Mohammed VI Polytechnic University

Data Science and Decision Support

THIRD SEMESTER PROJECT MEMORY

Applied Machine learning's Clustering Models in Customer Relationship Management

---

ABDENOURI Khaoula

Supervisor: Professor IDRI Ali

February 9, 2021

Acknowledgment

Firstly, I would like to address my sincere thanks to Professor IDRI Ali who kindly supervised my work and supported me with the necessary knowledge to go along with the study. I am increasingly grateful for his help as without him this work wouldn't be possible.

Also a big thanks to the teaching staff of the AL KHWARIZMI department at the Mohammed VI Polytechnic University for their generosity and their sacrifices during this hard time due to the COVID-19 Pandemic.

Finally, I thank all those who have contributed directly or indirectly to the success of this work. I hope that this work is within the horizon of your expectations as well as the level of the efforts provided.

Abstract

In the context of investigating the patterns of CRM systems and their uses across industries, this study will be tackling characteristics affect the application and support CRM systems for customer knowledge creation processes. Given the importance of data management and knowledge extraction in term of maintaining a good costumer life-cycle. We will try to see at what extent can we take advantage of these gathered data to scale the way CRM is implemented nowadays

In the following will give an elaborated explanation of how we are going to preprocess the data, a deep idea about the algorithms we will be implementing and the process of evaluating the results.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

CCI	Credit Card Information Data
CCU	Credit Card Use Data
CRM	Customer Relationship Management
DM	Data Mining
E-CRM	Electronic Costumer Relationship Management
KM	Knowledge Management
ML	Machine Learning
S-CRM	Social Costumer Relationship Management
WCSS	Within-cluster sum of squares

General Intoduction

0.1 Contexte and Motivations

In essence, Customer Relationship Management or CRM is a multitude of strategies, manoeuvres and technologies used to oversee the targeted customer's interactions with the firm. Primarily, CRM is a tool to maximise both customers loyalty and their retention rate. Since business revenue and customer loyalty are positively correlated, we can also define CRM as one strategy to increase profits for a business. In fact we can say that the strong association between the custumer's satisfaction and the company's gain is the main if not the only reason why we so commonly say "Customer is King".

As foregoing, the CRM concept is very simple. Though, it can be implemented in a huge array of methods: websites, mobile calls, emails, chats and various marketing materials that can be integrated into a CRM solution. Thus, th key to a successful management of customer relationships can be no other than technology which includes a front office application that can support the Marketing and sales work, but more importantly a back office application that mainly focuses on analysing customer's data.

In today's global business settings, top supervisors invest in customer relationship management as a strategic tool to refine end to end customer service and increase customer retention rate [1]. Along side with knowledge management (KM), costumer related knowledge exploitation, they are well recognised by many leading companies as a major success factor.

In fact, on newly emerging tool to analyse our customer knowledge from the customer information we managed to gather is Data Mining. DM has been successful in terms of maintaining a competitive advantage with a better understanding of what really need and supporting the decision that will eventually raise their satisfaction.

Also, combining Customer Relationship Management with Data Mining has proven to be cost effective to gain costumers. but it is important to note that the combination for it self is not as simple as it appears to be. Extracting valid knowledge can depend on a lot of parameters that we are far from mastering as it is the case concerning cross cultural differences and the data quality when it comes to information sharing.

0.2 Problematic and Objectives

It is essential to acknowledge that Customer Knowledge relays heavily on the information shared by the customer. And even so, it is absolutely crucial to pay great interest to the model used to extract customer knowledge as a corner stone of the firm's success. To further elaborate on that, costumer knowledge directly implies a strong understanding of your market in terms

of persona building and customer segmentation.

In this study we will try to build a customer segmentation model and explore to what extend could we extract knowledge from information held by customer's bank accounts.

Towards this purpose we will use two different datasets. The first will tackle general informations about Credit Card registered in a certain bank, and the second one will give use a better insight on how are these credit cards used in terms of how much money is spent in different market sector. The main motivation of using two datasets is not only to gain better insight on how our population is acting financially but also to see how our clustering models will perform on different datasets.

0.3 Study Structure

In this study we will be proposing a solution to better cluster potential clients in a given population. This solution is fundamentally built on Machine learning in the sense that we will test various algorithms on a data available on-line to analyse and understand what is the best way to handle the data and how to effectively build a clustering model.

In the following will give an elaborated explanation of the tools we wil be using to achieve a good customer segmentation. Then we will present the strategy that we will adopt and how we will manage the sequence of algorithms. After that we will be able to present the results and give an interpretation for the outcomes. Lastly we will present an insight of what we have been able to do in this work and it's application and also a suggested further works that will potentially enrich this project.

Chapter 1

Background and Related Work

1.1 CRM Overview

Overall, when review the CRM as a practice we distinctly recognize five sections of CRM strategies. Starting first with the basic : E-CRM. At it's core, E-CRM is a way of using Internet and Web services as a channel for information and commerce strategies. It gives the opportunity to manage costumers in a smart way and get a old of their life cycle from the acquiring phase to the retaining one. It also makes data easily accessible on demand. But, E-CRM is very time consuming to realise : at least 3 years of consistent work and attention [2].

Another strategy would be CRM in KM. As previously stated, knowledge is a concept and insight extracted from the information in the data. it is also defined as the process of understanding the logic relationship between the data gathered in a certain context. Knowledge is also an interpretation and a reflection that bridges the information available with a certain call to action and a decision making which is why it is so important to in the settings of CRM. However, in order to deduct a good action plan using KM, it is essential to use the right model to extract the right valuable knowledge.

Building a model to extract knowledge from data is exactly what is known as Data Mining which is a third strategy to implement CRM using data retrieval. In addition to being a newly emerging tool to successfully extract knowledge from data it has also been recognized as a strong asset to enhance a competitive advantage compared to other tools. In fact, DM appears to be extremely useful in CRM function as it is the go-to solution when it comes to identifying costumers need and establishing a costumer segmentation. Although it is very sensitive to the quality of the data furnished by the costumer as it relies heavily on what is provided to the model.

Hence, data quality is yet another strategy to discuss. When dealing with the challenge of ensuring data quality by instinct we think of the accuracy of the data. Not to lose focus on the timeliness of the information and it's completeness and consistency. The complication of data quality is appearing to be one of the greatest challenges facing the CRM pipeline[3]. One main reason behind that is the diversity of cultures and the cross cultural differences of the targeted customers which is difficult to normalise or generalise.

Last but nor least comes the Social CRM (S-CRM). One of the channels that has been successful in maintaining a durable relationship with it's costumer is social media. Besides, it can also be a viable source of additional data resource. Nowadays, S-CRM is the primary too lto create meaningful conversations and valuable relationship between the firm and it's costumer.

Again as convenient as it is, it also has some limitations: it is not easy to access your targeted social network and the fact that you are getting in touch with a selected audience can negatively affect the quality of your data for it's going to be bias to your restricted network.

All in all, CRM is not only an opportunity to get a hold of all the factors affecting your customer but it is also the pillar on which can stand the success of any given firm. Thus, we will discuss the Highlights of the state of art in the CRM context to have an insight about what has been done to correctly implement this promising strategy.

1.2 Clustering tools Description

1.2.1 K-Means Algorithm

K-means is with no doubt the most common classification model. Fundamentally, it tries to group the instances into K groups so as to maximise the distance between each groups and to minimise the distance between features of each group. Intuitively we understand that the performance of the algorithm will depend on the metric used to measure the distances.

To give a better idea about how this algorithm proceeds to classify some features we will go step by step through the process:

1. We first choose the number K of clusters we want to identify in our data. If the features of the data are represented in a limited dimension (three or two features), a good approach would be to visualise the data to help you spot how many clusters you will want to group. We will give as an example the data represented in the figure **??**. We will then initialise K center points for our clusters.

