Master thesis : Exploring Antitrust Cases with Clustering Methods

Auteur : Gilson, Maxence
Promoteur(s) : Ittoo, Ashwin
Faculté : Faculté des Sciences appliquées
Diplôme : Master : ingénieur civil en informatique, à finalité spécialisée en "management"
Année académique : 2021-2022
URI/URL : http://hdl.handle.net/2268.2/14582

UNIVERSITY OF LIEGE
FACULTY OF APPLIED SCIENCES

Exploring Antitrust Cases with Clustering Methods

This master thesis is submitted for obtaining the
Master's degree in Computer Science Engineering

Maxence Gilson - *S162425*

Academic supervisor
ITTOO Ashwin

Jury
GEURTS Pierre
ERNST Damien

ACADEMIC YEAR 2021-2022

Abstract

Exploring Antitrust Cases with Clustering Methods

Maxence Gilson

Faculty: Applied Science at the University of Liege
Section: Computer Science Engineering
Supervisor: Ashwin Ittoo
Academic year 2021-2022

The use of artificial intelligence in today's world has been growing for many years in many fields such as law. However, a part of law, called competition law or antitrust law, has been put aside. Hence, no machine learning or intelligent system helps antitrust law enforcers in their day-to-day work. This thesis seeks to fill this gap in order to determine whether it is possible not to automate the decision-making aspect of the antitrust judges' job. The goal is to know to what extent automation is possible thanks to AI applications in the antitrust field. Can a sentencing decision be taken by artificial intelligence systems? If not, is it possible to guide the judge by providing him patterns identified from older legal cases?

The first part of this thesis presents the various intersections between law and artificial intelligence. Then, there will be an explanation of what antitrust law is. This first part will end with the related works of the clustering methods used.

The second part is the one including the more technical aspects of the thesis. Firstly, the database and the different modifications made on it beforehand will be developed. Then, the different methods used to compute the performances of the algorithms will be presented. The different clustering algorithms will also be explained and analyzed. Moreover, several feature selection techniques will be developed and tested to determine the most relevant features. This part will conclude by determining that K-Means after the SPEC feature selection technique is the solution giving the best performances.

The last part presents a more legal analysis of the clusters formed using the most efficient methods for the available database. Indeed, the initial question which is to automate as much as possible the decision process of a judge, must be answered. In this part, the similarities of the legal cases within the same cluster will be put forward in order to prove that patterns exist and that the clustering method has allowed to determine them.

Résumé

Exploration des affaires dans le droit de la concurrence à l'aide de méthodes de classification

Maxence Gilson

Faculté: Sciences Appliquées à l'Université de Liège
Section: Ingénieur Informaticien
Promoteur: Ashwin Ittoo
Année académique 2021-2022

L'utilisation de l'intelligence artificielle dans le monde actuel se développe depuis de nombreuses années et ce dans de nombreux domaines tels que le droit. Cependant, une partie du droit, appelée droit de la concurrence, a été mise de côté. En effet, acutellement aucun apprentissage automatique ou système intelligent n'aide les responsables de l'application du droit de la concurrance dans leur travail quotidien. Cette thèse cherche à combler cette lacune afin de déterminer s'il est possible d'automatiser l'aspect décisionnel du travail des juges. L'objectif est de déterminer dans quelle mesure l'automatisation est faisable grâce aux applications de l'intelligence artificielle dans le domaine de l'antitrust. Une décision quant à la condamnation d'une personne ou d'une société peut-elle être prise par des systèmes d'intelligence artificielle ? Si non, est-il possible de guider le juge en lui fournissant des modèles identifiés à partir d'affaires juridiques plus anciennes ?

La première partie de cette thèse présente les différents croisements entre le droit et l'intelligence artificielle. Ensuite, il y aura une explication de ce qu'est le droit antitrust plus précisément. Cette première partie se cloturera par les travaux connexes des méthodes de classification utilisées.

La deuxième partie est celle qui comprend les aspects plus techniques de la thèse. Tout d'abord, la base de données et les différentes modifications qui y ont été apportées au préalable seront développées. Ensuite, les différentes méthodes utilisées pour calculer les performances des algorithmes seront présentées. Ces différents algorithmes seront également expliqués et analysés. De plus, plusieurs techniques de sélection de caractéristiques seront par la suite développées et testées afin de déterminer les caractéristiques les plus pertinentes. Cette partie se conclura en déterminant que l'algorithme K-Means après la technique de sélection de caractéristiques SPEC est la solution donnant les meilleures performances.

La dernière partie présente une analyse plus légale des groupes de cas formés en utilisant les méthodes les plus efficaces pour la base de données disponible. En effet, il s'agit de répondre à la question initiale qui est d'automatiser autant que possible le processus de décision d'un juge. Dans cette partie, les similarités des cas juridiques au sein d'un même groupe seront mises en avant afin de prouver que des modèles existent et que la méthode de classification a permis de les déterminer.

Acknowledgement

I would like to express my sincere gratitude to:

- My supervisor, Pr. Aschwin Ittoo for his support throughout this thesis. His availability during the semester was invaluable and I am very grateful for this. Moreover his enthusiasm for this subject allowed me to become passionate about this inter-disciplinary project. I would also like to thank him for his encouragement at the end of the redaction of the thesis which allowed me to finish in time.

- Dr. Giovanna Massaroto for her advice concerning the legal aspect of this thesis.

- My friends Julien Bolland and Myriam Menouer for reading this work and giving me feedbacks on the content and form of this thesis that punctuates my studies.

- My family and friends for the support they gave me during my studies.

Liège, June 8^{th}, 2022

Maxence Gilson

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and Motivations

For the past two decades, artificial intelligence has been growing steadily, impacting many fields in today's world. Indeed, AI is becoming an important tool in fields such as healthcare, information, finance and law. This thesis will focus on the usefulness of machine learning in law and more particularly in the field of antitrust law[1]. This field combined with AI has already been discussed in the literature (Nicolas Petit et al, 2017[44]) and came to the conclusion that AI and the field of competition law are complementary. Algorithms can therefore serve and help antitrust law enforcers. Hence, the aim of this thesis is to help decision making in antitrust law. This subject follows the reasoning of the publication of G. Massarotto and A. Ittoo (2021 [26]) which tries to answer as well as possible the questioning raised by Dr. Giovanna Massarotto [41] in her book *Antitrust Settlements : How a simple agreement can drive the economy* (Massarotto, 2019[40]).

From a practical point of view, it must be determined whether a certain method can help antitrust judges in decision making or even replace them in some cases if the algorithm works well and efficiently enough. The real question that is being answered in this work is : "To what extend can the automation of the decision-making can be implemented in today's antitrust judiciary field?" But why would the US government need to use artificial intelligence as they manage to deal with the antitrust cases for now? The answer is simple, the available data is growing exponentially which creates new opportunities for companies that cross the boundary of illegality in terms of competition law whether they are competitors or not. This leads to new market monopolies due to anti-competitive measures which the Federal Trade commission (FTC) is fighting against. One question that may be raised in relation to this new problem is "Can government agencies be equipped to handle these data flows and therefore the newly created antitrust cases in a more efficient way?". As Pr. Ittoo mentioned in his paper on this subject (Massarotto et al, 2021 [26]), even the Assistant Attorney General's speech was focused on the use of AI in the antitrust field [2]. This master thesis will try to answer the questions question by creating/adapting a machine learning algorithm to handle antitrust cases.

Then, the scientific aspect will be discussed. One needs to know how to approach this practical problem. Indeed, there is no structured database containing all antitrust cases that have taken place in the United States that has already been created. The information about those cases can be found on

[1] Also called "competition law", it is the field of law that promotes or seeks to maintain market competition by regulating anti-competitive conduct by companies[19]. This will be further-developed and explained in section 2.2

[2] Assistant Attorney General Makan Delrahim, Remarks at the Thirteenth Annual Conference on Innovation Economics, Aug. 27, 2020

different websites such as the FTC's website and on the United States department of justice's (DOJ) website. However, the needed data of the legal cases are usually saved in PDF or HTML files with different formats and layouts. Some of the PDF files are even documents that have been scanned. These reasons make data collection complicated and slow. All data on the different cases had thus to be curated by hand.

Furthermore, the number of antitrust cases is very limited if the analysis needs to make sense. In fact, taking cases like the one of standard oil in 1870, even though it is the most known one, is not very useful as society and customs have changed. Cases prior 2013 have not been taken into account because according to Dr. Giovanna Massarotto, almost all cases before this specific date are no longer considered as being relevant.

Finally, the database that has been created is not only small but is also not annotated. Therefore, two solutions have been identified: either the database have to be annotated by hand using the help of domain experts (e.g. antitrust lawyers), either the AI algorithm determines on itself the different labels and the underlying structures. However, annotation by hand, even with the help of professionals, is not the most sensible way. Indeed, even when neglecting the time-consuming point of view, the number of clusters desired is not known in advance. Therefore the chosen algorithms will be trusted to make these choices (i.e. choose the correct number of clusters). Therefore, in this master thesis, unsupervised methods, in particular, will be used.

Due to the lack of annotations (i.e. labels in the database), it is not possible to use supervised learning and it is not possible or rather not recommended to use a deep neural network due to the small size of the database. The solution to this issue will be exposed in section 1.2.

1.2 Objectives

Regarding the problems mentioned above, the subject of this thesis had to be deepened using unsupervised machine learning and focusing on different clustering methods such as K-Means, Bisecting K-Means, K-Modes and Self-Organizing maps (SOM).

This thesis will be articulated around four main points. Firstly, the clustering methods will be chosen, i.e. which ones are the most adapted to the database. Indeed, the efficiency of a clustering algorithm depends on the provided database. An algorithm may be efficient with a certain database but rather bad on another one. In addition to choosing the most efficient clustering method, the best number of clusters must be determined. This will be done using different metrics which will be explained in section 3.3. Secondly, a search through the features will be done to determine which features are the most promising and useful ones for the clustering methods. This part is important as well, as determining which features are more relevant than others is determining in decision making from an antitrust perspective. Thirdly, we will verify that an increase in performances of the algorithms occurs when using these specific features. Finally, we will analyse the created clusters from an antitrust point of view.

This thesis tries to answer the question of whether it is feasible or not to create automation through unsupervised clustering techniques in the decision-making aspect of antitrust judges' job.

1.3 Contribution

Our main contributions are that we managed to create a novel structured database composed of all antitrust cases regulated by the Federal Trade Commission in the USA. This was not available yet as all data was scattered in different files and in different formats. Moreover, we evaluated several clustering methods including K-Means, Bisecting K-Means, K-Modes and Self-Organizing maps to find the most efficient algorithm for the restricted database. This evaluation was done using different metrics which are the silhouette score, Calinski-Harabasz index, Davies-Bouldin index and Hamming score. This investigation has never been done before for antitrust cases in the USA to our knowledge. In addition, we identified the most important features to focus on when analysing an antitrust law case.

The results show that K-Means clustering algorithm after the SPEC feature selection gives the best performances among all the tested unsupervised clustering methods. Moreover, it shows that the basic neural network used, which is self-organizing maps, yields highly variable results. This analysis provides as general conclusion that AI can help to find patterns within datasets made up of antitrust cases even if the database was restricted in this case. Thus, it could help antitrust judges in their decision-making although total automation is not yet on the agenda.

All results, values of the different metrics and plots were obtained by computing the various python files available https://github.com/maxencegilson/Master_thesis_.git.

1.4 Content

This thesis will be structured in the following way :

- Chapter 2 gives some background information on AI in law and why it should be used the antitrust law field. It is composed of the machine learning techniques' related works as well.
- Chapter 3 presents the database, the metrics and the different clustering methods used and their results. Feature selection is also developed in this section.
- Chapter 4 describes the analysis of the database from an antitrust point of view.
- Chapter 5 ends this master thesis by presenting the final results. The possible improvements and a personal opinion is also developed in this section.

Chapter 2

Background & Related Works

This chapter starts with AI in Law and with AI and antitrust in order to provide the necessary background to the readers. Then, the related works of the clustering methods, which are those are used for the analysis part in chapter 3, will be given.

2.1 AI in Law

In the light of the different innovations and the fast-moving technological world we live in, the way law and legal cases are handled should be updated too. Indeed, in the frame of this thesis, we have to determine whether or not there is a way to detect underlying patterns from the past antitrust cases [26]. Then, it is important to know if AI and law has already been merged successfully or not. If not, it is important to determine the underlying issue(s).

In fact, AI and law have already been used together to determine whether an offender is likely to re-offend. This was done using the COMPAS algorithm and was tested in Florida[1] to help the judge determining if the offender should remain in jail during his trial or if he/she could be released pending sentencing. However, the algorithm very quickly showed signs of bias(Mark Coeckelbergh, 2020 [18] and Adrienne Brackey, 2019 [14]). It judged people of color more harshly and was therefore considered racist. This sets a precedent between AI and U.S. law that undermines our attempt to assist judges in their decision-making. This is an example where AI has been used but where the outcome did not put the AI algorithm COMPAS in the foreground. However, other AI and law related examples exist. For example, an AI approach in employment law to determine whether a worker has an employee or an independent contractor status has been created by Maxime Cohen et al in 2022 (2022, [7]). This predictive model was so successful that an open-access platform has been created in the US and in Canada. The is as decision-making algorithm based on previous cases as this thesis tries to create and therefore shows that the aim of this work could be reachable.

Regarding the COMPAS example, a crucial element playing in this thesis' favor, is that the subject discussed in this paper is to give an opinion or determine a pattern on economic elements and not on a racial subject. The machine learning algorithm used in this paper will therefore only have as input facts and not features that could play in our disadvantage like skin color. Therefore, this work would be a collaboration between an AI algorithm and the USA law enforcers without the risk of violating human rights (Massarotto and Ittoo, 2021 [26]). Concerning the employment law example, it shows that AI and law already had a successful collaboration.

[1]The law and AI relation has been mainly looked up in the USA as this thesis concentrates on US antitrust cases

In addition, predictive tools have proven to save precious time to law enforcers in the past as well as reducing price of expensive court trials (2022, [37]). In fact, as clearly explained by Angele Christin (2015,[2]), predictive tools are used in many parts of the U.S. judicial system and since several years even though all examples are not AI algorithms. The first area where predictive algorithms are used is in pretrial and bail. Indeed, the Arnold Foundation [2] created in 2015 ([31]) a tool to determine whether a defendant could commit a crime or would come back to court for his trial or not. This tool has been used successfully in twenty-one jurisdictions in the United States where the crime rates and jail population decreased. Secondly, AI is already used for parole and probation. Indeed, as for COMPAS, the LSI-R[2] (Level of Service Inventory Revised)software has been created as predictive tool to compute the level of eligibility for parole. Indeed, this software can be used in correlation with a judge's opinion to support his decision-making. Moreover, the LSI-R predictive system has been trained through data from different sources (experts' opinions, recidivism literature and social learning) as this study aims to do it. Furthermore, in 2014 a risk assessment instrument for juvenile criminals has been implemented in thirty-nine states. This tool, created by the Annie E. Casey Foundation[3], aims to decide whether a juvenile should go in detention, in a alternative program or whether he/she could go home. Finally, there are even older links between predictive tools and law. Indeed, in 1984, sentencing tables were created. These tables are uniform policies for sentencing individuals and organizations[4]. Even though they were not created using modern AI technologies, they were built by gathering huge amount of data as AI algorithms work nowadays. This proves that for many years, judges have been relying on more than just their knowledge and have also been relying on predictive tools that combine the knowledge of many previous legal cases.

In the UK, another AI system has been implemented to focus on the automation of online proceedings for dispute prevention, resolution and control. Judges are now no longer necessary anymore for some specific online dispute resolution services (ODR) shown in figure 2.1[5]. Indeed, as explained by Jeremy Barnett et al (2018 [27]), three to four percent of cases can be resolved without a hearing and thus ODR allows for people waiting for hearing not to wait 59 weeks which is the average time for a court to take the case. This proves that some cases can be judged accordingly to some AI systems' decisions.

