2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2008. *The Elements of Statistical Learning*. Springer.

[3] Brett Lantz. 2019. *Machine Learning with R*. Packt Publishing Ltd.

[4] Alboukadel Kassambara. 2017. *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*. Sthda.com.

[5] Alboukadel Kassambara. 2017. *Practical Guide to Principal Component Methods in R*. Sthda.com.

[6] Alboukadel Kassambara. 2017. *R Graphics Essentials for Great Data Visualization*. Sthda.com.

[7] Aurélien Géron. 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly.

[8] Philip D. Waggoner. 2020. *Unsupervised Machine Learning for Clustering in Political and Social Research*. Cambridge University Press.

[9] Ankur A. Patel. 2019. *Hands-On Unsupervised Learning Using Python: How to Build Applied Machine Learning Solutions from Unlabeled Data*. O'Reilly.

[10] Nayna Maheshwari. 2020. *Artificial Intelligence: Applications, Problem Solving, Machine Learning, Knowledge Representation and Reasoning*.

[11] Bradford Tuckfield. 2019. *Applied Unsupervised Learning with R: Uncover hidden relationships and patterns with k-means clustering, hierarchical clustering, and PCA*. Packt Publishing Ltd.

[12] Tarek Amr. 2020. *Hands-On Machine Learning with scikit-learn and Scientific Python Toolkits: A practical guide to implementing supervised and unsupervised machine learning algorithms in Python*. Packt Publishing Ltd.

[13] Morgan Maynard. 2020. *Machine Learning: Introduction to Supervised and Unsupervised Learning Algorithms with Real-World Applications*.

[14] LazyProgrammer. 2016. *Unsupervised Machine Learning in Python: Master Data Science and Machine Learning with Cluster Analysis, Gaussian Mixture Models, and Principal Components Analysis*.

[15] Rowel Atienza. 2020. *Advanced Deep Learning with TensorFlow 2 and Keras: Apply DL, GANs, VAEs, deep RL, unsupervised learning, object detection and segmentation.* 2nd edition, Packt Publishing Ltd.

[16] Fred Nwanganga and Mike Chapple. 2020. *Practical Machine Learning in R.* Wiley.

[17] Stephen Marsland. 2011. *Machine Learning: An Algorithmic Perspective*. Chapman and Hall/CRC.

[18] Abdulhafedh, A. (2016). *Crash Frequency Analysis*. Journal of Transportation Technologies, 6, 169-180.

[19] Steven L. Brunton and J. Nathan Kutz. 2019. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press.

[20] Pratap Dangeti. 2017. *Statistics for Machine Learning: Techniques for exploring supervised, unsupervised, and reinforcement learning models with Python and R.* Packt Publishing Ltd.

[21] Abdulhafedh, Azad. (2017). *Road Crash Prediction Models: Different Statistical Modeling Approaches.* Journal of Transportation Technologies, 7, 190-205.

[22] Marius Leordeanu. 2020. *Unsupervised Learning in Space and Time: A Modern Approach for Computer Vision using Graph-based Techniques and Deep Neural Networks*. Springer.

[23] Michael Colins. 2017. *Machine Learning: An Introduction to Supervised and Unsupervised Learning Algorithms.*

[24] Chirag Shah. 2020. *A Hands-On Introduction to Data Science*. Cambridge University Press.

[25] Sunil Kumar Chinnamgari. 2019. *R Machine Learning Projects: Implement supervised, unsupervised, and reinforcement learning techniques using R 3.5*. Packt Publishing Ltd.

[26] Abdulhafedh, Azad. (2017*). Incorporating the Multinomial Logistic Regression in Vehicle Crash Severity Modeling: A Detailed Overview*. Journal of Transportation Technologies, 7, 279-303.

[27] Kevin Jolly. 2018. *Machine Learning with scikit-learn Quick Start Guide: Classification, regression, and clustering techniques in Python*. Packt Publishing Ltd.

[28] Abdulhafedh, A. (2017). *A Novel Hybrid Method for Measuring the Spatial Autocorrelation of Vehicular Crashes: Combining Moran's Index and Getis-Ord Gi Statistic*. Open Journal of Civil Eng , 7, 208-221.

[29] Cory Lesmeister. 2017. *Mastering Machine Learning with R: Advanced prediction, algorithms, and learning methods with R 3.x.* Packt Publishing Ltd.

[30] M. Emre Celebi and Kemal Aydin. 2016. *Unsupervised Learning Algorithms*. Springer.