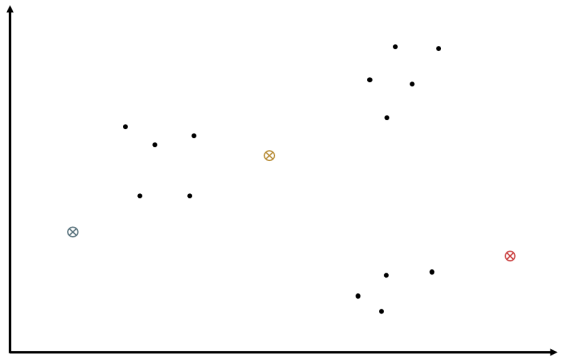

Figure 1.1: Initialisation of the KMeans center points

2. Now for each instance we will compute its distance with all the center points to find the closest and associate them to the same cluster as shown in the figure below.

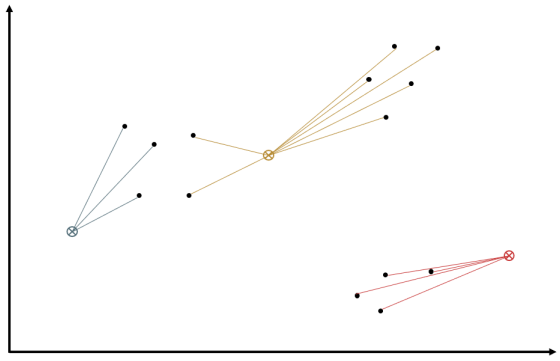

Figure 1.2: Initialisation of the KMeans center points

3. Based on the instance grouped in the same cluster we will update the position of the center point using the mean of the instances to which it is associated.

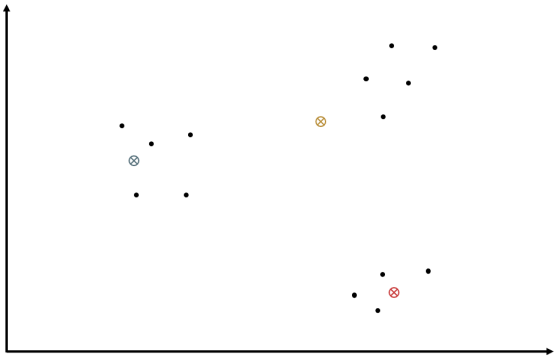

Figure 1.3: Bundling of the instances

4. We will keep iterating on the second and third steps expecting from the centroids to slowly find its place in the center of each clusters.

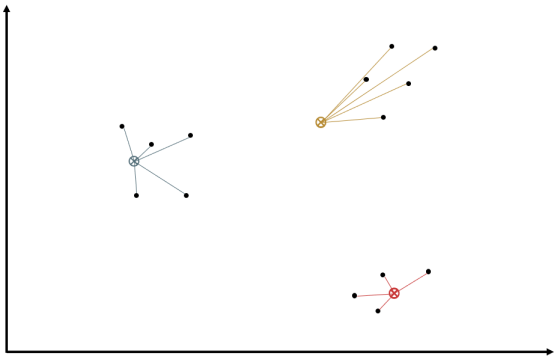

Figure 1.4: Centroid Update

5. Lastly we end the iterations when we conclude that the centroids are not being updated from an iteration to an other. We can then deduce the clusters of our data.

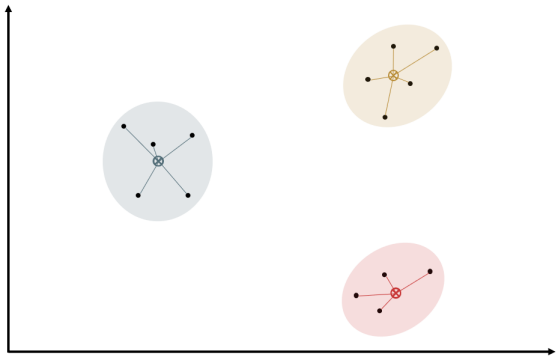

Figure 1.5: K-means clustering outcome

Algorithm 1 K-means Algorithm

Requirments: Dataset D; Distance Function d; Int K;

```
Let C_i be K randomly initialized centroids
Let L_i be K array lists associated with each centroid
while at least one centroid has changed its position do
  for all observation X_i in the dataset D do
    Find C_j so that d(X_i, C_j) is minimum
    Append L_j with X_i
  end for
  Update the position of each centroid C_i with the mean of all the observations in L_i
end while
return a dataset with all the observations and their associated class
```

As shown in Algorithm 1, in order to find the most resembling observation in our dataset to one of our centroids, we need to define a distance metric. There are several distancing norms applicable in this algorithm among which the most common is the Euclidean distance.

For $X = (x_1, ..., x_n) \in R^n$ and $Y = (y_1, ..., y_n) \in R^n$

$$d(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

One necessary condition to use norms as such is to only have numerical features describing the observations which is not always the case. In the following we will introduce the common solution for this problem.

1.2.2 K-Modes Algorithm

When dealing with categorical features, the K-Means algorithm is pretty limited in terms of calculating the dissimilarity between the observations. On common solution to get around this problem is to tweak the model so as to adapt the metric to the nature of attributes we

are working with. For instance, the K-Mode algorithm is a modified version of the K-Mean algorithm that instead of using the Euclidean, the Manhattan or the Minkowski distance, it uses the Hamming metric defined as the minimum number of substitution needed to change one character to another. It is commonly represented by the minimum numbers of strings between two pints in a n-bit binary space.

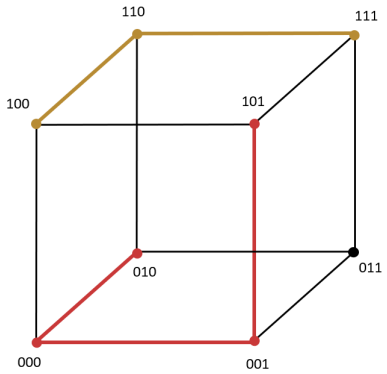

Figure 1.6: Hamming distance in a 3-bit binary cube

For example hamming distance between " 0 1 **1** 0 1 **0** " and " 0 1 **0** 0 1 **1** " is 2

Thus, to update the centroids at each iteration, it doesn't make sense to compute the mean of a certain distribution. Naturally the Centroids will be updated with the mode of the observations in the same cluster. Moreover, It is interesting to note that the Hamming metric can be applied to ordinal data too.

Algorithm 2 K-modes Algorithm

Requirments: Dataset D; Distance Function d; Int K;

Let C_i be K initialized centroids from random existing observations in the dataset
Let L_i be K array lists associated with each centroid
while at least one centroid has changed its position **do**
  for all observation X_i in the dataset D **do**
    Find C_j so that $Hamming(X_i, C_j)$ is minimum
    Append L_j with X_i
  end for
  Update the position of each centroid C_i with the mode of all the observations in L_i
end while
return a dataset with all the observations and their associated class

1.2.3 Hierarchical Algorithm

In contrast with the previously stated algorithms, the Hierarchical clustering is an algorithm that has a Bottom-up approach as it starts mapping two elements at a time utile it gathers the hole picture of the clusters. In the plus side, the hierarchical clustering doesn't require a specification of the number of clusters that we want to work on. Instead it visualised the information

loss at each step and leaves the decision makers detect the most suitable segmentation. Once again, in order to group two elements in the same cluster the Hierarchical algorithm finds the closest elements according to a certain metric.