In addition, AI is already used in other countries' judicial courts. In Malaysia for example (Mahyuddin Daud, 2022[21]), artificial intelligence has been used in sentencing. AI has therefore been used in decision-making on how should accused people be sentenced. The used algorithm based itself on cases that were adjudicated between 2014 and 2019. The AI system was only used to give an advice on what should the sentence be but it would the judge who had the last call. The idea was to gain time as well as to be more consistent from one trial to another. However, although the judge sided with the AI by assigning a sentence of twelve months to one inmate (ten months were advised by the AI) and nine months to the second (as advised by the AI), the lawyers are challenging the sentence. This examples shows that AI systems can be used to assist judges in the decision-making aspect of their jobs.

Along with the previous examples, China also implements intelligent courts (Yadong Cui, 2020 [20]). Indeed, machine learning extracts information from huge past legal cases and predict results

[2] https://www.arnoldventures.org/
[3] https://www.aecf.org/
[4] https://en.wikipedia.org/wiki/United_States_Federal_Sentencing_Guidelines
[5] https://academic.oup.com/comjnl/article/61/3/399/4608879?login=false

	Negotiation		Mediation		Arbitration	
	Full-automated	Human-assisted	Full-automated	Human-assisted	Full-automated	Human-assisted
Consumer ODR						
EU ODR Platform	✓		✓	✓		✓
eBay Res Center		✓		✓		
Mondria	✓		✓			
SmartSettle	✓		✓			
Judicial ODR						
VirtualCourtHouse		✓	✓	✓		✓
CyberSettle	✓		✓			
Corporate ODR						
'future system'	✓		✓		✓	

Figure 2.1: Online dispute resolution services examples[5]

of the trial. These results are used by judges and thus help standardizing the judgement process as well as decreasing the number of cases where people are wrongly sentenced. This proves that artificial intelligence is used as a tool in the decision-making process of a judge. Nevertheless, the judgement is not entirely automated.

Moreover, unfortunately, artificial intelligence has already been used for anti-competitive purposes and therefore AI and antitrust has already been studied. Indeed, algorithmic price discrimination and tacit collusion have been discussed (Axel Gautier et al, 2020 [4]) as of how machine learning can be used as an harmful tool. However, in this work AI is being used in a totally opposite way, trying to give a legal advice on how to handle cases.

In light of the elements presented in the previous paragraphs and knowing that a collaboration between AI and antitrust law has been overlooked in the US[6], the aim of this thesis makes sense. Indeed, AI and law has already been discussed as mentioned in the previous paragraph however only in a harmful way. In this paper, the aim is to create a collaboration between AI algorithms and law in order to identify patterns between existing antitrust cases and thus help judges in their decision-making.

2.2 Antitrust

Before diving into the algorithmic part of this project, it is necessary to have some idea of what antitrust law is. The goal of this paper is to help antitrust agencies by automating some decision making tasks using artificial intelligence. Antitrust law, which is also called competition law, is composed of several federal laws that aims to:

- promote competition between different companies in the market
- prevent from the emergence of unjustified monopolistic markets
- try to reduce as much as possible collusive practices

Antitrust has mainly an economic aspect. Indeed, as Giovanna Massaroto describes it in her book (Massaroto, 2019 [40]), there are regularly financial agreements between the antitrust agencies and the

[6]Indeed, AI and antitrust law has never been used in collaboration in the US to our knowledge

targeted companies. This is a solution quite favorable to both parties as it avoids the state to pay for very expensive trials and avoids the companies an uncertain output concerning the trial as well as a risk to have a bad corporate branding.

As mentioned in chapter 1, there are mainly two large government agencies which are the DOJ (department of justice) and the FTC(federal trade commission), that handle these specific competition law cases. However, only the cases from the Federal Trade Commission were considered in this thesis. The reason for this is quite simple: as mentioned earlier, both the government and the companies involved would rather avoid criminal court. Moreover, the FTC is not required to go through the court system unlike the DOJ. The cases treated by the FTC are thus much more numerous and it is therefore these that are dealt with in this thesis.

Although the structure and usefulness of the features of our database will be discussed in section 3.2 and in section 3.5 as it seems important to explain what the different features refer to.

2.3 Clustering Methods

Several clustering techniques are used throughout this thesis. These are K-means, Bisecting K-means and K-Modes clustering. Self-Organizing maps are also prospected in order to see the efficiency of basic neural networks. The different clustering techniques are detailed and tested in chapter 3.

K-Means algorithms is predominantly applied to customer segmentation as it was the case in Tushar Kansal et al's paper (2019, [8]) as well as in Utsav Sharma et al's paper (2022 [10]). Moreover, the latter article, hierarchical clustering was used. Thus Bisecting K-Means, which is the collaboration of K-Means for 2 clusters and hierarchical clustering seems to be also a good solution. Indeed, Bisecting K-Means has been used as well in customer segmentation in articles written by Novianti Puspitasari et al (2020 [46]).

Regarding the K-Modes clustering method, it is still one of the state-of-the-art method for categorical variables. Indeed, it has been used recently by Tess Cersonsky et al (2022, [16]) in order to asses people to determine whether they could have a cognitive decline or not. Moreover, K-Modes has also been used to find co-workers having same competencies in Sweden (2022, [54]).

Moreover, those three methods were used in order to deepen the study of G. Massarotto and A. Ittoo (2021, [26]).

Self-organizing maps (SOM) where used in this thesis to try a quite different approach from the three others as it is a simple neural network approach. Moreover, a research has been initiated by an US government's agency to visualises and explores datasets using SOM (R. Ponmalai and C. Kamath, 2019 [47]). In this work, it will serve the same purpose.

Chapter 3

Methodology & Results

3.1 Introduction

This chapter will proceed as follows. Initially, the database will be described as well as the first modifications that were made. Then, all the different metrics that have been used to determine the performances of the different clustering algorithms will be explained and developed. Afterwards, each clustering method will be detailed and their performances will be analyzed to determine the most efficient. The way the model will be analysed will depend on the way the method works. Feature selection for the studied database will be the next step of the thesis. Furthermore, each feature selection method will be detailed and used to determine the best features. The outperforming clustering algorithm with the different feature selection methods will be computed in order to determine again which feature selection method(s) is (are) the best.

In this thesis, where the database is limited and therefore the computation time is very low, the analysis will only be based on the different metrics developed in section 3.3.

3.2 Data Exploration

3.2.1 Dataset creation

First, it is important to create a database with all the antitrust cases since 2013 as explained in section 1.1. Indeed, there is currently no structured repository including all antitrust cases and therefore one had to be created. Actually the data was available on different websites[1] but these data were scattered in an unstructured way on these websites in PDF or HTML formats. Moreover, antitrust being a specialized field, this dataset creation required the collaboration of domain experts (i.e. competition lawyers). For those reasons we had to crawl the different websites with an expert to get the needed data. The expert (Giovanna Massarotto[41]) provided guidance to determine what information was to be extracted or not. It should be noted that the focus of the cases used was made only on the website of the federal trade commission as it will be explained in section 2.2. The database [2] was finally built with the needed help of Giovanna Massarotto[41] and of the academic supervisor of this theis, Ashwin Ittoo. Indeed, it took many iterations and meetings to have the most optimal and complete database possible.

[1] https://dc.gov, https://www.justice.gov, https://www.ftc.gov and https://law.justia.com

[2] https://docs.google.com/spreadsheets/d/14nXr8Cm1psosEnEAYy-I62Oje0WGYrVj/edit?usp=sharing&ouid=103570306248967671612&rtpof=true&sd=true

A structured database with 91 legal cases has been created where each case is composed of 32 variables. These 32 features are composed of the name of the company's industry, the 17 conducts for which a company could be judged and the 14 remedies deployed to address the problem.

3.2.2 Database

A simplified version of the database is provided in table 3.1 in order to visualise it. A case is composed of thirty-two types of features: the name of the industry in which the company operates, the 17 types of conducts and the 14 types of remedies. Every feature is described below:

Case number	Industry	Conduct A	...	Conduct Q	Remedy 1	...	Remedy 14
0	Healthcare	0	...	0	0	...	0
1	Computer	1	...	0	0	...	0
...	...	...	...	...	...	...	...
90	Healthcare	0	...	0	0	...	0

Table 3.1: Simplified Database

Let's start by the different types of conduct :

- Conduct A : Exclusionary conduct. "Exclusion involves a firm (or group of firms) raising the costs or reducing the revenues of competitors in order to induce the competitors to raise their prices, reduce output, or exit from the market.[3]" .
- Conduct B : Predatory conduct. It can be defined as dropping a company's prices in order to become dominant in the market and then increase its prices back up[4].
- Conduct C : Refusal to deal. This can seem odd as usually a company can choose with whom it wants to deal its contracts. However in some situation the federal court can judge that this refusal to deal strengthens its dominant position in the market[4].
- Conduct D : Tying conduct. This conduct occurs when two products are tied together. Even if the price will seem reduced consumers will need to buy both products in order to have the wanted one. If this technique is used to increase the sales of the tied product in a competitive purpose to gain a dominant position (when the company already has enough market power), it violates antitrust laws[4].
- Conduct E : Price fixing. It is when a company reach an agreement with (a) competitor(s) in order to decide together the price of products. This is considered illegal because each company must decide, independently from the others, the price of its products[4].
- Conduct F : Rebates. They can, when used by a company having a dominant market position, harm customers by reducing the ability of rival companies to compete [5].
- Conduct G : Discriminatory practices. This conduct happens when different customers pay different amounts of money for the same good for no cost related issues[5].

[3] https://scholarship.law.georgetown.edu/cgi/viewcontent.cgi?article=1197&context=facpub&httpsredir=1&referer=

[4] https://www.ftc.gov

[5] https://www.concurrences.com/

- Conduct H : Customer allocation agreement. This is when different competitors agree on how the sales territories are attributed or on how the consumers can be assigned to one company or another[4].

- Conduct I : Pay for delay. This occurs when a supplier pays the buyer. In return, the buyer will not sell competing products. This kind of conduct occurs often in the pharmaceutical industry[4].

- Conduct J : Disruption in the bidding process. Also called bid rigging, it is an anti-competitive behaviour where competitors agree on an auction that will take place. This happens regularly in auctions to win public contracts where companies will accept not to compete on a certain contract in order to have the next one for example[5].

- Conduct K : Agreement orchestration. Any kind of tacit agreements orchestration between competitors could harm customers[6]. This focuses especially on price-fixing orchestrations.

- Conduct L : Invitation ton collude. This occurs when a competitor makes a unilateral proposal to another competitor in order to coordinate on a competition term (e.g.the price). [7]

- Conduct M : Agreement not to compete. This conduct is self-explaining. One specific case of agreement not to compete is "Customer allocation agreement" where a company A accept not to compete in a specific geographic region B' that is B's company market. In return, company B won't compete in A's market, which is geographic region A'[4].

- Conduct N : Unlawful exchange of information. This occurs when "undertakings reciprocally provide or receive fact reports or details about business valuable information[5]". It is illegal when used in an anti-competitive purpose.

- Conduct O : Concerted practices. This occurs when competitors coordinates undertakings without any formal agreement[5].

- Conduct P : Conspiracy. It is an agreement between two or more people for anti-competitive purposes and therefore to acquire a certain monopoly at some point in the future[4].

- Conduct Q : No poach. This occurs when agreements are made in order for a company A to be sure that a company B won't solicit his employees. It can also be an agreement in order to fix the wages or to fix the terms of employment. All of this would be done without telling the concerned employees[8].

The remedies are the consensus between the government and the company on what the company is willing to do to rectify the current problem. These need less explanation as many of them speak for themselves.

- Remedy 1 : Amendments of contract provision. It is the fact of returning to the original contract for the different parties.

- Remedy 2 : Amendments of the code of ethics/code of conduct/association's rules. It is the fact of returning to the initial code of ethics/code of conduct/association's rules.

[6] https://globalcompetitionreview.com/guide/e-commerce-competition-enforcement-guide/e-commerce-competition-enforcement-guide/article/united-states-e-commerce-big-data-and-algorithms-antitrust

[7] https://www.vbb.com/media/Insights/%27Invitations_To_Collude%27_Targeted_By_US_And_EU_Enforcement.PDF

[8] https://www.linklaters.com/fr-be/insights/publications/2018/september/no-poach-agreements-whats-the-big-deal

- Remedy 3 : Obligation to disclosure information. This occurs when information has been acquired through anti-competitive purpose and means that the information has to be revealed.
- Remedy 4 : Limitation to enter into specific markets
- Remedy 5 : Refrained from the investigated conduct. This is withdrawing from the actual conduct that is being investigated.
- Remedy 6 : Compliance obligations
- Remedy 7 : Implementation of an antitrust compliance program
- Remedy 8 : Contract limitations
- Remedy 9 : Divestiture. It is the disposal of activities in the market that raises antitrust issues.
- Remedy 10 : Impose specific contract requirements
- Remedy 11 : Limitation in the exchange of information
- Remedy 12 : Permanent injunction. This is a permanent interdiction on a specific matter.
- Remedy 13 : Other performances obligations

3.2.3 Preprocessing

Now that a database has been created, preprocessing is needed. At first, the dataset had to be uploaded using the Pandas library to be able to read the CSV file containing the database. Then, the NaN values were cleared to have an easily usable database.

Afterwards, the categorical values had to be tackled as those kind of variables are often an issue for machine learning algorithms. Having categorical values is a problem because the vast majority of clustering algorithms need to compute the distance between different data objects. However, it is not possible to calculate a distance mathematically using the Euclidean distance for example, between categorical values. Therefore, "one hot encoding" (available with Pandas library [38]) was used to get rid of the categorical values. This method is used to create new features using binary values. An illustration of the database modified using one hot encoding is provided in table 3.3. Looking at the example, a new feature has been created for each different name of industry. In this new feature's column, each law case with this particular name of industry (in the database without one hot encoding) will have a value of 1. Whereas if this law case does not have this industry's name, it will have a value of 0. Note that after this one hot encoding, the database's frame is 91×41 as the "Industry" column is composed of 10 different industry types are in table 3.2.

Communications/Media	Professional/Trade association others	Computer industry
Healthcare/Pharmaceutical	Professional/Trade association healthcare	Gas & Oil
Real Estate industry	Professional/Trade association real estate	Transportation industry
	others	

Table 3.2: Name of industries

Case number	Industry
0	Computer
1	Healthcare
...	...
90	Healthcare

Case number	Computer Ind.	Healthcare Ind.
0	0	1
1	1	0
...	...	...
90	0	1

Table 3.3: Simplified Database with one hot encoding

However during this thesis, a database with these categorical values had also to be kept in order to use the main asset of the K-Modes clustering method. Indeed, this method is able to handle categorical and numerical values but this will be explained in more detail in section 3.4.3.