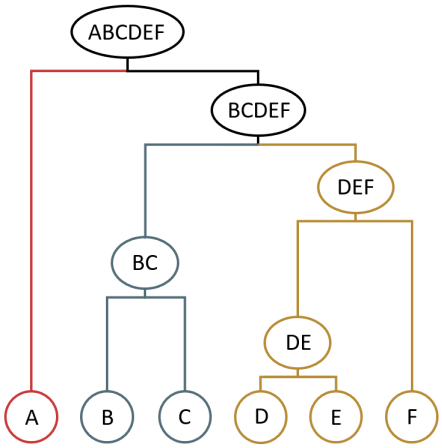

Figure 1.7: Traditional representation of the Hierarchical clustering

Algorithm 3 Hierarchical Algorithm

Requirments: Dataset D; Distance Function d;

Let L_0 be an array of all the observations X_i in D as initial clusters
i = 0
while L contains more than just one element **do**
  Find the closest two clusters in L and group them
  Update the new structure in a new array L_i
  i++
  Compute the information loss
end while
return the succession of structures L_i combined with their information loss

1.3 Presentation de la literature

In Custuerm Relationship Management, Data Mining tools have proven to be extremely useful in term of extracting knowledge from the huge mass of data available in industries nowadays. But so far, Business Intelligence and knowledge discovery are the two academic disciplines that are the most common when it comes to customer management.
Primarly there are four main basis on which CRM stands:

- Customer Identification:
  Target customer analysis and customer segmentation

- Customer Attraction:
  Direct Marketing

- Customer Retention:
  Loyalty Program, One-to-One Marketing and Complaints management

- Customer Development:
  Customer lifetime value, Up/Cross selling and Market basket Analysis

Over the decade, Machine learning has been successful in supporting these 4 pilars on which depends a good costumer life cycle. Manly, machine learning contribution reside in applying visualisation tools to have a better sense of the distribution of the variables for the observations, also Regression and forecasting are very commonly used to better attract costumers. Aside from that there is Classification and Clustering that are widely applied to identify our present costumers and the potential ones.

In this study we will try to evaluate at what extent can we apply clustering in Target costumer analysis and how it could help us have a better segmentation for our customers.

Chapter 2

Experimental Design

2.1 Credit Card Data Description

2.1.1 Context

In this study we will be working on two data sets imported from the Kaggle Website. These two datasets tackle Users Credit Card Information [4] and Credit Cards Use [5]. Throughout these two sets our goal is to gain understanding of how people manage their bank account and how they carry on their spending. This could be used to localise potential costumers concerning a certain market study in term of identifying the users financial behaviours and the way they handle their cash flow in regards to their interest. As it can also be the guideline for CRM strategies and customer attraction.

The Credit Cards Information dataset contains about 9K observations with 17 features that are mostly numerical. Meanwhile the Credit Card Use dataset hold in 19K observations and 19 features associated to it.

2.1.2 Features description

Starting by the Credit Card Information dataset, we will go through the 17 features and present a brief description for each one:

Balance: Balance amount left in their account to make purchases.

Balance_frequency: How frequently the Balance is updated, score between 0 and 1.

Purchases: Amount of purchases made from account.

Oneoff_purchases: Maximum purchase amount done in one-go.

Installments_purchases: Amount of purchase done in instalment.

Cash_advance: Cash in advance given by the user.

Purchases_frequency: How frequently the Purchases are being made, score between 0 and 1.

Oneoff_purchases_frequency: How frequently Purchases are happening in one-go.

Purchases_installments_frequency: How frequently purchases in installments are being done.

Cash_advance_frequency: How frequently the cash in advance being paid.

Cash_advance_trx: Number of Transactions made with "Cash in Advanced".

Purchases_trx: Number of purchase transactions made.

Credit_limit: Limit of Credit Card for user.

Payments: Amount of Payment done by user.

Minimum_payments: Minimum amount of payments made by user.

Prc_full_payment: Percent of full payment paid by user.

Tenure: Tenure of credit card service for user.

Likewise, we will present a brief overview of the 19 attributes elaborated for each observation in the Credit Card Use data:

AVG_BALANCE: Balance amount left in their account to make purchases.

TENURE: Tenure of credit card service for user.

NUM_TRANS: Number of Transactions.

ACCESSORIES: Purchases concerning accessories.

APPLIANCES: The amount of appliance loan.

CULTURE: Purchases concerning culture.

GAS: Purchases concerning gas.

BOOKS: Purchases concerning culture.

APPAREL: purchases concerning apparel.

FITNESS: Purchases concerning fitness.

EDUCATION: Purchases concerning education.

ENTERTAINMENT: Purchases concerning entertainment.

FOOD: purchases concerning food.

HEALTH: purchases concerning health.

HOME_GARDEN: purchases concerning home garden.

TELCOS: Purchases concerning telecoms.

TRAVEL: purchases concerning traveling.

PURCHASES_AMOUNT: the amount of purchases.

Loyalty: Measurement of customers loyalty, score between 0 and 1

2.1.3 Statistical Description and overview of some features

Before taking a look at any of the features and their characteristics in both datasets, let's first deal with the existing missing values existing in the Credit Card Information data. We have 313 missing values in the **Minimum payments** feature and one other missing value in **Credit limit** in contrast to the 19K available information in the data. Therefore, we could simply drop the missing values as their proportion is insignificant.

We can now have a summary of the statistical description of the our datasets starting first by the Credit Card Information Data:

Feature	**Mean**	**Std**	**Min**	**25%**	**50%**	**75%**	**Max**
Balance	1601.22	2095.57	0	148.09	916.85	2105.19	19043.13
Balance_frequency	0.89	0.21	0	0.91	1	1	1
Purchases	1025.43	2167.11	0	43.37	375.41	1145.98	49039.57
Oneoff_purchases	604.90	1684.30	0	0	44.99	599.10	40761.25
Installments_purchases	420.84	917.24	0	0	94.78	484.14	22500.00
Cash_advance	994.17	2121.45	0	0	0	1132.38	47137.21
Purchases_frequency	0.49	0.40	0	0.08	0.50	0.91	1
Oneoff_purchases_frequency	0.20	0.30	0	0	0.08	0.33	1
Purchases_installments_frequency	0.36	0.39	0	0	0.16	0.75	1
Cash_advance_frequency	0.13	0.20	0	0	0	0.25	1.5
Cash_advance_trx	3.31	6.91	0	0	0	4	123
Purchases_trx	15.03	25.18	0	1	7	18	358
Credit_limit	4522.09	3659.24	50	1600	3000	6500	30000
Payments	1784.47	2909.81	0.04	418.55	896.67	1951.14	50721.48
Minimum_payments	864.30	2372.56	0.01	169.16	312.45	825.49	76406.20
Prc_full_payment	0.15	0.29	0	0	0	0.16	1
Tenure	11.53	1.31	6	12	12	12	12

Table 2.1: Credit Card Information features description

This representation of the data is only possible because all the features are of numerical type. Else we would have to add a description of the quantitative features. For instance it is also the case for the Credit Card Use datasets. Accordingly we will proceed with the same description for our second data:

Feature	Mean	Std	Min	25%	50%	75%	Max
AVG_BALANCE	2249.45	2427.72	0	512.46	1375.58	3135.04	27615.33
TENURE	11.38	1.47	6	12	12	12	12
NUM_TRANS	18.95	35.54	-0.97	0	7.27	24.95	1301.27
ACCESSORIES	176.55	776.65	0	0	0	78.63	34398.47
APPLIANCES	219.65	1006.84	0	0	0	62.56	49262.11
CULTURE	5.75	103.43	0	0	0	0	8767.02
GAS	105	570.15	0	0	0	0	19416.92
BOOKS	31.89	657.16	0	0	0	0	79068.88
APPAREL	477.69	1377.34	0	0	0	488.02	70464.64
FITNESS	2.83	70.80	0	0	0	0	6557.90
EDUCATION	8.91	322.94	0	0	0	0	38718.09
ENTERTAINMENT	117.17	1653.57	0	0	0	0	149716.89
FOOD	185.19	829.93	0	0	0	0	48731.64
HEALTH	101.40	556.36	0	0	0	0	20979.62
HOME_GARDEN	117.69	555.76	0	0	0	0	14901.72
TELCOS	158.54	667.88	0	0	0	0	27035.76
TRAVEL	181.24	988.00	0	0	0	0	59238.41
PURCHASES_AMOUNT	1889.56	3742.65	0	0	997.39	2333.73	166608.31
Loyalty	0.27	0.44	0	0	0	1	1

Table 2.2: Credit Card Information features description

We notice that in both cases the standard deviation is taking remarkably big values which is surly going to result into a lot of outlier in the variety of features we have. In the following, we will have the chance to look closely into this and potentially finding the appropriate solution for it.