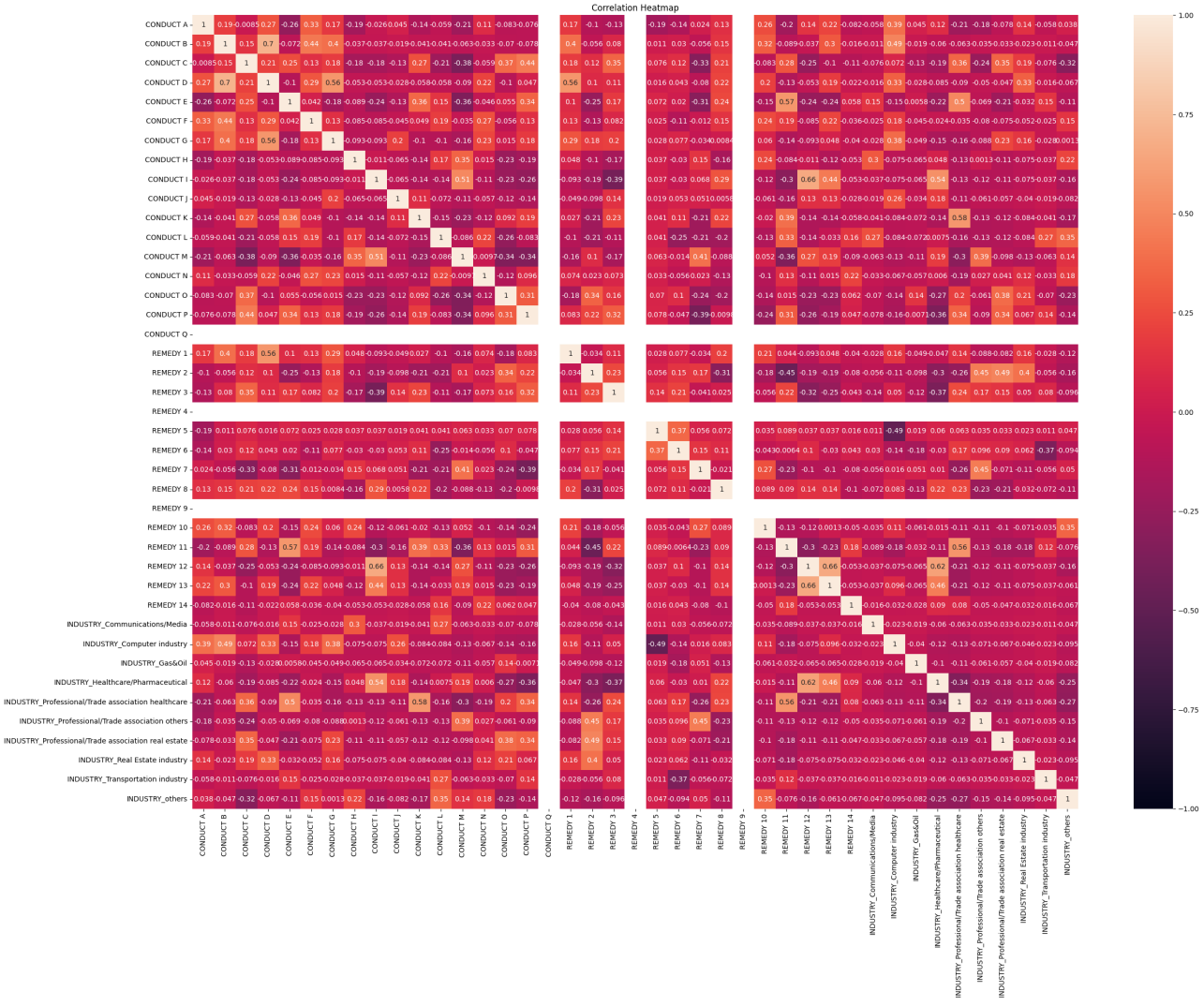


Figure 3.1: Correlation Heatmap with one hot encoded features

Finally, before starting the clustering, the correlation heatmap [29] (using the seaborn library [57]) between all features has been computed. This can be useful as if, between two features, the :

- correlation = 1 (perfect positive) : this means that the correlated features will occur at the same time.
- correlation = 0 : this means that there is no correlation between the two features. Therefore, knowing one doesn't help in order to determine the other.

- correlation = -1 (perfect negative) : this means that the perfectly negative correlated features will not occur at the same time.

The correlation heatmap representation of the database can be seen in figure 3.1. It is quite easy to observe that three variables are problematic : "CONDUCT Q", "REMEDY 4" and "REMEDY 9". In fact, the three features are null all the time and therefore have a correlation score of 0 with all the other features. From a practical point of view, this means that this particular conduct has not been crossed yet since 2013 and that the two remedies have not been used either. The three features do not allow us to induce anything towards the other features and are therefore useless for the clustering. Hence, those features won't be used in the rest of this master thesis. It can be observed in figure 3.2 that getting rid of those three particular features allows to obtain a heatmap without problems and especially allows to make a beginning of feature selection without too many efforts. The final database is thus composed of 91 law cases with 38 features each. The thirty-eight features corresponds to the thirty-two from the initial database and nine from the one hot encoding (ten industry names minus one column which contained the old industry column) from which the three columns from the correlation analysis a retrieved.

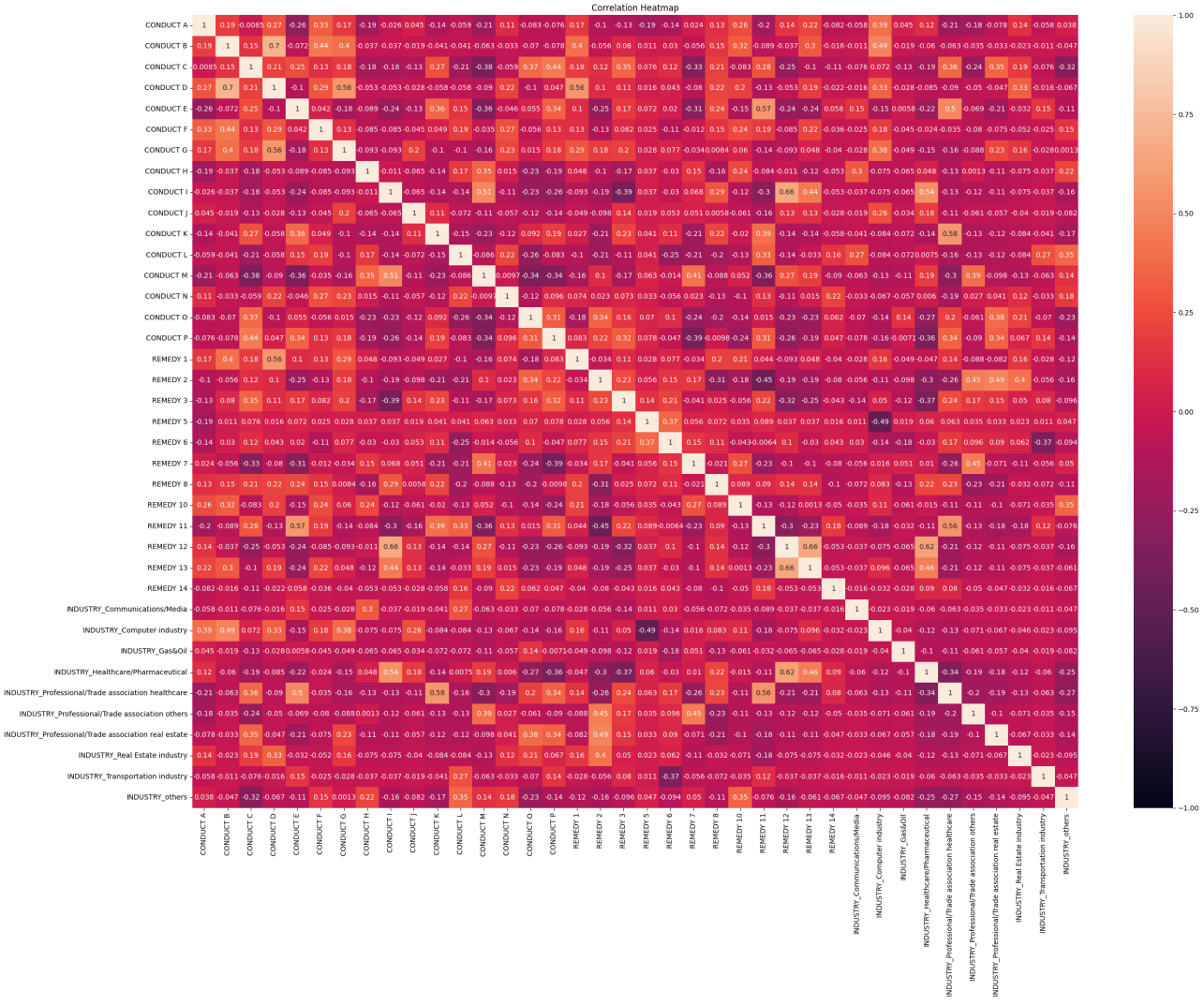


Figure 3.2: Correlation Heatmap with one hot encoded features without "CONDUCT Q", "REMEDY 4" and "REMEDY 9" features

3.3 Metrics

This section focuses on how the performances of the clustering methods have been determined. It is obvious that the most efficient way to determine whether certain clusters make more sense than others is to ask for the advice of an expert in the field, in the case an antitrust expert. However, this would require considerable financial and time resources. An automated approach is therefore more suitable. This approach is evaluated through five different scoring methods. Those are used to describe and determine the performances of the different clustering techniques, which will be explained in section 3.4. Furthermore, the best number of clusters depending on the method has also to be chosen. Due to the unsupervised nature of the database, a lot of scoring metrics had to be discarded. Thus, the analysis was focused on the five metrics discussed in this section (metrics from sklearn library[6]).

3.3.1 Silhouette score

The silhouette score [42] is the mean of the silhouette coefficients. This silhouette coefficient is, for every single data object, the difference between the average intra-cluster distance and the average inter-cluster distance. The intra-cluster distance a for a object i is the mean distance between this data object i and the other objects of the cluster. *a(i)* is thus defined as :

$$a(i) = \frac{1}{n_i - 1} \sum_{j \in C_i, j \neq i} d\,(i, j)$$

where n_i is the number of objects belonging to cluster C_i, where j is a different object from i, where C_i is the cluster to which both objects i and j belong and where $d(i, j)$ is the computed distance between the two data objects. The distance is often the Euclidean distance (formula available in section 3.4.1) or the Hamming distance for this study. If the value of a is large, it means that the cluster is not well clustered, whereas if it is small, the cluster is well grouped.

For the inter-cluster distance *b(i)*, it is defined as :

$$b(i) = \min_{C_k \neq C_i} \frac{1}{n_k} \sum_{j \in C_k} d\,(i, j)$$

where *min* means that the cluster C_k closest to the cluster C_i is chosen, where n_K is the number of data objects of cluster C_k and where $d(i, j)$ is the computed distance between two data objects. If the value of b is small, the two cluster i and k may overlap and if its value is high, both clusters are well separated.

The silhouette coefficient of a single data point i is defined as :

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1$$

The silhouette score[22] of the entire dataset for K clusters as the average silhouette score of each cluster is expressed as:

$$S = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n_i} \sum_{i \in C_i} s\,(i)$$

The higher this metric is, the best clustered the dataset is. On the contrary, if the obtained value of the silhouette score is very small, it means that there certainly are overlaps between the created clusters.

3.3.2 Davies-Bouldin Index

The Davies-Bouldin score/index[22] is the mean similarity of the different clusters with their most similar cluster. Therefore the lower the index is (i.e. close to zero), the more dissimilar the clusters are. It would thus mean that the clustering algorithm performs well. From a mathematical point of view [22], one need to compute the average distance/similarity μ_k between all points of cluster C_k and the cluster's barycenter G_k noted μ_k :

$$\mu_k = \frac{1}{n_k} \sum_{i \in C_k} ||P_i - G_k||$$

where P_i is one data object i of cluster C_k.

One also need to compute the distance between the barycenters G_k and $G_{k'}$ of clusters C_k and $C_{k'}$ denoted δ :

$$\delta_{k,k'} = ||G_{k'} - G_k||$$

Davies-Bouldin score can thus be denoted as :

$$DB = \frac{1}{K} \sum_{k=1}^{K} \max_{k \neq k'} \left(\frac{\mu_k + \mu_{k'}}{\delta_{k,k'}} \right)$$

where K is the number of clusters and where the *maximum* is used as one want to compare a cluster to its most similar cluster.

3.3.3 Calinski-Harabasz Index

This metric is simply explained as the ratio of the inter-cluster dispersion and of the intra-cluster dispersion. Unlike the Davies-Bouldin index, this metric must be as large as possible to have good performances for a clustering algorithm. Indeed, a small value for this index means that either the different clusters are not differentiable enough (a small numerator), or that in one cluster, the data are very scattered (a high denominator) or both propositions at the same time.

From a mathematical point of view [9], one first needs to express the intra- and inter-cluster values respectively W and B :

- $W = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \|x_i - z_k\|^2$
- $B = \sum_{k=1}^{K} n_k \|z_k - z\|^2$

where K is the number of clusters, where x_i is a data object of cluster C_i, where z is the centroid of the entire database and where z_k is the centroid of cluster C_k.

The Calinski-Harabasz Index can thus be defined as :

$$CH = \left(\frac{B}{K-1} \right) / \left(\frac{W}{n-K} \right)$$

where n is the entire number of data points.

3.3.4 Hamming variance score

This metric was implemented after a discussion with Pr. Geurts. Indeed, Mr. Geurts pointed out that the used database was only made of binary data and that calculating its hamming distance would therefore make sense. The hamming distance between two arrays of similar length is the sum of their dissimilar binary values. For example, if $x = 0001$ and $y = 0000$ are two arrays of binary values, the hamming distance between them would be equal to one. Indeed, only the last value of the arrays differ.

Thus an implementation of the intra- and inter-cluster hamming distance could be useful. Indeed, simply calculating the sum of the hamming distances between each data object did not make much sense. After looking in several papers [11][50] on the evaluation of clustering methods, the idea was to compute the variance of the hamming distances of data objects. For the intra-cluster distance, the hamming distance was computed between each pair of data objects of a same cluster and it was reiterated for every cluster and every obtained values were entered in a array. Concerning the inter-cluster distance, the distance between each pair of points that were not in the same cluster was computed. Having the intra- and inter-cluster distances, the intra- and inter-variances were computed.

Let $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_m)$ be respectively the different intra and inter-cluster hamming distances. Their variance are respectively :

$$V_X = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \text{ and } V_Y = \frac{1}{m}\sum_{i=1}^{m}(y_i - \bar{y})^2$$

where $\bar{x}$ and $\bar{y}$ are respectively the mean of the intra- and inter-cluster hamming distance arrays.

To remain consistent with previous methods, the hamming distance variance score is based on the way the Calinski-Harabasz Index is computed. There are some small differences, as the distance between each object is computed and not the distance between an object and the centroid. However, a high output is still desired in order to obtain good performances. The final formula is therefore :

$$HV = \frac{V_Y}{V_X}$$

3.4 Model Selection

First, it is important to note that all the following methods need a similar input and will all give a same output as they are all unsupervised clustering methods. All of them will all take as input a database (X) and the desired number of clusters (K) as output. The algorithm will thus provide K clusters from X. These different algorithms will try to respect two constraints. The first one is that the data points in the same cluster must be as similar as possible. The second constraint being that the data points of a specific cluster must be as different as possible from the data points of the others K-1 clusters.

3.4.1 K-Means

3.4.1.1 Introduction

The K-Means algorithm, although published in 1982, is a clustering method more than often used even today. It is still said to be state-of-the art.

This algorithm, as shown in algorithm 1, chooses k random data points of the database. The other data points are assigned to the closest centroid. A centroid is defined as being geometric center or arithmetic mean of the cluster and feature i of the centroid can be computed thanks to the following formula :

$$c_i = \frac{\sum_{j=0}^{n-1} x_{ji}}{n}$$

where n is the number of data objects and x_{ji} is the feature i of data object j.

In order to compute a distance between two data points, many measures can be used. However, in this case, the basic metric was used, which happens to be the Euclidean distance. The formula is as follows,

$$d(x, c) = \sqrt{(x_1 - c_1)^2 + ... + (x_m - c_m)^2}$$

where $x = (x_1, ..., x_m)$ is a data object, $c = (c_1, ..., c_m)$ is a centroid and m is the number of features.

Then, the mean of each cluster will be computed to design a new centroid. This is done by summing all data points' feature vectors belonging to the cluster and dividing it by the number of data points in this cluster. This will go over and over until no data point moves again from one cluster to another.

Algorithm 1 K-Means(X,K)[55]

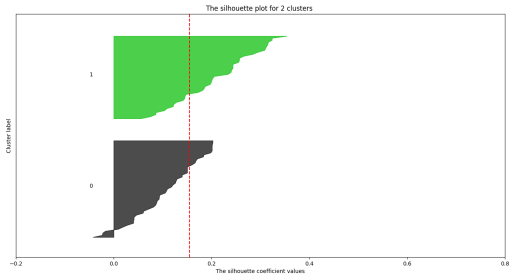
```
Initialize k data objects as centroids
while data objects still change from cluster or until the max number of iterations is not reached do
    for each data object x ∈X do
        Compute distance between x and all the centroids
        Assign x to its nearest centroid
    end for
    Compute the mean of each cluster which will become the new centroids
end while
Output Array with each data object labeled
```

3.4.1.2 Results

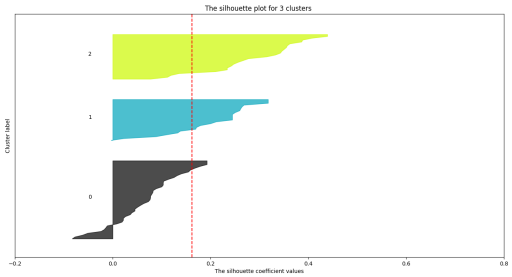
The different results for the K-Means clustering were computed thanks to scikit-learn library [6].

Let's start with the silhouette analysis for the K-Means clustering algorithm. This silhouette analysis has been done and plotted using the silhouette analysis technique presented on scikit-learn's website [52]. This analysis will show throughout the thesis that the silhouette cluster representation is never great which is certainly due to the small database.

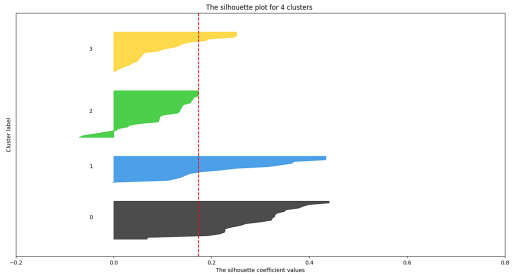
The aim of the silhouette representation is to have clusters with similar silhouette coefficients and with the clusters' size as equal as possible. The silhouette coefficient can be compared to the average silhouette score represented by the red line. The number of objects in a cluster is represented by the thickness of the cluster. Hence, a cluster three times thicker than another one has three times more data objects in it.



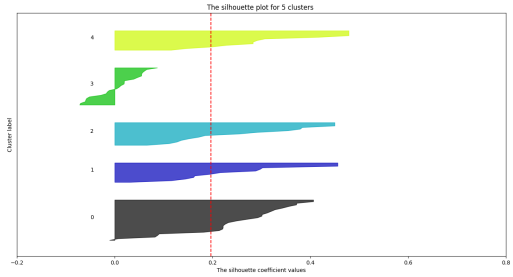
(a) The silhouette plot for the 2 clusters



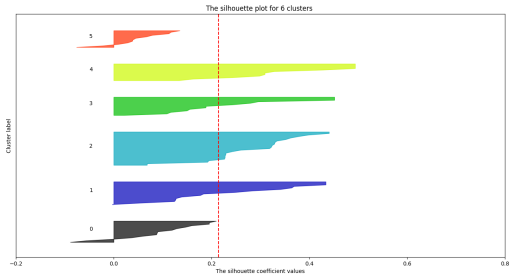
(b) The silhouette plot for 3 clusters



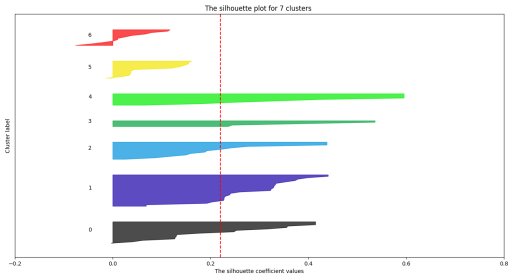
(c) The silhouette plot for 4 clusters



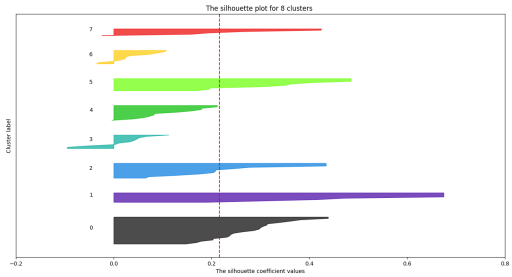
(d) The silhouette plot for 5 clusters



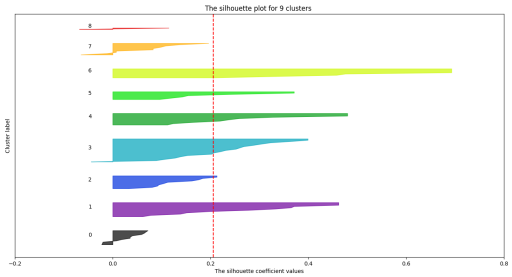
(e) The silhouette plot for 6 clusters



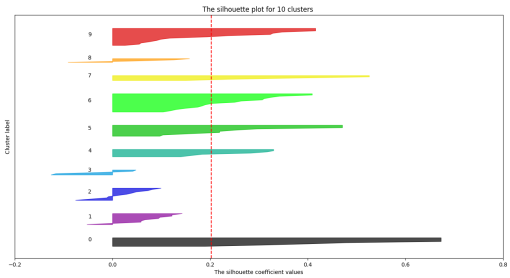
(f) The silhouette plot for 7 clusters



(g) The silhouette plot for 8 clusters



(h) The silhouette plot for 9 clusters

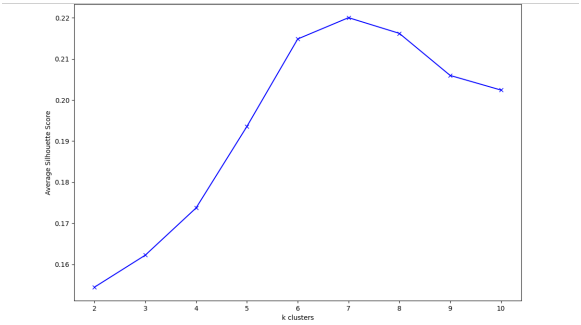


(i) The silhouette plot for 10 clusters

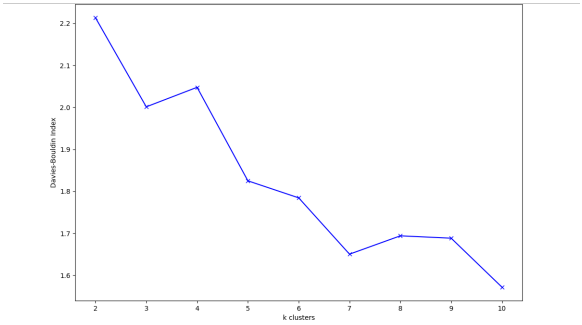
Figure 3.3: The silhouette plot of K-Means clustering method for the various clusters

By undertaking this analysis, some values can rather quickly be put aside for the number of clusters for this method. Indeed, figure 3.3h and figure 3.3i can be discarded as the clusters have different sizes and that they have many negative values for their silhouette score. Moreover, figure 3.3d shows a cluster three with a low value of its silhouette coefficient compared to the others. After the analysis of the silhouette representation, three different numbers of clusters can already be discarded. Furthermore, one can notice that the average silhouette score when there is two, three and four clusters is much lower than the highest average value. Indeed, for two clusters, the value is more than 30% lower. For three and four clusters, there are respectively a decreases of 26% and 21% as it can be computed from table 3.4. Moreover, using two or three clusters does not give enough information form a legal point of view.

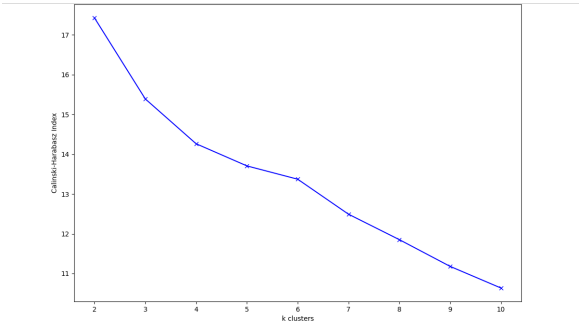
Now, let's move on to the analysis of the different elbow plots. The elbow method is an heuristic method which consists of plotting a graph of obtained values for a specific clustering method for different numbers of K clusters (two to ten clusters in this case). The aim is to choose the elbow of the curve as being the optimal number of clusters. The ideal is not to find the best value but to find a trade-off between the increase in performance of a metric and the cost that an additional cluster would take. Indeed, increasing the number of clusters too much does not give a much better apprehension of the initial dataset and would lead to over-fitting. On the contrary, not having enough clusters would lead to under-fitting.



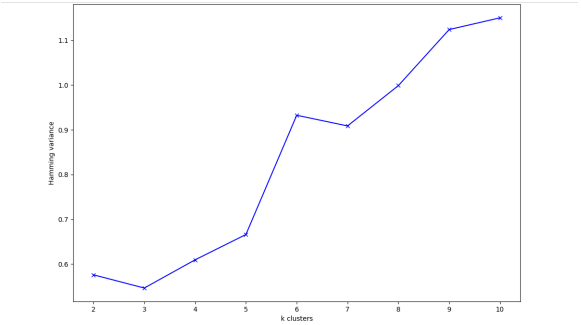
(a) Elbow plot using Silhouette score



(b) Elbow plot using Davies-Bouldin Index



(c) Elbow plot using Calinski-Harabasz Index



(d) Elbow plot using Hamming variance score

Figure 3.4: The Elbow Plots of K-Means

For this particular case with K-Means, figure 3.4a presents a "knee" or "elbow" to the curve for six clusters in a fairly intuitive way. The figure 3.4d shows a fairly large increase in performance for

six clusters as well. Concerning figure 3.4c, this plot is less expressive but a slight elbow to the curve can be identified for six clusters. If only the figure 3.4b was analyzed, seven clusters would probably be the best solution. However, this is not surprising given that this is the number of clusters with the best silhouette score as shown on figure 3.3g and in table 3.4.

3.4.1.3 Conclusion

After this analysis, the general opinion of the different graphs chooses K-Means with six clusters as the best solution. In fact, according to the elbow plot analysis, three out of four graphs would put this number of clusters forward. Moreover, according to the silhouette representation analysis, the solution with seven clusters (as put forward with figure 3.4b) has much more variations in their cluster sizes than with six clusters. Furthermore, the solution with K=7 has two clusters with much lower scores than the average silhouette score as opposed to the solution with six clusters which only has one.

The table 3.4 shows the chosen number of clusters in red and best performances in bold. By looking at table 3.4, choosing the best performance for each metric is not the most relevant. A reasonable trade-off is preferable to avoid over-fitting.

Metrics / # of clusters	Silhouette score	Davies-Bouldin Index	Calinski-Harabasz Index	Hamming Variance Score
2	0.154452	2.213624	**17.427595**	0.575848
3	0.162242	2.001298	15.390047	0.546311
4	0.173736	2.047795	14.262858	0.609325
5	0.193562	1.825063	13.705986	0.665921
6	0.214856	1.784464	13.371051	0.932545
7	**0.220040**	1.650407	12.491318	0.908713
8	0.216222	1.694081	11.852183	0.998828
9	0.205982	1.688460	11.179833	1.124170
10	0.202425	**1.571785**	10.636045	**1.150373**

Table 3.4: Metrics Values for K-Means

3.4.2 Bisecting K-Means

3.4.2.1 Introduction

The bisecting K-Means clustering method is an algorithm mixing top-down hierarchical clustering and K-Means clustering with two clusters. The aim is to start from one single cluster containing all data points and to, at every iteration, divide the cluster with the higher intra-cluster distance into two new clusters. This division is done in an iterative way until K clusters are obtained. This intra-cluster distance is computed as the SSE (sum squared error), denoted by :

$$SSE = \sum_{i=0}^{n} (x_i - \bar{x})^2$$

where x_i is a data object in the cluster, n is the number of data objects in the cluster and $\bar{x}$ is the average of the data objects in the cluster.

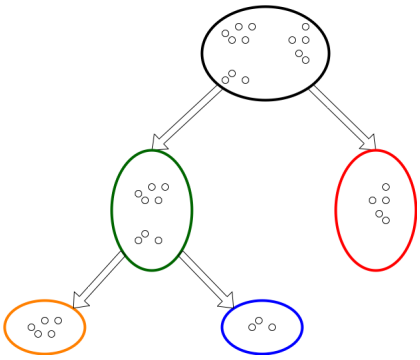


Figure 3.5: Bisecting K-Means representation[9]

In order to explain in depth how the bisecting K-Means algorithm 2 works, let's use the example shown in figure 3.5[9] where the wished number of clusters is K=3. In order not to make the following explanation unnecessarily long, the clusters will be designated by color on the diagram. As mentioned in the algorithm, the bisecting K-Means method starts by clustering all data points of the data set in one single cluster (the black one). Then, the K-Means algorithm comes into play to divide the cluster into two new distinct ones, the green cluster and the red cluster. Now, two clusters are available, which are not enough as the goal is obtaining three clusters. Therefore, the intra-cluster distance will be calculated for each cluster. As said before, the intra-cluster distance is defined as the sum of the squared errors (SSE). Then, after having computed the SSE, the cluster having the highest SSE (i.e. the one with the least clustered data points) is identified. Therefore, the green cluster is selected. Let's compute again K-Means on the green cluster in order to get two new clusters, the blue cluster and the orange cluster. The goal of getting three different clusters (clusters orange, blue and red) is reached which means that the bisecting K-Means algorithm is finished.

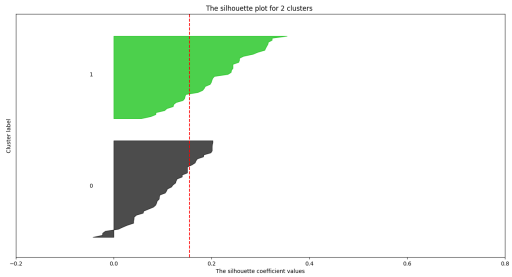
Algorithm 2 Bisecting K-Means(X,K)[53]

```
Initialize a cluster containing all data objects
while until K clusters are created do
    if Actual number of clusters = 1 then
        Use K-Means for 2 clusters on the cluster
    else if Actual number of clusters != 1 then
        Compute the SSE for each cluster
        Select the cluster with the highest SSE          ▷ SSE is the sum squared error
        Use K-Means for 2 clusters on this cluster
    end if
end while
Output Array with each data object labeled
```

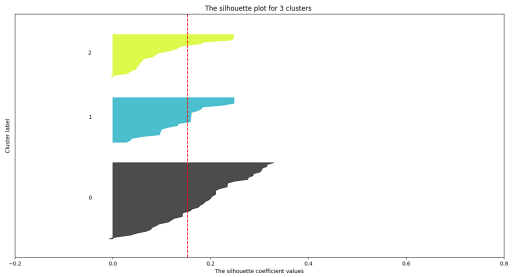
3.4.2.2 Results

The different results for the Bisecting K-Means clustering were computed thanks to scikit-learn library [6].