2.2 Experimental Process

Now that we have clearer idea of h ow the variables are distributed through the data we can now start experimenting on what we can do to better prepare the datasets for the training models. Since we are working with two different datasets they will require different manurers for the preprocessing. We will go through the Preprocession sequentially for the two datasets starting by the CCI data (Credit Card Information)

2.2.1 Preprocessing phase on the Credit Card Information Data

(i) Feature selection

In order to have a successful training it is essential to have consistent information. Thus it is unnecessary o drag features with redundant information to our learning process. We will then proceed with some filters to detect the features that are unnecessary. For that we can go through the Correlation filter . Correlation describes the linear relationship between two entities. A high correlation between two features means that we can constructively predict one from the other. They then provide the same redundant understanding in reference to the target. Again, one of them should be dropped from the data.

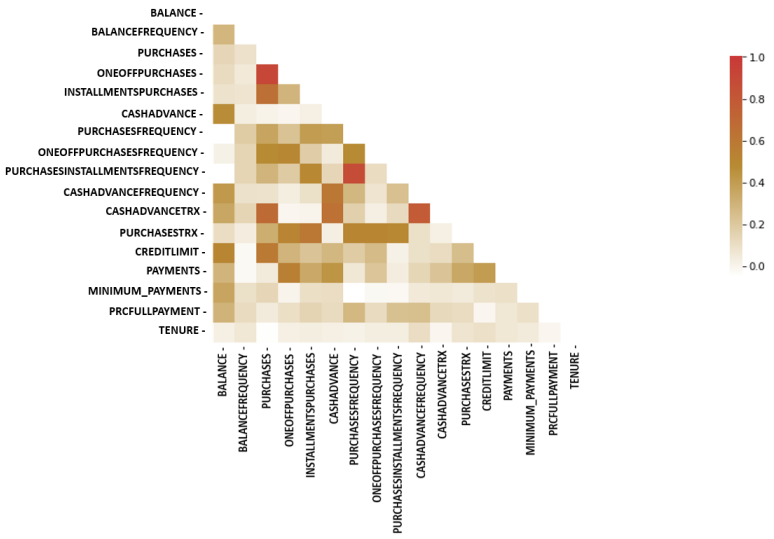

Figure 2.1: CCI data correlation heat-map

We notice the highest correlations we have are

$$Cor(Oneofpurchases, Purchases) = 0.92$$

$$Cor(Purchasesinstalmentsfrequency, Purchasesfrequency) = 0.86$$

$$Cor(Cashadvancefrequency, Cashadvancetrx) = 0.80$$

we will naturally drop the "Purchases", "Purchases_instalments_frequency" and "Cash_advance_trx" features.

(ii) Discretization of some quantitative features

To go back the the problem we faced when studying the statistical description of the data, we could first make sure that indeed we are dealing with outlier by representing the distribution of a feature using the box plot.

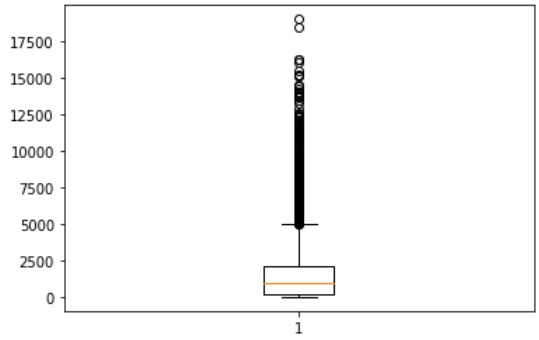

Figure 2.2: Box plot distribution for the "Balance" feature in the CCI dataset

One of the common ways to deal with outliers is to discretize the features. And since outliers are present in all our features we will have to go with the discretisation of all our features into ranges mostly going from 0 to 5 (where 0 is very low and 5 is very high)

2.2.2 Preprocessing phase on the Credit Card Use Data

(i) Feature selection

Similarly, we will apply the Correlation filter to our dataset to make sure the information in our data is truly consistent and redundancy free.

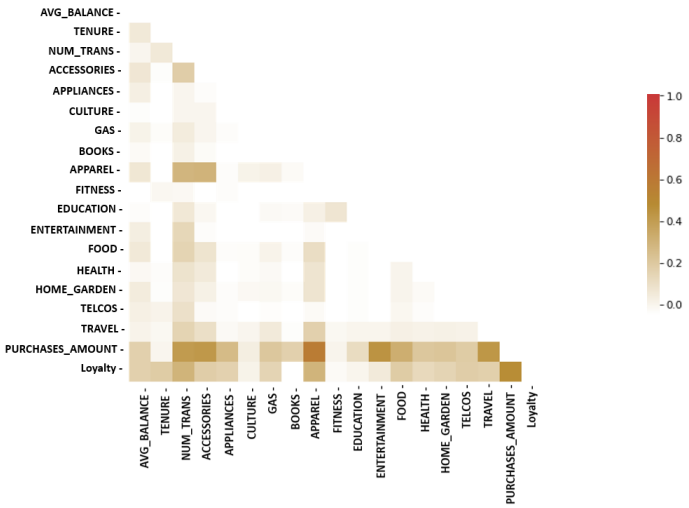

Figure 2.3: CCU data correlation heat-map

For the CCU data we notice that the correlation between the variables is not quite significant. Consequently, we will proceed with the rest of the preprocessing phase with all our.

(ii) Discretization of some quantitative features

As suspected, the CCI data has as much outliers as the previous data

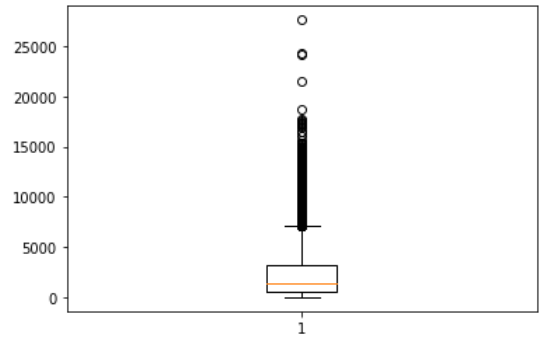

Figure 2.4: Box plot distribution for the "Balance" feature in the CCU dataset

As we mentioned previously, to surpass this problem we will simply discretize all the features in our CCU data.

2.2.3 Learning phase

- **K-means**

We will first try to train the K-means algorithm on our data before preprocessing the data to see if there was a significant added vale to preprocess our data, our just to understand the effects of the preprocessing phase on the learning phase. The algorithm will be trained by initialising the K in the following range [2, 30]

- **K-mode**

After preprocessing our data, naturally we cannot use the K-means algorithm as our dataset in now composed of categorical features. Accordingly we will train the K-mode algorithm using the Hamming metric that could be used for ordinal features and initialise the K parameter within the following range [2, 30]

- **Hierarchical**

We will also train our Hierarchical training model after preprocessing our data. Obviously this time we will not have to put a specification on the number of clusters but we will have to define a metric. In our case we will use the 'euclidean' metric

- **K-modes combined with the Hierarchique**

It is proven that the Hierarchical model does not perform in a good way when dealing with a relatively big number of observations [6]. That is why a recomended structure to use when it has to do with a lot of observations is to first use the K-means/K-modes to reduce the N number of observations into K classes (K << N) then apply the hierarchical clustering of these K classes. For the CCI data we will reduce the number of Observations to K = 300 and in the CCU data to K = 800.