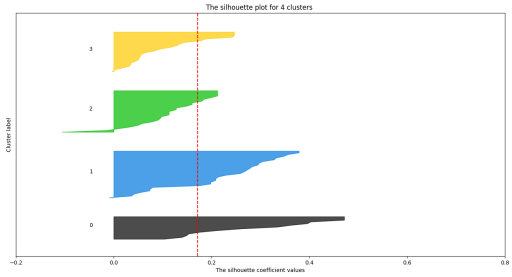
[9]This representation of Bisecting K-Means has been inspired from https://towardsdatascience.com/bisecting-k-means-algorithm-clustering-in-machine-learning-1bd32be71c1c



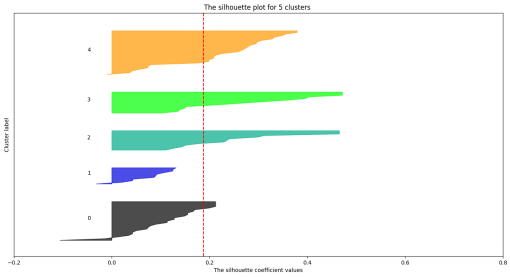
(a) The silhouette plot for the 2 clusters



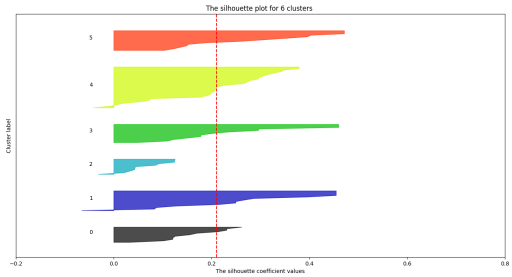
(b) The silhouette plot for 3 clusters



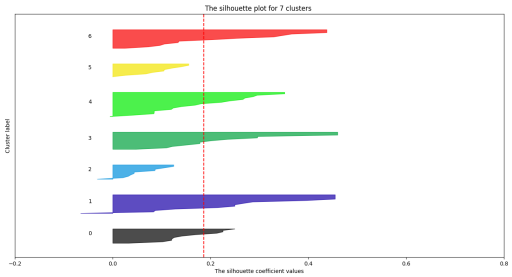
(c) The silhouette plot for 4 clusters



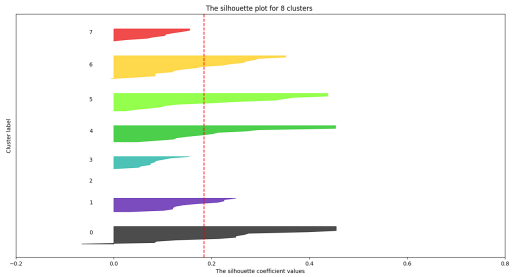
(d) The silhouette plot for 5 clusters



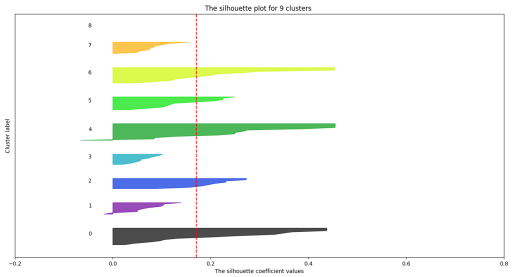
(e) The silhouette plot for 6 clusters



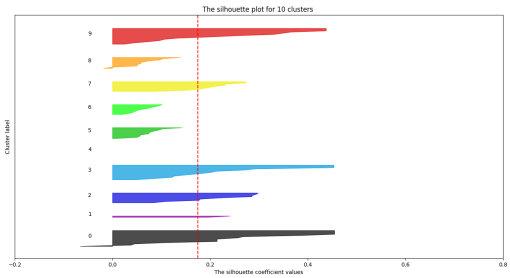
(f) The silhouette plot for 7 clusters



(g) The silhouette plot for 8 clusters



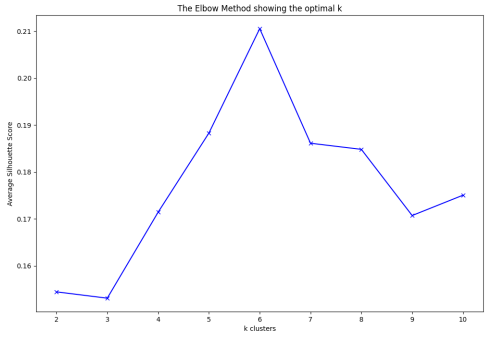
(h) The silhouette plot for 9 clusters



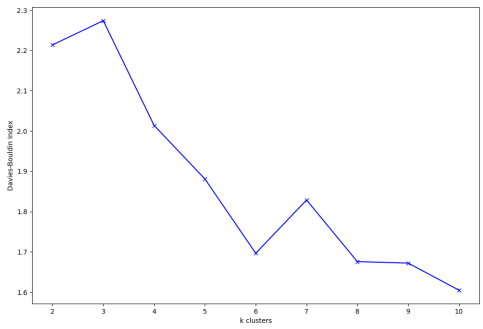
(i) The silhouette plot for 10 clusters

Figure 3.6: The silhouette plot of Bisecting K-Means clustering method for the various clusters

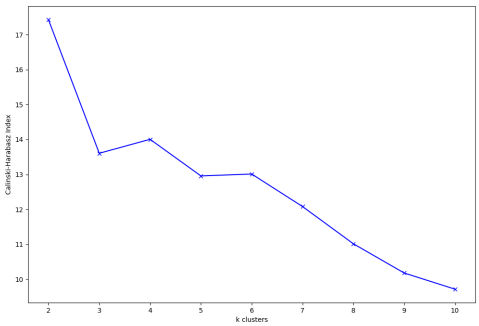
Let's now analyze the results obtained with the K-Means bisecting algorithm with different numbers of clusters using the silhouette representation. It can be observed on the figure 3.6g, figure 3.6h and figure 3.6i that, respectively, the clusters with the number 2, 8 and 4 are empty. These cases can thus be put aside as the algorithm did not give the correct output. Concerning figure 3.6g and figure 3.6h, that problem aside, the other clusters are equivalent and even highly symmetrical. For figure 3.6i, the missing cluster is not the only problem as there are also big differences in size between the different clusters. Indeed, cluster one is almost seven times larger than cluster one. Next, figure 3.6b presents a factor two between two different clusters. This is quite problematic as the number of clusters is quite low and there is a risk of under-fitting. This under-fitting problem is likely to occur if figure 3.6a is chosen. Indeed, with only two clusters, it might not give enough information to a judge if he/she wants to use the clustering algorithm for the decision-making part of his work. Moreover, figure 3.6d shows clusters with wide variations in their plot's sizes as well, which is not ideal as mentioned earlier. In a nutshell, this analysis considers that four clusters is the best solution as all values of the silhouette coefficient are higher than the average silhouette score. However, six and seven are still valid as number of clusters.



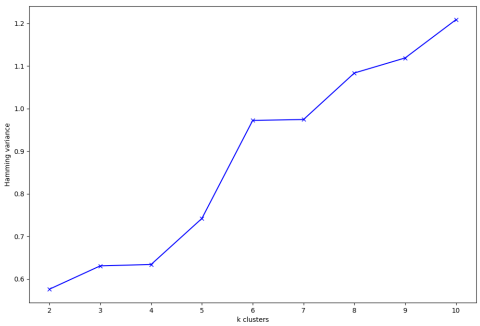
(a) Elbow plot using Silhouette score



(b) Elbow plot using Davies-Bouldin Index



(c) Elbow plot using Calinski-Harabasz Index



(d) Elbow plot using Hamming variance score

Figure 3.7: The Elbow Plots of Bisecting K-Means method

Let's dive into the analysis of the elbow plot in order to have some certainty about the ideal number of clusters. The figure 3.7a, figure 3.7b and figure 3.7d show that the number of clusters K=6 is the best trade-off in terms of performance. Indeed, figure 3.7a shows a peak in K=6 and table 3.5 indicates that the silhouette score is ten percent higher for K=6 than for all the other values of K. Figure 3.7b shows a rather marked elbow shape in K=6 too. Figure 3.7d shows an elbow shaped plot for K=6 but K=7 could be a solution as well as they almost have the exact same score. Figure 3.7c indicates that K=6 is a good solution because it precedes a fairly severe decrease. Knowing that you are ahead

of a drop is favorable to good performances as the Calinski-Harabasz index must be as high as possible.

3.4.2.3 Conclusion

For this clustering method, all metrics agree and go in the same direction to indicate that the ideal number of cluster is K=6. Although the silhouette representation (figure 3.6) does not determine K=6 as the best, it is still the number of clusters for which the average silhouette score is the highest. The elbow plot analysis also determines that K=6 is the best solution for each even though K=5 was a good solution for figure 3.7c and K=7 was also a good solution for figure 3.7d. Moreover, K=4 has never been found to be a satisfying solution regarding elbows plots.

In this case, unlike the previous method and as far as the silhouette score is concerned, the graphical analysis and the value of the average silhouette score agree. Indeed, it is for K=6 that the highest value of the silhouette score occurs.

Metrics / # of clusters	Silhouette score	Davies-Bouldin Index	Calinski-Harabasz Index	Hamming Variance Score
2	0.154452	2.213624	**17.427595**	0.575848
3	0.153107	2.273729	13.605782	0.630642
4	0.171411	2.013843	14.004665	0.633811
5	0.188285	1.880885	12.957827	0.741963
6	**0.210528**	1.696555	13.012042	0.972333
7	0.186118	1.828724	12.081386	0.974422
8	0.184816	1.675803	11.016905	1.083660
9	0.170716	1.672129	10.180744	1.119070
10	0.175047	**1.604943**	9.717669	**1.208959**

Table 3.5: Metrics Values for Bisecting K-Means

3.4.3 K-Modes

3.4.3.1 Introduction

K-Modes [3] is a clustering algorithm allowing to take into account numerical and categorical variables. In this master thesis, it will be the only method allowing this that will be investigated. Indeed, only one method that can take into account categorical and numerical variables seemed sufficient since, out of the three dozen features studied, only one is categorical. Moreover, this feature can be easily transformed using one-hot encoding as mentioned in table 3.3. It is important to note that K-Modes inventors mention K-Means' algorithm as an inspiration and thus, K-Modes works in a similar way.

K-Modes clustering method will start on the same way as K-Means and not as bisecting K-Means. Indeed, it is not a top-down hierarchical clustering method as bisecting K-Means and therefore starts directly by specifying K the number of desired clusters. In the beginning, K leaders will be randomly designed. There are no longer centroids as in K-Means because now there is categorical values in the database as well as the numerical ones. Then, it is necessary to determine how many dissimilarities there are between each data object. The dissimilarity means the number of mismatches that occur. Each data object will then be assigned to the leader with which it has the least mismatches (i.e. the

one to which it is the most similar). The leader and the data objects that will be attributed to him are considered as a single cluster. Then, as in K-Means where the mean of each cluster is computed, here for K-Modes, it is the new mode that is computed. This mode is determined, for each cluster, by looking at which features are the most recurrent. The algorithm will stop running when no more data objects change from one cluster to another, as for K-Means algorithm.

The only thing left to explain is how this dissimilarity is computed. This is done in the paper [28] explaining in detail K-Modes. Let X and Y be two data objects with F features and $d(X,Y)$ being the dissimilarity measure between the two objects :

$$d(X,Y) = \sum_{i=1}^{F} \delta(x_i, y_i) \tag{3.1}$$

$$\text{where } \delta(x_i, y_i) = \begin{cases} 0 & \text{if } (x_i = y_i) \\ 1 & \text{if } (x_i \neq y_i) \end{cases}$$

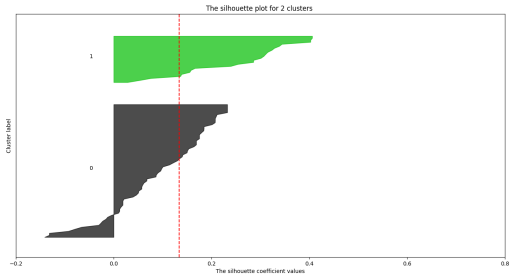
Algorithm 3 K-Modes(X,K)[28]

1: Initialize k data objects as leaders/modes
2: **while** data objects still change from cluster or max number of iterations not reached **do**
3: **for** each data object **x** $\in X$ **do**
4: Compute dissimilarity measure between **x** and all the leaders/modes ▷ The dissimilarity measure is computed like in equation (3.1)
5: Assign **x** to its nearest leader/mode
6: **end for**
7: Compute the new modes for each cluster
8: **end while**
9: **Output** Array with each data object labeled

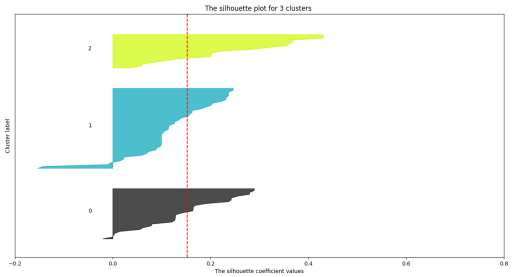
3.4.3.2 Results

The different results for the K-Modes clustering were computed thanks to k-modes library [56].

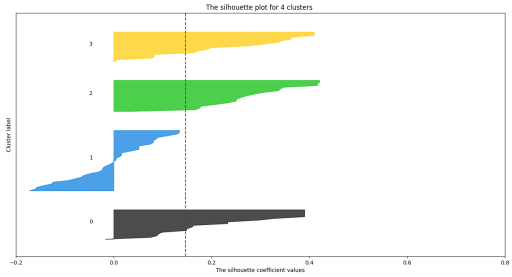
As for the two previous methods, the analysis starts with the silhouette representation plots for each value of K. First, for figure 3.8a, the difference in size is far too important. Indeed, in addition to the fact that there is only two clusters which often advocates under-fitting, there is a cluster three times bigger than the other one. This is a problem that pops out with figure 3.8b and figure 3.8c. However, in addition to having strongly different sizes, the second cluster for figure 3.8b has strongly negative silhouette scores. For figure 3.8c, cluster one is almost twice as large as all the others and has more than half negative values for its silhouette coefficients. It is for figure 3.8g, figure 3.8h and figure 3.8i that there is noticeably important variations for the cluster sizes as well. Indeed, for figure 3.8i, there is a factor seven between some clusters. Moreover, for figure 3.8g, from all the different plots, there is the most negative values for the silhouette scores. Then, for figure 3.8f, there is a cluster with almost only negative values for the silhouette score in addition to having a cluster almost three times smaller than the largest cluster. This size difference is even less important with seven clusters than with two, as in figure 3.8a.



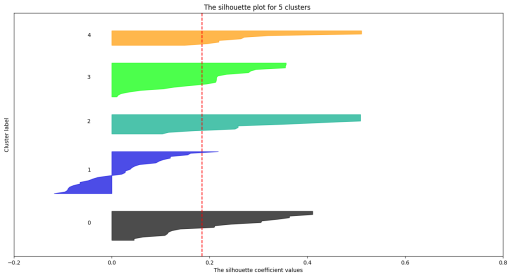
(a) The silhouette plot for the 2 clusters



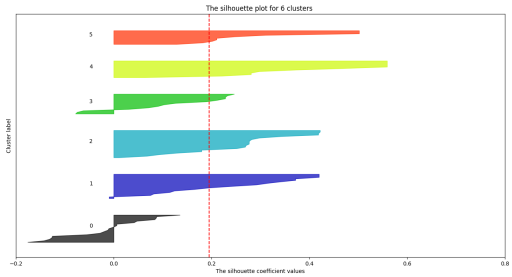
(b) The silhouette plot for 3 clusters



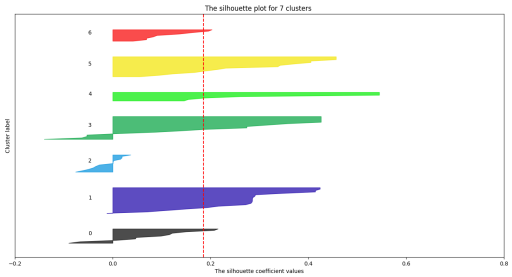
(c) The silhouette plot for 4 clusters



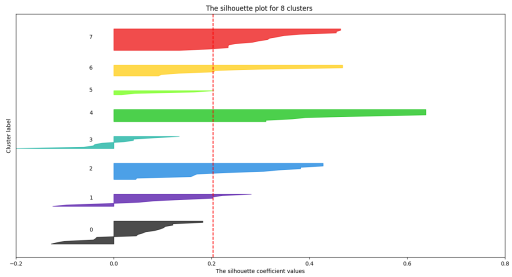
(d) The silhouette plot for 5 clusters



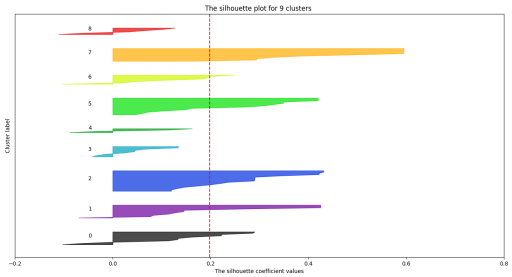
(e) The silhouette plot for 6 clusters



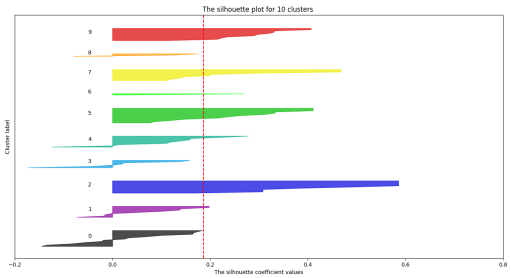
(f) The silhouette plot for 7 clusters



(g) The silhouette plot for 8 clusters



(h) The silhouette plot for 9 clusters

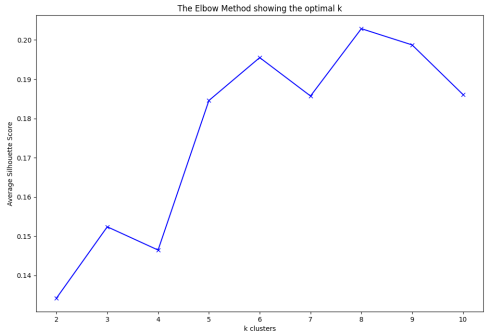


(i) The silhouette plot for 10 clusters

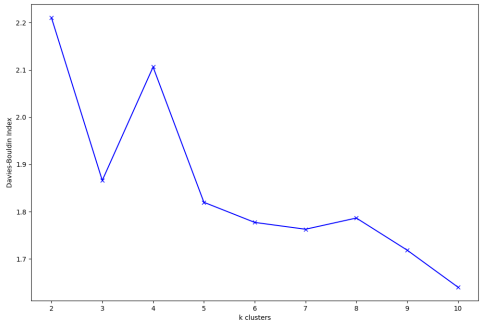
Figure 3.8: The silhouette plot of K-Modes clustering method for the various clusters

This leaves two values for K for which nothing has been said. These are the two best ones, although they both present issues. The figure 3.8d has almost a factor three between its smallest and largest clusters. Nevertheless, it has, in each cluster, s silhouette coefficient higher than the average, which proves a certain balance between all clusters. As for figure 3.8e, it is generally the most balanced except for one cluster which behaves a little less well according to the silhouette representation.