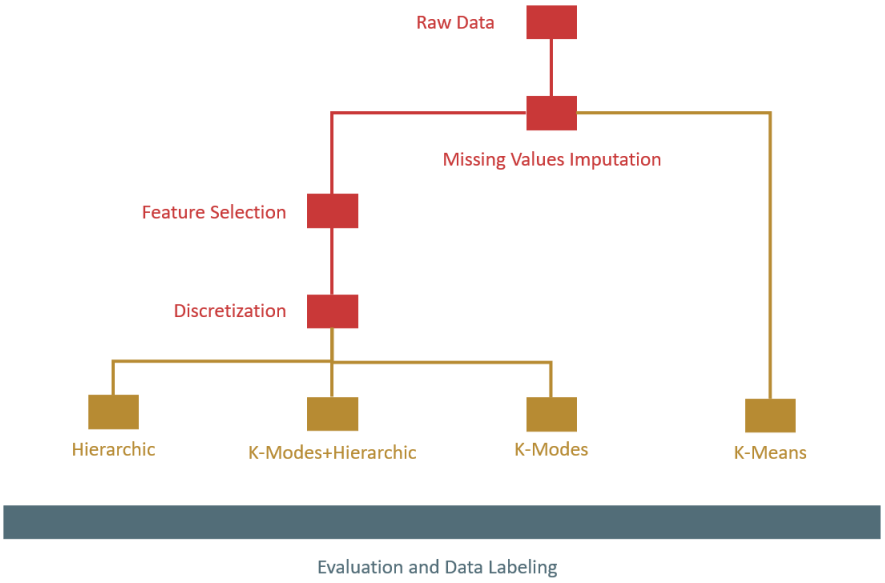

Figure 2.5: Summary diagram of the experimental process

2.3 Performance Criteria

2.3.1 Silhouette Coefficient

The silhouette coefficient is a score used to evaluate the clustering outcome of a giving model. This Score takes values from -1 to 1. The ideal value of a silhouette score is 1 and the worst possible silhouette value is equal to -1. Generally this score is used to know if a certain point is correctly classified or it has been misclassified. Its calculation with respect to a certain observation is done as follow:

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

$a(i)$ = Average distance between the observation i and the observations in the same cluster as i
$b(i)$ = Average distance between the observation i and the observations in the nearest cluster

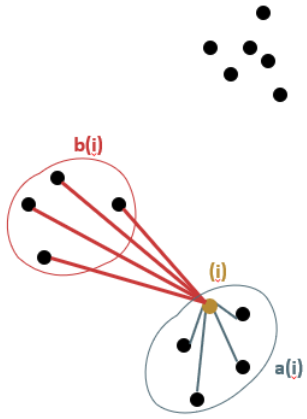

Figure 2.6: Silhouette score computation representation

If we wanted to compute the silhouette score of the hole classification model we should compute the average of all the silhouette coefficients for all the observations

$$Silhouette = \sum_i s(i)$$

WCSS Score

The Within-cluster sum of squares WCSS is simply as its name is suggesting the sum of square distances between observations within the same cluster. The dilemma of this score is that if we go to the extreme cases and choose to associate to each observation it's own cluster then WCSS = 0, because in this case there is as much clusters as observations and in each observation we will find one and only one observation. And if we gather all the observations in one and only one cluster then WCSS $\simeq \infty$.
This makes us think that whether we maximise or minimise the WCSS Score we are not going to have the optimal solution. Well the answer is that instead of focusing only on the WCSS Score we should pay attention to the number of clusters associated with this WCSS Score. The goal

that we should aim to reach the optimal solution is to minimise the WCSS score and minimise the number of clusters.

Elbow Criteria

Finding the optimal value in this case is not quite easy. A suggested method to find this optimum is to plot the WCSS Scores accordingly with the number of clusters. The resulted plot usually looks like an elbow where is we try to minimise both variables the solution should be the tip of the elbow as shown in the figure bellow.

Figure 2.7: Elbow Criteria Representation

Chapter 3

Implementation Results

Now that we have a clear idea about the data in term of content and features and that we have presented an elaborated description of our experimental design, we will now proceed with the execution of our models. We will also try to present a detailed view of of the results that we had for out classification with a brief observation for each result.

3.1 Results Presentation

We will start by mainly unveiling the outcome of the experimental process going form an algorithm to another to find the optimal number of clusters to use in each case.

3.1.1 K-Means

You may recall that the K-Means Clustering is designed to be applied directly to the raw data before preprocessing. The aim of its application is to understand the added value of the preprocessing in our case.
There fore we will introduce the WCSS Score of it's application fore both the CCI and CCU data sets:

- **CCI Dataset**

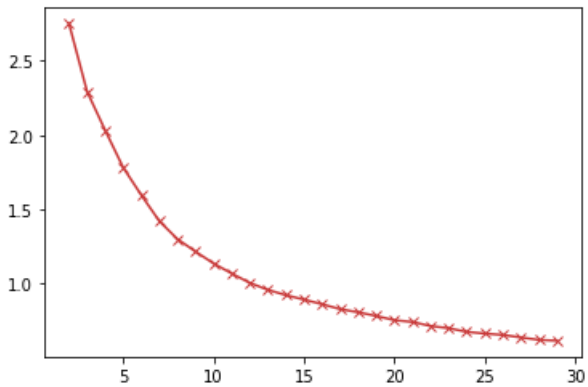

Figure 3.1: WCSS Score for the K-means algorithm on the CCI dataset

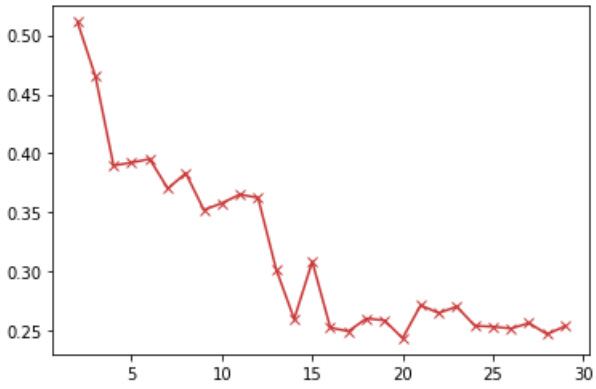

Figure 3.2: Silhouette Score for the K-means algorithm on the CCI dataset

We can notice that as expected the plot of the WCSS score takes the shape of an elbow. So if we were to choose an optimal value of clusters in this case it would be 8 clusters with a Silhouette score of 0.38

- **CCU Dataset**

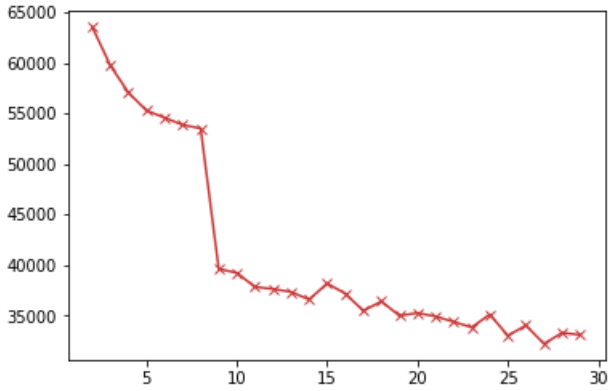

Figure 3.3: WCSS Score for the K-Means algorithm on the CCU dataset

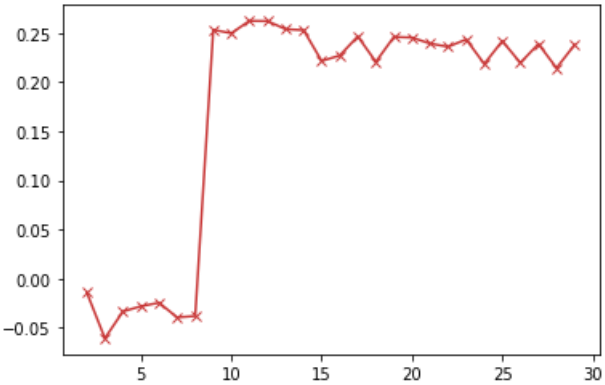

Figure 3.4: Silhouette Score for the K-Means algorithm on the CCU dataset

And Concerning the CCU Dataset the K-Means clustering algorithm shows that the custumers should be segmented into 9 clusters with a Silhouette score of 0.25

3.1.2 K-Modes

As Designed the K-Modes algorithm is executed after preprocessing the data sets. We will elaborate in the following the results of its implementation.

- **CCI Dataset**

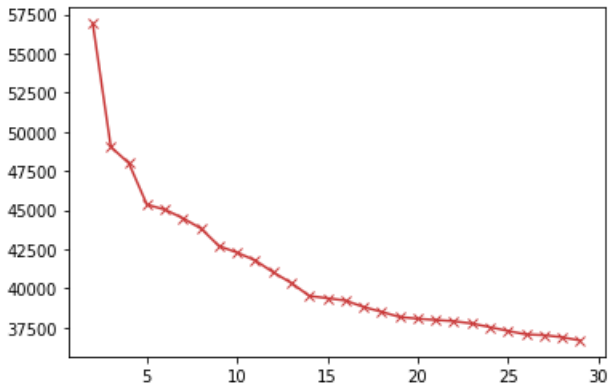

Figure 3.5: WCSS Score for the K-modes algorithm on the CCI dataset

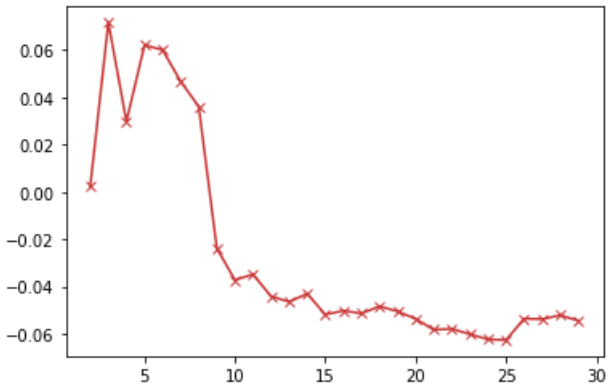

Figure 3.6: Silhouette Score for the K-modes algorithm on the CCI dataset

After preprocessing the data and applying the K-Modes algorithms, we find that according to this model the optimal number of cluster is 5 clusters with a Silhouette score of 0.6 for the CCI Data

- **CCU Dataset**

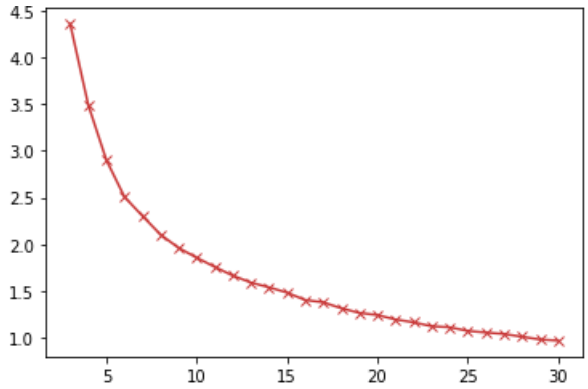

Figure 3.7: WCSS Score for the K-Modes algorithm on the CCU dataset

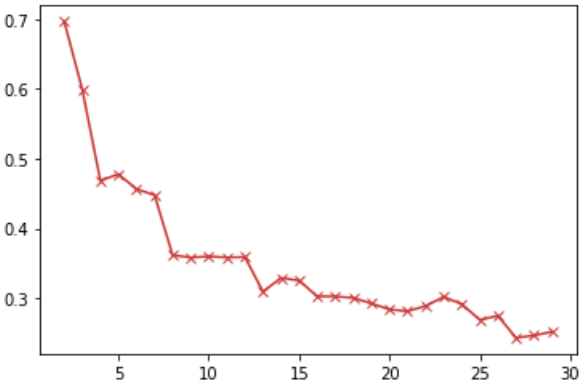

Figure 3.8: Silhouette Score for the K-Modes algorithm on the CCU dataset

Concerning the CCU Data the optimal number of clusters to be used is 7 clusters with a Silhouette score of 0.44

3.1.3 Hierarchic

Aside from the K-Modes clustering another algorithm we suggested to work with inorder to proceed with the customers segmentation is the Hierarchical clustering. This Model shows the following results:

- **CCI**

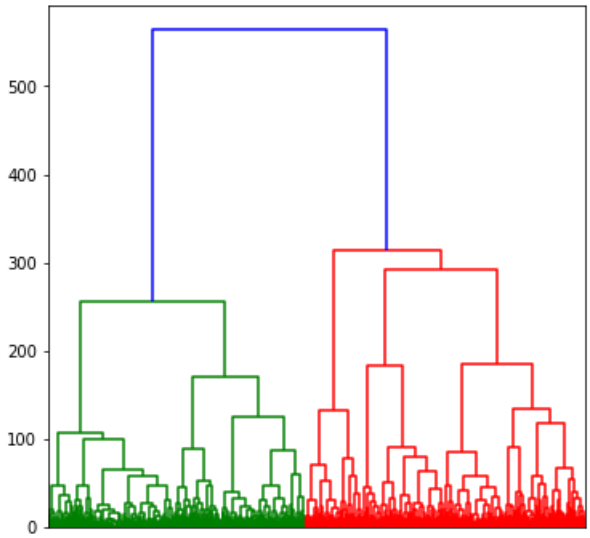

Figure 3.9: Hierarchical representation of the CCI dataset

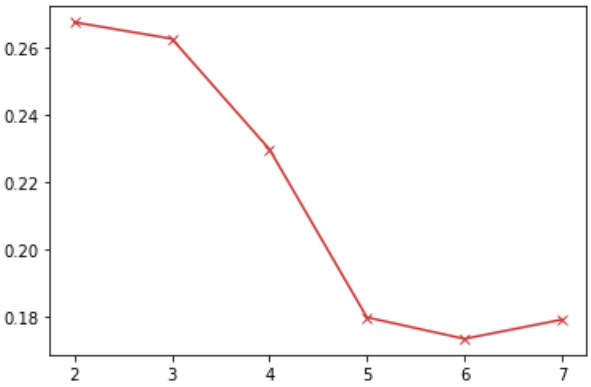

Figure 3.10: Silhouette Score for the Hierarchical algorithm on the CCI dataset

In regards of the Hierarchical clustering, we can see that the best clustering would be if we split the data into 2 clusters with a Silhouette score of 0.25 on the CCI Dataset

- **CCU**

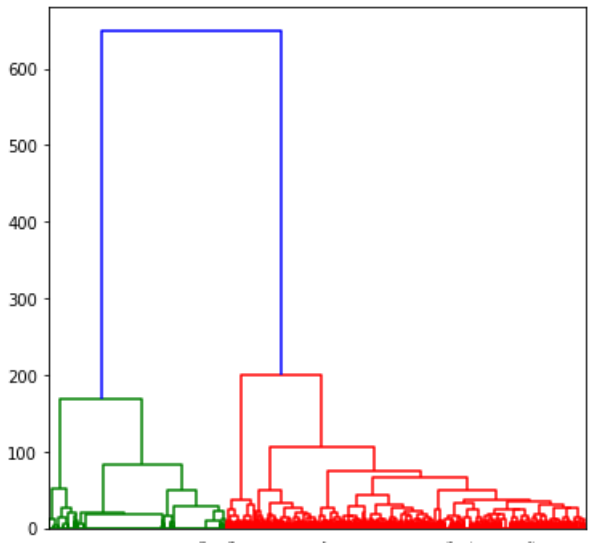

Figure 3.11: Hierarchical representation of the CCU dataset

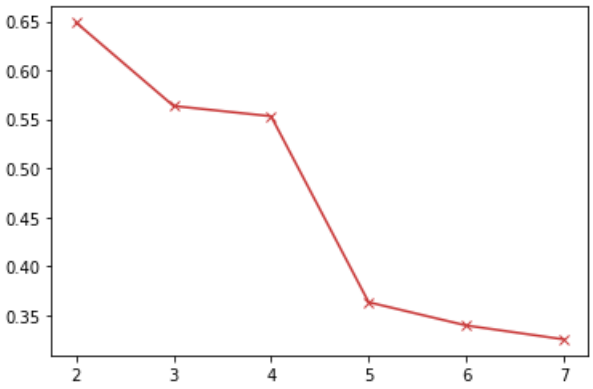

Figure 3.12: Silhouette Score for the Hierarchical algorithm on the CCU dataset

Also according to this model the CCU Dataset should be clustered into 4 clusters with a 0.56 Silhouette score.