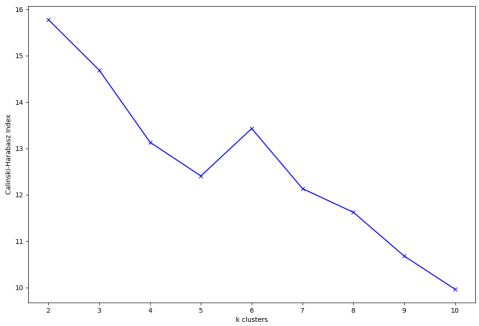
Let's move on to the analysis of the elbows plots. As one could have guessed after the previous analysis of the silhouette representation, there is an elbow shaped segment in the plot figure 3.9a for K=5. However, six clusters give performances as well. Looking at figure 3.9c, a peak is quickly noticed, again when is chosen 6 as number of clusters. Concerning figure 3.9b, the elbow shape is at the level of K=5. This is not very surprising because, as pointed out earlier, the silhouette representation for K=5 was also valid. Thankfully, the value for K=6 is almost identical the Davies-Bouldin index. For the Hamming metric in figure 3.9d, this graph is much more severe with the value 5 as the number of clusters. Indeed, in this case, K=6 is strongly emphasized and thus favored.



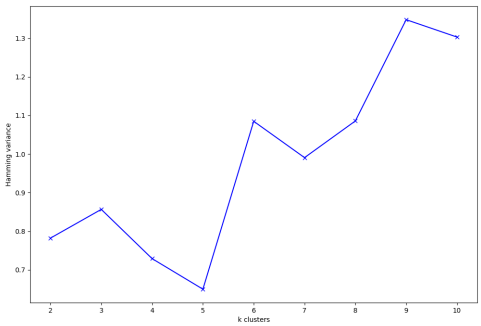
(a) Elbow plot using Silhouette score



(b) Elbow plot using Davies-Bouldin Index



(c) Elbow plot using Calinski-Harabasz Index



(d) Elbow plot using Hamming variance score

Figure 3.9: The Elbow Plots of K-Modes method

3.4.3.3 Conclusion

As noticed during the analysis of the representation silhouette, K values 5 and 6 stood out. The figure 3.9c and figure 3.9d defined K=6 as the best solution and sometimes by far (figure 3.9b and figure 3.9d). Although figure 3.9a and figure 3.9b puts forward 5 as the ideal number of clusters, it showed that K=6 was only slightly worse. For those different reasons, 6 is chosen as the best number of clusters for K-Modes. This is not really surprising since K=6 was already the ideal choice in sec-

tion 3.4.1 and in section 3.4.2.

Metrics / # of clusters	Silhouette score	Davies-Bouldin Index	Calinski-Harabasz Index	Hamming Variance Score
2	0.134196	2.210586	**15.778262**	0.782028
3	0.152390	1.866099	14.687782	0.856504
4	0.146460	2.106203	13.132777	0.728932
5	0.184567	1.819836	12.406879	0.649732
6	0.195509	1.777216	13.429965	1.084786
7	0.185716	1.762776	12.133495	0.990638
8	**0.202877**	1.786505	11.623234	1.085631
9	0.198732	1.718527	10.679703	**1.348115**
10	0.186096	**1.640137**	9.963981	1.303051

Table 3.6: Metrics Values for K-Modes

3.4.4 Self-Organizing Maps

3.4.4.1 Introduction

Self-organizing maps (SOM) are also called Kohonen maps after the professor who is considered as the father of the SOM. As its name suggests, the model organizes itself through learning rules and interactions. It is the only method that will be seen in this paper that uses an artificial neural network (ANN). Moreover, unlike the first three algorithms, the number of clusters desired can be given as input. Indeed, as mentioned in section 3.3, this method is the only one that does not require the number of clusters wanted in its input but it only needs the database. This will be discussed in section 3.4.4.2.

Before diving into the functioning of the algorithm, it is necessary to explain the neighborhood principle for a SOM. As shown in figure 3.10a, the central neuron has 9 neighbors in its neighborhood. Self-organizing maps are made up of neurons which are organized as low dimensional rectangular grids in this case. Hexagonal shaped grid can also be used. Finally, the aim of a self-organizing map is to find neurons' weights values such that the adjacent neurons (i.e. in the neighborhood) have similar weight values.

Let's explain the SOM algorithm, which is certainly well defined in [12]. The grid of size $n \times m$ which will be composed of neurons is created. These neurons will of course share a dimension of the same size as the number of features of the input data objects in order to obtain a weight for each feature. The initialization part is simply attributing a random number for each neuron and normalizing it. From now on, the algorithm can be seen as three different parts succeeding one another in a loop, as explained at the University of Waterloo in 2019 [35].

1. Competition: A data object is randomly taken from this database and sent to the SOM where a winner neuron has to be chosen. This one will be the neuron with the lowest dissimilarity value between the input data object and the different neurons. The dissimilarity calculation can be simply computed with the Euclidean metric.

2. Collaboration: This is the idea that the winner neuron "exchanges information" with its neighbors. A neighbor rate is thus computed and denoted as :

$$h_{i,j}\left(d_{i,j}\right) = \exp\left(\frac{-d_{i,j}^2}{2\delta^2}\right) \tag{3.2}$$

where $d_{i,j}$ is the distance between the winner neuron i and its neighbor neuron j and where δ is the deviation of the Gaussian neighborhood representation denoted as $\delta(n) = \delta_0 \exp\left(\frac{-n}{T}\right)$ (n being the number of iteration and T a constant). This neighbor rate will therefore define the rate at which the information is shared.

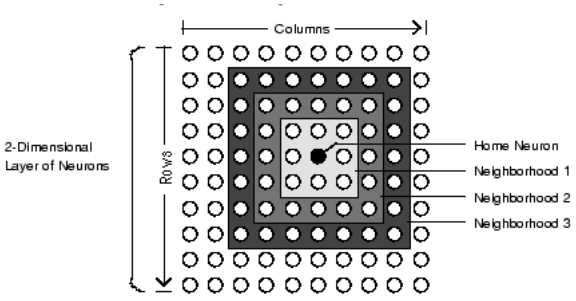
3. Weight update: It is now possible to update the different weights of the winner neuron as well as the weights of neighboring neurons. The update of the different weight's values is done as shown in equation (3.3). After this update, the winner neuron and its neighbors have a greater similarity with the data object provided as input.

$$\mathrm{m_i(t+1)} = \mathrm{m_i(t)} + \alpha(\mathrm{t}) \cdot \mathrm{h_i(t)} \cdot |\mathrm{x(t)} - \mathrm{m_i(t)}| \tag{3.3}$$

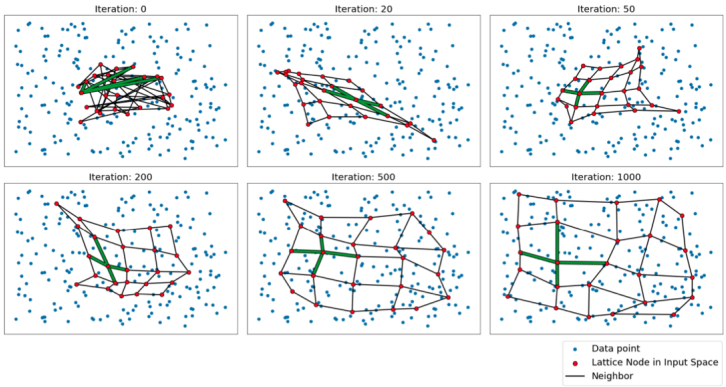
where :

- $\mathrm{m_{i(t)}}$ is the original neuron i
- $\mathrm{m_i}(t+1)$ is the updated neuron
- α(t) is the learning rate
- $\mathrm{h_{ij}}$ is the neighbor rate defined above.
- x(t) is the input data object vector and thus $|\mathrm{x(t)} - \mathrm{m_i(t)}|$ is the distance between the input and the neuron vectors.

Note that the neuron grid adapts more and more to the database as more data objects are provided as input. This last fact is illustrated on figure 3.10b.



(a) SOM neighbors illustration [17]



(b) SOM adaptation to input database illustration [47]

Figure 3.10

Algorithm 4 SOM(X)[12]

Create a n x m size grid of neurons
Assign a random number to each neuron
while all data object are not inputted to SOM **do**
    Input a random data object from database to SOM
    Find the winner neuron ▷ using a (dis)similarity measure as described before
    Adapt the winner and its neighbors' weight vectors ▷ using equation (3.3)
end while
Output Array with each data object labeled

3.4.4.2 Results

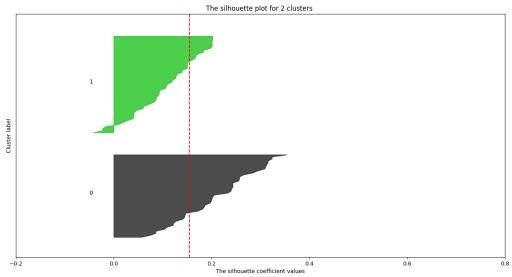
The different results for the SOM clustering were computed thanks to sklearn-som library [48].

As the number K of clusters could not be given as input, the number of clusters was found by trial and error. Indeed, several sizes of n and m (dimension of the neural grid) were tried in order to have more values of K. The ideal is to have a grid shaped and not an array SOM in order to have a better collaboration between the various neighbors.

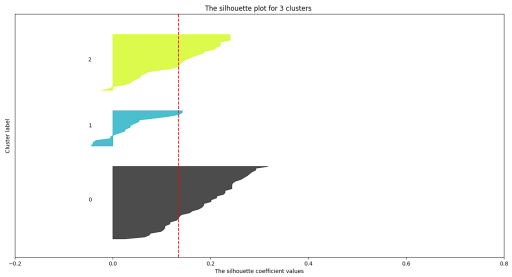
As for the three previous methods, the different graphs of the silhouette representation figure 3.11 will be reviewed.

First of all, figure 3.11i is problematic because it has only eight formed clusters and the formed clusters are of very different sizes. Then, figure 3.11h has cluster (cluster 3) with almost all the silhouette coefficients are negative. Reminder, having a negative silhouette score means that the data objects of a cluster are not welded or/and that the clusters are mixed. Then, K=2 (figure 3.11a) and K=3 (figure 3.11b) are still too small numbers to be able to draw an important lesson from the clustering method. In addition to that, figure 3.11b is highly asymmetric for three clusters. The figure 3.11f and figure 3.11g have big differences in size between clusters and have, in addition to this, some clusters with very low or even negative silhouette scores. The figure 3.11e is not bad even though it has a much smaller cluster (cluster 2) and with lower values for the silhouette score. Regarding this, figure 3.11c and figure 3.11d have the most symmetrical representation with all their silhouette scores above the average.

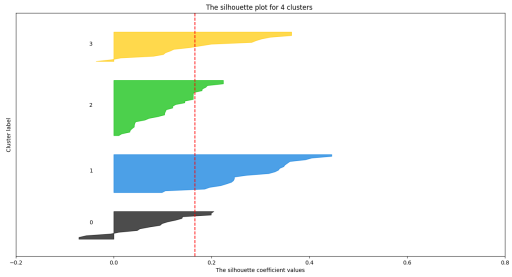
Let's move on to the analysis of the elbows plots (figure 3.12). As far as figure 3.12a is concerned, its representation would put forward 6 or 7 as the ideal value of K. Indeed, for K=6 it is possible to observe a slightly elbow-shaped segment and for K=7, a maximum is observable just before a severe drop. With regards to the Calinski-Harabasz Index (figure 3.12c), the most appropriate choice would be K=4 but values 5 and 6 would also be suitable. Indeed, for these three numbers, the graph remains steady. For figure 3.12b, it is obvious that the best solution would be to have 7 different clusters. For the Davies-Bouldin index, K=6 would be the worst solution. Finally, looking at figure 3.12d with the Hamming variance index, the ideal number of clusters would be six and five being a very bad solution.



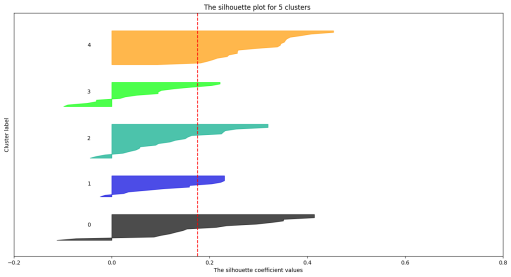
(a) The silhouette plot for the 2 clusters