3.1.4 K-Modes combined with Hierarchic clustering

Lastly, we will try to combine the previously elaborated algorithms and try to see if there is any remarkable difference that we could elaborate on.

- **CCI**

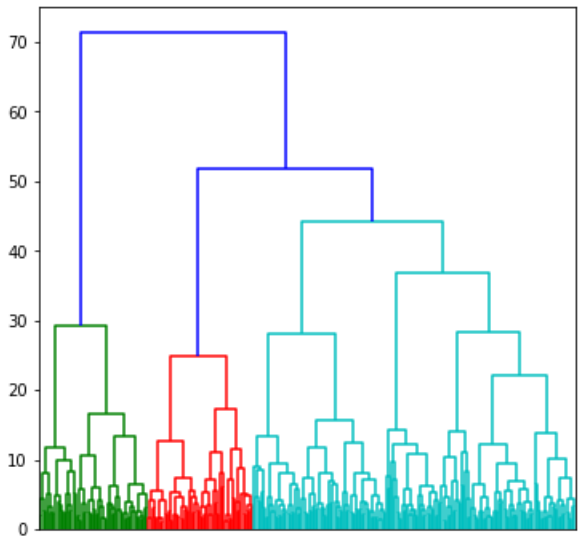

Figure 3.13: Hierarchical representation of the CCI dataset after applying the K-Modes clustering

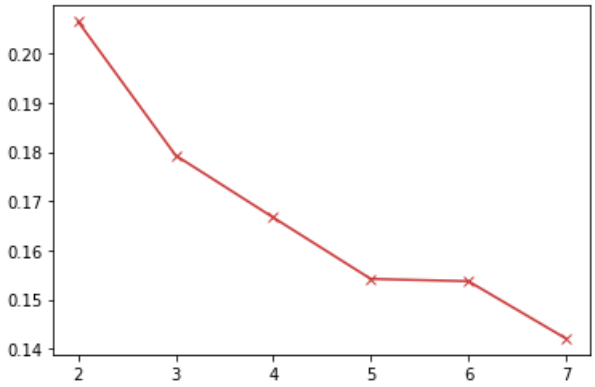

Figure 3.14: Silhouette Score for the K-Modes combined with Hierarchical algorithm on the CCI dataset

We can see that combining the two clustering models gives us a result that suggests having 3 clusters with a Silhouette score of 0.18

- **CCU**

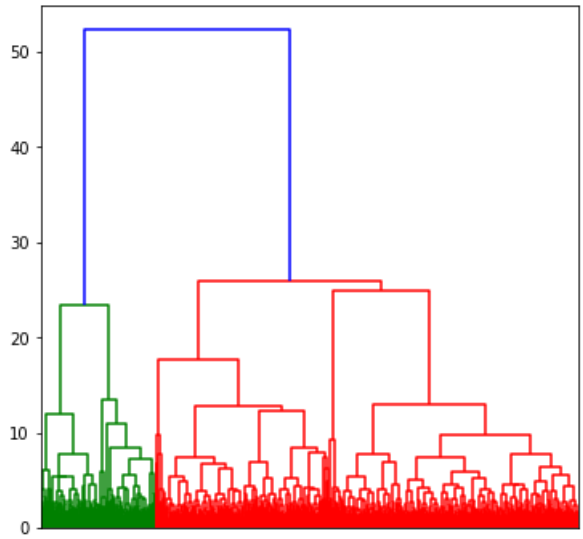

Figure 3.15: Hierarchical representation of the CCU dataset after applying the K-Modes clustering

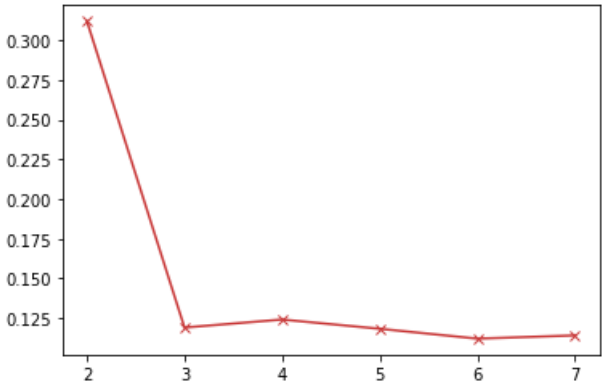

Figure 3.16: Silhouette Score for the K-Modes combined with Hierarchical algorithm on the CCU dataset

Last but not least, this model is indicating that the CCU dataset should be clustered into 2 clusters with a Silhouette score of 0.32

3.2 Results Discussion

To further gain an understanding on the model's performance and to understand the results previously presented. We will look closer at each model and decide which one performed the best for each dataset.

3.2.1 K-Means

If we look back at the results presented for the K-Means clustering we would see that they don't really fall far from the results for the different other algorithm.

CCI Data		CCU Data	
Number of Clusters	**Silhouette Score**	**Number of Clusters**	**Silhouette Score**
8	0.38	9	0.25

Table 3.1: Results on the performance of the K-Means clustering for both the CCI and CCU datasets

But we cannot ignore the fact that the number of cluster resulting from this algorithm is relatively larger with a lower Silhouette score concerning both the CCU and CCI datasets. Infact this effect generally happens when dealing with outlier. In conclusion, preprocessing our data has allowed us to have a better distribution for our data resulting in a better silhouette score associated with a lower number of clusters

3.2.2 K-Modes

Now for the K-Modes clustering, we could see that the performance is much better and the preprocessing really played a role in improving the results.

CCI Data		CCU Data	
Number of Clusters	**Silhouette Score**	**Number of Clusters**	**Silhouette Score**
5	0.6	7	0.44

Table 3.2: Results on the performance of the K-Means clustering for both the CCI and CCU datasets

As we see the number of clusters resulting is relatively low with a much better Silhouette Score

3.2.3 Hierarchic

We will now introduce the results of training the Hierarchic model on both datasets:

CCI Data		CCU Data	
Number of Clusters	**Silhouette Score**	**Number of Clusters**	**Silhouette Score**
2	0.25	4	0.56

Table 3.3: Results on the performance of the Hierarchical clustering for both the CCI and CCU datasets

We Notice that the Hierarchical clustering has failed to correctly cluster the observations in the first dataset (CCI dataset) but in the other hand has done a good job clustering the features for the CCU dataset. As previously munitioned, the Hierarchical clustering can sometimes be confused by the large number of features. We may suggest that if we try to reduce the number of by first applying the K-Modes algorithm we will be able to improve the performance of our model.

3.2.4 K-Modes combined with Hierarchic clustering

Let's test our hypothesis and see if indeed we can improve the performance of our Hierarchical clustering.