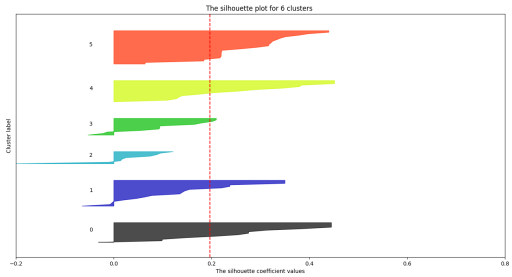
(b) The silhouette plot for 3 clusters



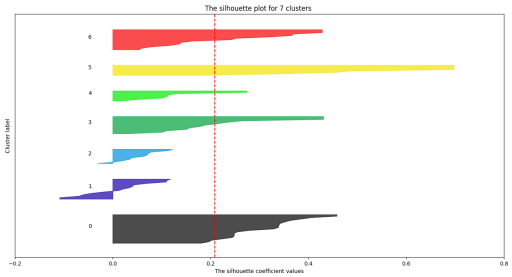
(c) The silhouette plot for 4 clusters



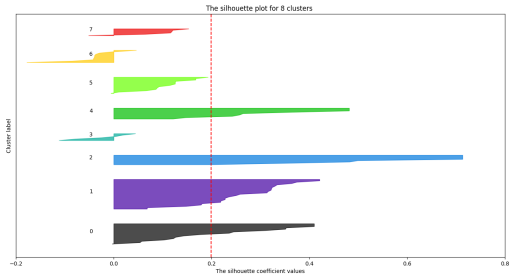
(d) The silhouette plot for 5 clusters



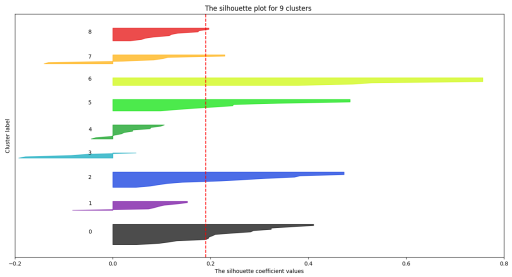
(e) The silhouette plot for 6 clusters



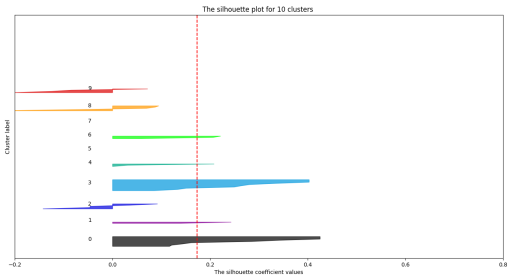
(f) The silhouette plot for 7 clusters



(g) The silhouette plot for 8 clusters

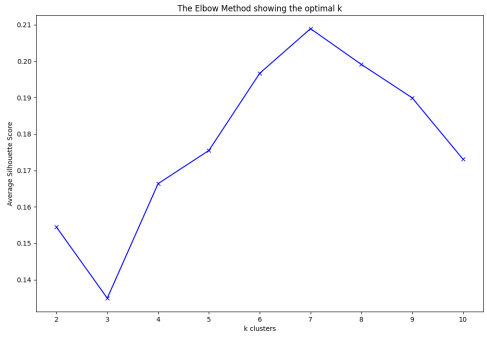


(h) The silhouette plot for 9 clusters

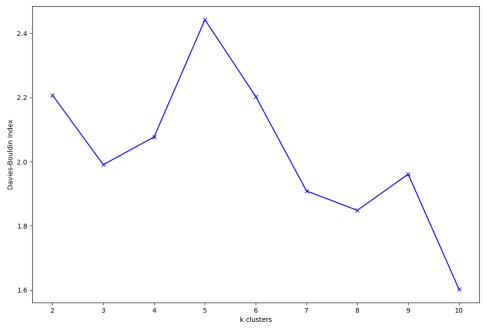


(i) The silhouette plot for 10 clusters

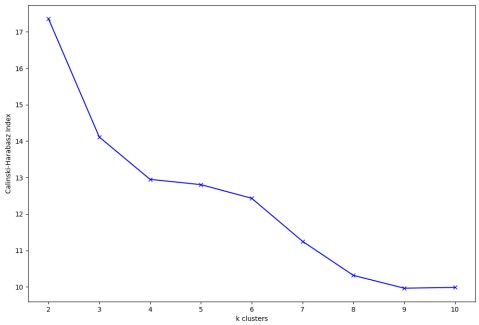
Figure 3.11: The silhouette plot of Self-Organizing maps method for the various clusters



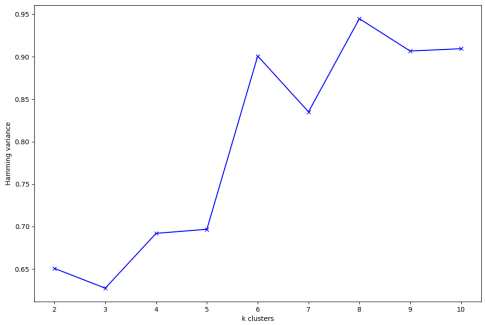
(a) Elbow plot using Silhouette score



(b) Elbow plot using Davies-Bouldin Index



(c) Elbow plot using Calinski-Harabasz Index



(d) Elbow plot using Hamming variance score

Figure 3.12: The Elbow Plots of SOM method

3.4.4.3 Conclusion

As shown in table 3.8, there is no common ideal number of clusters for all metrics. The best clusters depending on the metrics are in black and the worst in red. Indeed, it has been observed many times while testing that the values of the different metrics strongly vary from one iteration to another. It is therefore a method that is not suitable for the current database.

Metrics / # of clusters	Silhouette score	Davies-Bouldin Index	Calinski-Harabasz Index	Hamming Variance Score
2	0.154452	2.207501	**17.363078**	0.650758
3	0.134930	1.990606	14.112968	0.627447
4	0.166378	2.077068	12.950673	0.692030
5	0.175493	2.205290	12.806269	0.696831
6	0.196690	2.203250	12.430964	0.900898
7	**0.208931**	1.908620	11.250080	0.835120
8	0.199114	1.848375	10.314062	**0.944849**
9	0.189948	1.961034	9.962851	0.9067642
10	0.173090	**1.602479**	9.984559	0.909499

Table 3.7: Metrics Values for Self-Organizing Maps

Metrics	Silhouette representation	Silhouette score	Davies-Bouldin Index	Calinski-Harabasz Index	Hamming Variance Score
Ideal # of clusters	4 or 5	6 or 7	7 (6)	4, 5 or 6	6 (5)

Table 3.8: Metrics Values for Self-Organizing Maps

3.4.5 Model Selection : Conclusion

As shown in table 3.9, K-Means, Bisecting K-Means and K-Modes output the best performances. The silhouette score and Calinski-Harabasz index indicate that K-Means has the finest performances. Whereas, it is the K-Modes clustering method that has a higher Hamming variance score. Bisecting K-Means has the highest performances regarding the Davies-Bouldins index. Concerning Self-Organizing maps, it is not possible to determine the best number of clusters as mentioned in section 3.4.4 and thus it is not a good clustering method for the database used. Moreover, as explained, this methods has variable outputs from one iteration to another and is therefore not reliable. In conclusion, all metrics being taken into account, K-Means is the most reliable clustering method for the actual database.

Method used	# of clusters	Silhouette score	Davies-Bouldin Index	Calinski-Harabasz Index	Hamming Variance Score
K-Means	6	0.214856	1.784464	13.371051	0.932545
Bisecting K-Means	6	0.210528	1.696555	13.012042	0.972333
K-Modes	6	0.195509	1.777216	13.429965	1.084786
SOM	/	/	/	/	/

Table 3.9: Recap table of the methods metrics with the most efficient number of clusters

3.5 Feature selection

Feature selection (Liu and Motoda 1998 [33]) is an important part of machine learning. In addition, determining the most relevant features of data objects in a dataset regularly results in better performance. In this thesis, it is important to identify the most relevant features in order to guide a judge/lawyer on what the most important aspects of an antitrust case are. This will allow to identify which features characterized the different antitrust cases at best. Moreover, given the database, one cannot rely on classical feature selection methods such as chi square, entropy, etc. Indeed, as for the clustering algorithms, the unsupervised feature selection methods are the only methods that can be relied on. The methods used are Low-Variance, Laplacian Score, SPEC, MCFS and USFSM.

This master thesis focuses only filter features selection techniques. This type of feature selection technique can be categorized of uni-variate or multi-variate[13]. Both will be used in this thesis.

All the values of the different scores obtained according to the feature selection method used and without feature selection are available in appendix A. In this thesis, six features were determined as the unsupervised spectral feature selection method is choosing the best number of features on its own (which is six) and that the analysis must stay consistent. Moreover, A. Ittoo chose the approximate same number of features in his paper [26], this thesis therefore stayed consistent with his analysis.

3.5.1 Uni-variate Filters : Spectral Similarity

Uni-variate filters techniques' purpose is to evaluate each feature according to a specific criterion to create a ranked list of features[39]. The output of such a feature selection technique would be a created list with all selected features' name. The spectral similarity approach [34], which is the approach used in the following uni-variate filter techniques, computes the mentioned list using similarities between objects. These similarities are represented in a similarity matrix S. So having n data objects, it is now possible to measure the similarity $s_{ij} = s(x_i, x_j)$ between two data objects x_i and x_j which can be computed in different ways as explained in the following sections.

The similarity matrix S is thus constructed using relations between objects which is s_{ij} described above. From this similarity matrix, a similarity graph S is built and is denoted as :

$$G = (V, E)$$

Where V is the vertex set of all vertices v_i (a vertice v_i representing an object x_i)
Two vertices v_i and v_j are thus connected if s_{ij} is positive or higher than an imposed threshold. Finally, s_{ij} is used as weight for the edge also denoted as w_{ij} . Now that the structure of the graph is understood, let's link it with clustering. The aim for clustering would be to find groups within this graph with high weights (i.e. the objects are similar and thus this group would be a cluster) and that edges of different groups have low weights (i.e. the objects are dissimilar and thus aren't in the same cluster).

From this graph representation, it is now possible to compute the Laplacian matrix L denoted as :

$$L = D - W$$

where

- D is the diagonal degree matrix where $d_i = \sum_{j=1}^{n} w_{ij}$ are the degree situated on the diagonal of D
- W is the weighted adjacency matrix $W = (w_{ij})_{i,j=1,...n}$

Thus, from the graph representation is created an eigen-system of Laplacian or normalized Laplacian matrices used to determine the wished ordered list.

3.5.1.1 Laplacian Score

The Laplacian score technique aims to evaluate the features locality preserving power[43]. This method works as follows :

1. A graph G of n nodes has to be constructed as explained in section 3.5.1
2. Compute the weight matrix W where $w_{ij} = \exp -\frac{||x_i - x_j||^2}{t}$, ($t$ being a constant) if x_i and x_j are connected and $w_{ij} = 0$ if they aren't connected. This means that features preserving the graph's initial structure have high weight's values.
3. Compute for the i^{th} feature $\widetilde{\mathbf{f_i}} = \mathbf{f_i} - \frac{\mathbf{f_i^T} L \mathbb{1}}{\mathbb{1} D \mathbb{1}} \mathbb{1}$ where $\mathbf{f_i}$ is the vector of all samples of feature i $[f_{i1}, ..., f_{in}]$ and where $\mathbb{1} = [1, ..., 1]^T$

4. Compute the Laplacian score for the i^{th} feature denoted as : $L_r = \frac{\tilde{\mathbf{f}}_\mathbf{i}^\mathbf{T} L \tilde{\mathbf{f}}_\mathbf{i}}{\tilde{\mathbf{f}}_\mathbf{i}^\mathbf{T} D \tilde{\mathbf{f}}_\mathbf{i}}$

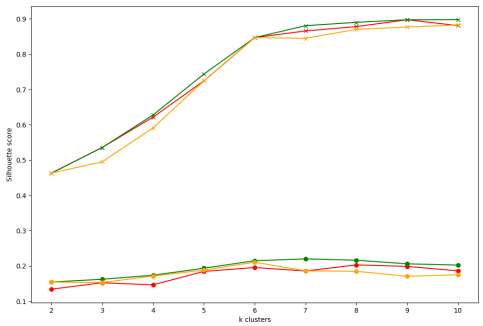
Assigning high weights to features preserving the initial similarity graph G suggests that objects that are near are probably in the same cluster and will therefore have high valuers. On the contrary, distant objects will have smaller weight as they probably aren't in the same cluster.

When running lap_score (from library [32]) on the database and asking for the best 10 features, the following ones are chosen :

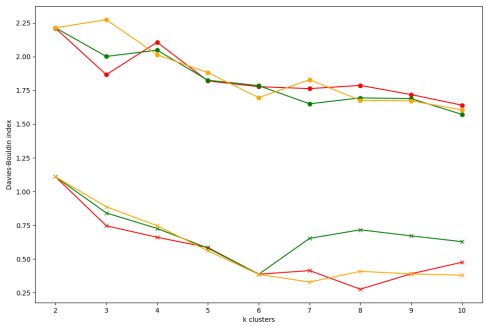
REMEDY 13	INDUSTRY_Transportation industry	CONDUCT P
REMEDY 1	INDUSTRY_Professional/Trade association healthcare	CONDUCT H

Table 3.10: Selected features with Laplacian score

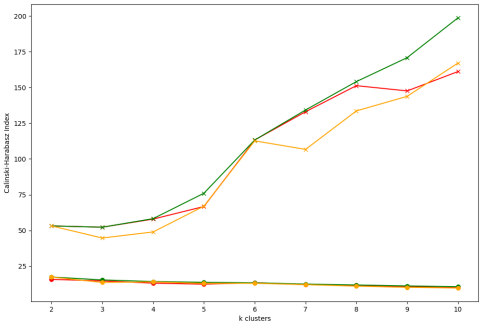
Now, let's analyze the performance of clustering algorithms when using these features. In order to understand figure 3.13, it should first be noted that the graphs marked by circles are graphs of scores computed from the different clustering methods that have not undergone any feature selection. On the contrary, the graphs marked by a crosses are the scores obtained thanks to the clustering methods after feature selection. In this case, the feature selection technique is the Laplacian score. The red curves correspond to the scores of the K-Modes algorithm. Whereas the orange and green curves are respectively to the scores obtained by the Bisecting K-Means and K-Means algorithms.



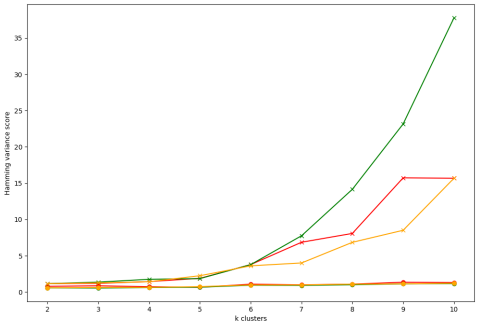
(a) Elbow plot using Silhouette score



(b) Elbow plot using Davies-Bouldin Index



(c) Elbow plot using Calinski-Harabasz Index



(d) Elbow plot using Hamming variance score

Figure 3.13: The Elbow Plots of the different clustering using Lap Score feature selection

Looking first at the graph representing the silhouette score (figure 3.13a), there is an increase of this score for the three clustering methods. This is a good thing knowing that the silhouette score shows good performances for the clustering algorithms if it is high. The same analysis can be drawn with the Calinski-Harabasz score figure 3.13c) and hamming variance score (figure 3.13d). It is still important to note that the scores obtained with Calinski-Harabasz index increase, while without feature selection, they were slightly decreasing. Moreover in regard of the silhouette scores, Calinski-Harabasz index and Hamming variance scores, K-Means outperforms both other clustering methods. Concerning the scores obtained with the Davies-Bouldin index (figure 3.13b), they are halved on average. This also induces good performances because it must be as small as possible. Indeed, the closer this score is to zero, the less overlaps can occur between clusters. It is the only metric for which K-Means is not the best performing clustering algorithm. However all three clustering techniques performs in a similar way. In conclusion, the curve representing K-Means performs better than the other curves.

3.5.1.2 SPEC

The SPEC unsupervised feature selection was established in 2007 by Zheng Zhao et al [59] stems for SPECtral decomposition. It works as follows.

First, the similarity matrix must be computed using a specific similarity measure. As mentioned Zhen Zhao et al. in [59], the best similarity measure for small databases is the RBF kernel function :

$$s_{ij} = \exp -\frac{||x_i - x_j||^2}{2\delta^2}$$

where δ is a parameter.

Now, the feature selection ranking function also has to be defined. Zheng Zhao et al described three different functions which were each more efficient in specific cases. The second one will be discussed in this thesis for the small database used. Knowing that ξ_0 if the first eigen-vector of L and that $\mathbf{f_i} = [f_{i1}, ..., f_{in}]$ is the feature vector composed of all instances of feature F_i, the ranking function is denoted by ϕ:

$$\phi(F_i) = \frac{\widehat{\mathbf{f}}_i^T L \widehat{\mathbf{f}}_i}{1 - \widehat{\mathbf{f}}_i^T \xi_0}$$

where $\widehat{\mathbf{f}}_i = \frac{||\widetilde{\mathbf{f}}_\mathbf{i}||}{\widetilde{\mathbf{f}}_\mathbf{i}}$ with $\widetilde{\mathbf{f}}_\mathbf{i}$ being the weighted vector of F_i.
Now that all the basics of SPEC are explained, let's dive into how it works.

1. First, the similarity matrix S needs to be built in order to get the similarity graph G
2. Then, W, D and L must be built
3. For each feature : Update the value of $\widehat{\mathbf{f}}$ to compute the ranking function $\phi(F_i)$ or the i_{th} feature. The computed value of each feature is stored in the selected features (SF) vector. This vector thus contains the values computed by the ranking function $\phi(F_i)$ for each feature.
4. Finally, the SF vector is ranked in ascending order as smaller ϕ describes a good separability.

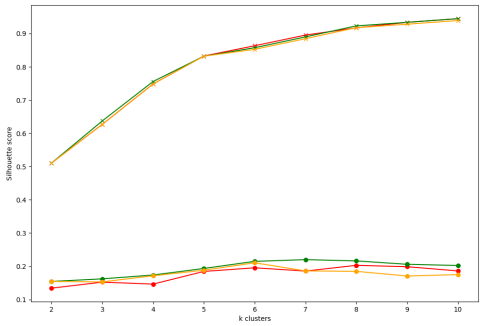
After running the SPEC feature selection algorithm (from library **??**), the features shown in table 3.11 are outputted.