CCI Data		CCU Data	
Number of Clusters	**Silhouette Score**	**Number of Clusters**	**Silhouette Score**
3	0.18	2	0.32

Table 3.4: Results on the performance of the Hierarchical clustering for both the CCI and CCU datasets

In our cases we can notice that The K-Modes contribution in this case has only confused the learning process of the Hierarchical clustering. It is important to note that it is not always the case. Initially the K-modes algorithm could stand for the reinforcement of the model's performance. But sometimes and as it is our case, due to the complex distribution of our features K-modes could also sabotage the performance of our learning process.

3.2.5 Results summary

To wrap up our results, we want to display the performances of each Algorithm for each dataset and compare between them.

Name of the algorithm	CCI Data		CCU Data	
	Number of Clusters	**Silhouette Score**	**Number of Clusters**	**Silhouette Score**
K-Means	8	0.38	9	0.25
K-Modes	5	0.6	7	0.44
Hierarchic	9	0.25	4	0.56
K-modes + Hierarchic	3	0.18	2	0.32

Table 3.5: Results on the parameter tuning of the ANN algorithm

We conclude that with the the best performance for the CCI Dataset is noted for the K-Modes algorithm that clusters the data in 5 clusters with a Silhouette score of 0.6
Lets take a closer look at the resulting clusters:

Cluster Number	0	1	2	3	4
Tenture	11.41	11.65	11.39	11.56	11.88
Balance_range	1.71	2.58	2.85	2.45	2.40
Oneoff_purchases_range	0.56	1.40	0.17	1.64	2.74
Installments_purchases_range	1.33	0.50	0.14	0.73	2.13
Cash_advance_range	0.55	1.39	2.42	0.95	0.77
Credit_limit_range	3.38	4.09	3.63	3.53	4.69
Payments_range	1.95	2.49	2.33	2.27	3.44
Minimum_payments_range	1.44	1.75	1.84	1.67	1.66
Balance_frequency_range	8.80	8.75	9.23	9.42	9.78
Purchases_frequency_range	7.45	3.45	0.76	5.66	9.50
Oneoff_purchases_frequency_range	1.28	2.10	0.34	3.78	6.79
Cash_advance_frequency_range	0.70	1.68	2.94	1.34	0.94
Prc_full_payment_range	2.64	0.68	0.53	0.90	3.17
Purchases_TRX_RANGE	2.87	1.85	0.39	2.65	5.66

Table 3.6: The mean Observation for each detected Cluster on the CCI dataset using the K-Modes clustering

And if we pay close attention to the cluster's destribution reguading each feature we will be able to detect the following labels for each cluster:

- Cluster Number 0: People who have the lowest balance but do purchase very frequently and do not use cash in advance.

- Cluster Number 1:People who have a high credit card limit with a high balance and do not use cash in advance.
- Cluster Number 2:People who have a high credit card limit with a high balance but do not purchase frequently and use cash in advance.
- Cluster Number 3:People who have a high credit card limit with a high balance and do purchase very frequently.
- Cluster Number 4:People who has a high credit card limit who purchases the most often

In regards to the CCU Dataset the best performing algorithm was the Hierarchic algorithm that clustered the data into 4 clusters with a Silhouette score of 0.56
Once again we will Lets take a closer look at the resulting clusters:

Cluster Number	0	1	2	3
TENURE	11.82	11.90	7.32	7.88
Loyalty	1	0	0	0
AVG_BALANCE_RANGE	1.48	1.69	1.24	1.22
NUM_TRANS_RANGE	1.05	0.03	1.01	0.02
ACCESSORIES_RANGE	0.47	0	0.37	0
APPLIANCES_RANGE	0.45	0	0.39	0
CULTURE_RANGE	0.02	0	0.02	0
GAS_RANGE	0.20	0	0.13	0
BOOKS_RANGE	0.13	0	0.06	0
APPAREL_RANGE	0.65	0	0.54	0
FITNESS_RANGE	0.02	0	0.01	0
EDUCATION_RANGE	0.01	0	0	0
ENTERTAINMENT_RANGE	0.17	0	0.10	0
FOOD_RANGE	0.35	0	0.25	0
HEALTH_RANGE	0.23	0	0.23	0
HOME_GARDEN_RANGE	0.28	0	0.20	0
TELCOS_RANGE	0.25	0	0.13	0
TRAVEL_RANGE	0.26	0	0.16	0
PURCHASES_AMOUNT_RANGE	6.93	0.01	6.69	0.00

Table 3.7: The mean Observation for each detected Cluster on the CCU dataset using the Hierarchic clustering

Again we will look deeper into the distribution of the features in each class to gain insite on the label to use for each cluster.

- Cluster Number 0: People who have a high balance and spend their money in mostly everything specially in accessories, appliance and apparel.
- Cluster Number 1:People who have a high balance but mostly do not spend their money.
- Cluster Number 2: People who have a low balance and spend their money in mostly everything specially in accessories, appliance and apparel.
- Cluster Number 3:People who have a low balance and mostly do not spend their money.

3.3 Study Limitation

The results provided by applying the K-modes and the Hierarchical clustering model are pretty satisfactory. However, the results are not ready to be generalized as there are several data sets concerning Users Credit Cards available on-line. It is then recommended to train this model on the maximum of data that could be gathered before actually generalizing the outcoming clusters. It is important to keep in mind that our study only gives an idea of what works better for this type of data and to what extend we can successfully segment clients based on their credit cards information and spendings. Moreover, the avenue of more data could reveal some trends we were not aware of with just a limited data.

Chapter 4

Conclusion and Future Work

In a few word, this work proposes a solution to support the segmentation of clients in terms of their financial capacities and their main spending interests. It also tackles the problem of choosing the right way to preprocess the data, the right algorithm to build the model and the right parameters to use on a certain models. The proposed solution is pretty iterative and consisted of building different segmentations of the same data resulted from processing different algorithms.

Consequently, each segmentation version will yeald a diffrent number of cluster and Silhouette score that we use to evaluate interpret and discuss the performance of each algorithm. To Classify our observations, we tested the performance of diverse learning algorithms namely: the K-Means Clustering Algorithm; the K-Modes Clustering Algorithm; the Hierarchical Clustering Algorithm and the Combination between the K-modes and the Hierarchical clustering.

Eventually after gathering all the results we compared the different performances based on their Silhouette score on both Credit Card Information Dataset and the Credi Card Use Dataset. This final scored allowed us to rank the models and as a result the K-Modes performance was buy far the best among the models for the CCI dataser with a silhouette score of 0.6 and the Hierarchic clustering showed the beat results when it comes to cluster the CCU Dataset with a Silhouette score of 0.56.

Moreover, further work on this problematic will consist of testing more learning algorithm such as the Spectral Clustering following the same work flow . Trying to expand the data volume would also be very effective to better predict the lawn status of clients. It is also recommended to develop a classification predictive model to be able to predict customers preferences even though he is not present in our initial dataset.

Bibliography

[1] Chuang, S.-H., & Lin, H.-N. (2013b). The roles of infrastructure capability and customer orientation in enhancing customer-information quality in CRM systems: empirical evidence from Taiwan. International Journal of Information Management, 33(2), 271e281.

[2] Foss, B., & Stone, M. (2002). CRM in financial services: A practical guide to making customer relationship management work. Kogan Page Publishers.

[3] Thompson, E., & Sarner, A. (2009). Key issues for CRM strategy and implementations: Gartner research report series, Gartner research.

[4] https://www.kaggle.com/arjunbhasin2013/ccdata

[5] https://www.kaggle.com/safwanumer/credit-card-use-crm-system

[6] Tung-Shou Chen; Tzu-Hsin Tsai; Yi-Tzu Chen; Chin-Chiang Lin; Rong-Chang Chen; Shuan-Yow Li; Hsin-Yi Chen. A combined K-means and hierarchical clustering method for improving the clustering efficiency of microarray