CONDUCT F	INDUSTRY_Gas&Oil	CONDUCT L
REMEDY 7	REMEDY 8	CONDUCT B

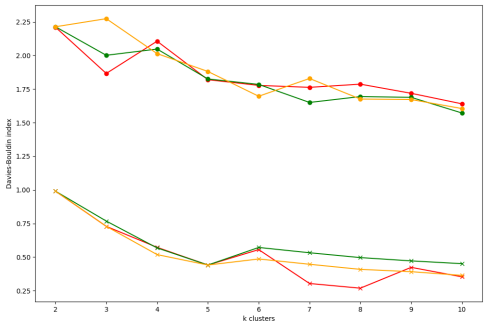
Table 3.11: Selected features with SPEC

Let's now take a look at the different graphs shown in figure 3.14. The specificities of the marks ('o' for results without feature selection, 'x' for results with feature selection) and the colors of the graphs are the same as described in section 3.5.1.1 (red is K-Modes, orange is Bisecting K-Means and green is K-Means).

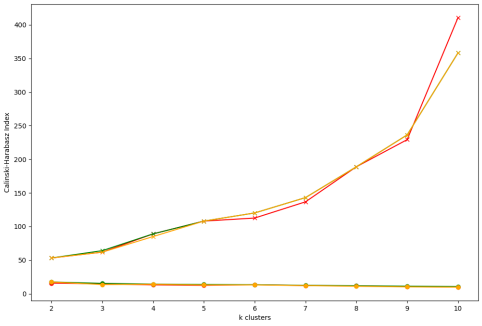
In contrast with the conclusions drawn Lap score concerning the best performing clustering method, here Bisecting K-Means and K-means seem to have similar performances. The silhouettes (figure 3.14a), Calinski-Harabasz (figure 3.14c and Hamming scores (figure 3.14d) increase and the Davies-Bouldin index (figure 3.14b) decreases for all clustering algorithms. Nevertheless, the performances are almost exactly the same for the Calinski-Harabasz, silhouette and hamming score. Regarding the Davies-Bouldin, Bisecting K-Means seems to outperform the other clustering methods. However this is verified only for values of K higher to six. Moreover, as explained in section 3.4, having more than six clusters is never an optimal solution. Therefore, for this feature selection technique, K-Means and Bisecting K-Means have similar performances.



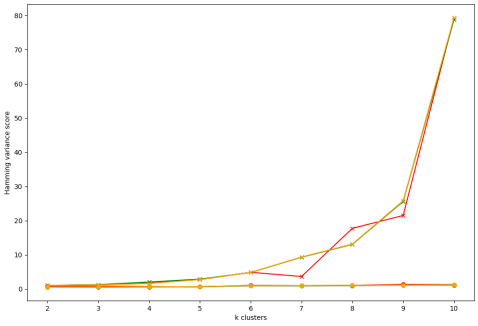
(a) Elbow plot using Silhouette score



(b) Elbow plot using Davies-Bouldin Index



(c) Elbow plot using Calinski-Harabasz Index



(d) Elbow plot using Hamming variance score

Figure 3.14: The Elbow Plots of the different clustering using SPEC feature selection

3.5.1.3 USFSM

USFSM stands for unsupervised spectral feature selection method for mixed data. It has been developed by Saúl Solorio-Fernández et al [51] in 2019. As the name suggests, this method is used for databases with mixed data (i.e. categorical and numerical). USFSM thus allows to make feature selection on the database including the categorical variables. It was mainly used because k-modes is used as a clustering method for this thesis. This method has been implemented based on their paper. Let's explain how it works.

The real problem that is faced is to create a similarity matrix with categorical values as well as with numerical values. The similarity matrix uses a clinical kernel [1] as similarity measure. Thus, the similarity measure w_{ij} between two objects x_i and x_j is denoted by :

$$w_{ij} = \frac{1}{n} \sum_{p-1}^{n} k(x_{ip}, x_{jp})$$

where n is the number features per object and k is a sub-kernel [1]. These sub-kernels can be computed as :

$$k(a, b) = 1 - dist(a, b)$$

where a and b can be categorical or numerical variables. Moreover there is also the $dist()$ function. It is this function that allows to manage the different types of variables. It is defined as follows :

$$dist(a,b) = \begin{cases} 1 & \text{if } a \text{ and } b \text{ are categorical variables and if } a = b \\ 0 & \text{if } a \text{ and } b \text{ are categorical variables and if } a \neq b \\ \frac{|a-b|}{\max_F - \min_F} & \text{if } a \text{ and } b \text{ are numerical variables} \end{cases}$$

where $\max_F$ and $\min_F$ are the maximal and minimal values that can take the feature F. Now that the similarity matrix can be built, the spectral gap score needs to be computed as well [51] in order to understand how the USFSM's algorithm works. This score is computed as follows :

$$\gamma(X, k) = \sum_{i=2}^{k} \sum_{j=i+1}^{k+1} \left| \frac{\lambda_i - \lambda_j}{\sum_{i=2}^{k} \lambda_i} \right| \tag{3.4}$$

where λ is the eigen-values of L sorted in an ascending order. Finally, this spectral gap score is used to compute $\phi(F_i)$, the feature evaluation score denoted as :

$$\phi(F_i) = \gamma(X, k) - \gamma(X_i, k) \tag{3.5}$$

where X_i is the database without the i^{th} feature. A feature is defined as relevant if $\phi(F_i)$ is positive. Now that the different variables used in the USFSM algorithm are understood, this latter is shown in algorithm 5.

Algorithm 5 USFSM(X = dataset with n features)[51]

Build laplacian L and similarity matrix W from X
Compute the eigen-values of L
Compute $\gamma(X, k)$ ▷ see (3.4)
for i in n **do**
    Build L_i and W_i from X_i ▷ where X_i is the database X without the i^{th} feature
    Compute the eigen-values of L_i
    Compute $\gamma(X_i, k)$ ▷ see (3.4)
    Compute $\phi(F_i)$ ▷ see (3.5)
    if $\phi(F_i) > 0$ **then**
        Append feature F_i to array FS ▷ FS = Selected feature array
    end if
end for
Output Array of selected features FS

A big difference between this algorithm and the two previous ones is that the number of features to select can not be chosen in advance. This feature selection method has therefore been tested on the database with the categorical and numerical values which is shown in table 3.12. This dataset is used by the K-Modes (section 3.4.3) clustering method. Moreover, USFSM has also been used for the database used by the other clustering algorithms which are K-Means(section 3.4.1) and bisecting K-Means(section 3.4.2). USFSM on this dataset can be seen in table 3.13.

CONDUCT D	CONDUCT F	CONDUCT M
CONDUCT P	REMEDY 3	REMEDY 12

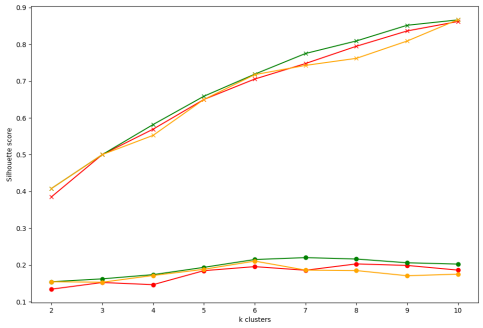
Table 3.12: Selected features with USFSM from the dataset with categorical and numerical variables

CONDUCT C	INDUSTRY_Healthcare/Pharmaceutical
REMEDY 8	INDUSTRY_Professional/Trade association healthcare
REMEDY 11	INDUSTRY_Professional/Trade association others
CONDUCT E	INDUSTRY_Professional/Trade association real estate
REMEDY 2	

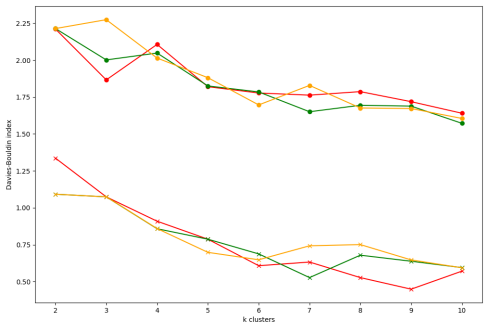
Table 3.13: Selected features with USFSM from the dataset with numerical variables

Let's now take a look at figure 3.14. The marks ('o' for results without feature selection, 'x' for results with feature selection) as well as the colors of the graphs (red is K-Modes, orange is Bisecting K-Means and green is K-Means) are the same as described in section 3.5.1.1.

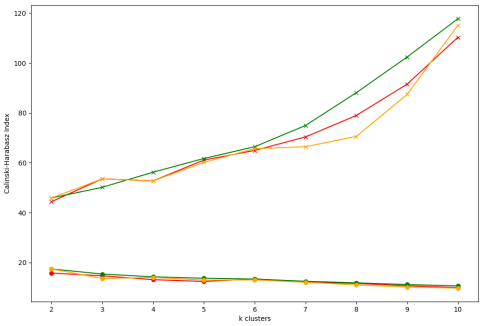
The scores obtained via the Davies-Bouldin index with feature selection(figure 3.15b) are lower than the scores obtained without feature selection. Hence, the feature selection improves the performances. Moreover, by looking at figure 3.15a, one can see that the silhouette scores improving after feature selection. Furthermore, concerning figure 3.15c and figure 3.15d, an exponential improvement in the displayed scores can be observed. A general performance enhancement is therefore observed when using USFSM. Again with this feature selection method, K-Means stems as the best clustering algorithm among the three studied.



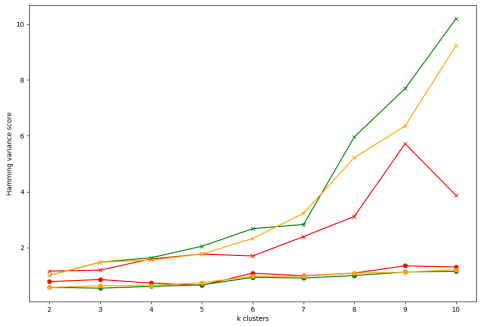
(a) Elbow plot using Silhouette score



(b) Elbow plot using Davies-Bouldin Index



(c) Elbow plot using Calinski-Harabasz Index



(d) Elbow plot using Hamming variance score

Figure 3.15: The Elbow Plots of the different clustering using USFSM

3.5.2 Multi-variate Filters

Unlike feature selection methods which are uni-variate, multi-variate techniques study features jointly rather than considering them individually. This regularly allows to detect redundant and irrelevant variables[45]. These techniques are divided into three parts, the first being bio-inspired techniques, the second, stat-based methods and finally the third is a spectral approach introduced by Garcia-Garcia et al (2009[49]). The two latter ones are discussed in this study.

3.5.2.1 Spectral/sparse learning : MCFS

This multi-variate spectral/sparse learning methods has been introduced and developed by Cai et al (2010[25]). The MCFS stands for Multi-Cluster Feature Selection is multi-variate spectral feature selection approach. This method is composed of three distinct parts as described by Cai et al :

1 : Spectral embedding for clusters analysis

2 : Learning sparse coefficient vectors

3 : Feature selection from the sparse coefficient vectors

[1] : This first part of the algorithm works as explained in section 3.5.1. Indeed, the Laplacian L is computed using the similarity matrix of the database X. Moreover, the eigenvectors are also computed from the eigen-problem :

$$\mathbf{Ly} = \lambda \mathbf{Dy} \tag{3.6}$$

where λ is a eigenvalue and y an eigenvector.

Moreover, the solution of the above problem can be noted as $\mathbf{Y} = [\mathbf{y_1}, ..., \mathbf{y_k}]$ where $\mathbf{y_k}$ are the eigenvectors and k is the number of clusters. $\mathbf{Y}$ is also called the flat embedding of the data points of X. Now that the spectral part of MCFS is done, let's discuss about how the learning sparse coefficient vectors are computed.

[2] : From the flat embedding $\mathbf{Y}$ the subset of relevant features can be found. In Multi-Cluster Feature Selection, the aim is to find a solution to the regression problem denoted by :

$$\min_{\mathbf{a}_k} ||\mathbf{y_k} - \mathbf{X}^T \mathbf{a}_k|| \tag{3.7}$$

where $\mathbf{a_k}$ is M-dimensional array (M being the number of features) containing the combination coefficient for different features in $\mathbf{y_k}$. The regression problemed is solved using Least Angle Regression algorithm (LAR) introduced by G. Efron (2004, [5]).

[3] : The features can now be selected. It is done by computing the MCFS score of every feature i as follows,

$$MCFS(i) = \max_k |a_{k,i}| \text{ where } a_{k,i} \text{ is the } i^{th} \text{ element of vector } \mathbf{a_k} \tag{3.8}$$

The MCFS scores are ordered in descending order. The algorithm is algorithm 6.

Algorithm 6 MCFS(X = dataset with m features, K = number of clusters, d = the number of selected features needed)[25]

1: Build similarity graph ▷ As described in section 3.5.1
2: Compute the K eigenvectors $\mathbf{y_k}$ using (3.6)
3: **for** i in K **do**
4: Compute the sparse coefficient vector $\mathbf{a_i}$ by solving (3.7)
5: Compute the MCFS(i) score by solving (3.8)
6: **end for**
7: Rank the MCFS scores in descending order
8: **Output** d selected features

The selected features [10] outputted by MCFS are displayed in table 3.14.

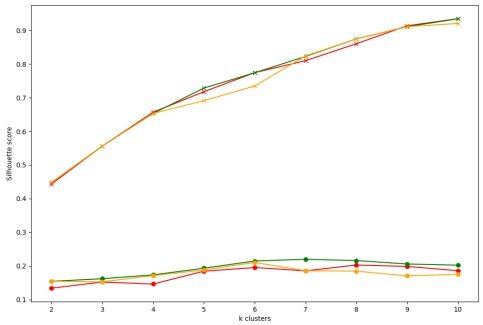
REMEDY 13	INDUSTRY_Real Estate industry	REMEDY 2
REMEDY 14	INDUSTRY_others	CONDUCT A

Table 3.14: Selected features with Low Variance

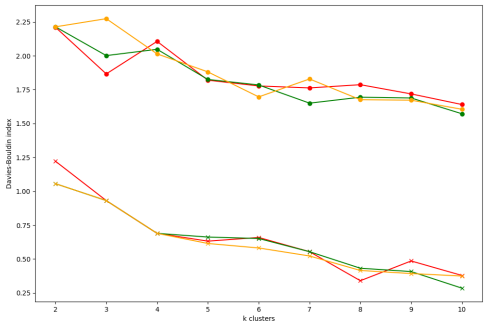
Let's move on to the analysis of figure 3.16. The marks ('o' for results without feature selection, 'x' for results with feature selection) as well as the colors of the graphs (red is K-Modes, orange is Bisecting K-Means and green is K-Means) are the same as described in section 3.5.1.1.

[10] Implementation implementation of MCFS from library [32]

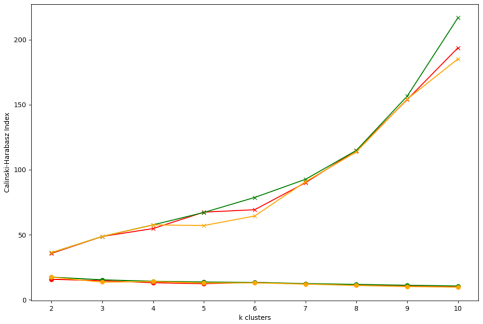
The silhouette graph figure 3.16a show, K-Means' curve generally shows better performance. This is still the case for the figure 3.16c where K-Means is the clustering method giving the best performances, whatever the number of clusters. Regarding the performances with the hamming score in figure 3.16d, K-Modes performs the worst and K-Means is still the best performing clustering algorithm. Finally, regarding figure 3.16b, none of the three clustering methods stands out. Indeed, all clustering algorithm have good performances but none of them outperforms the others. As for the clustering algorithm that works best with MCFS, it is still K-Means in most cases.



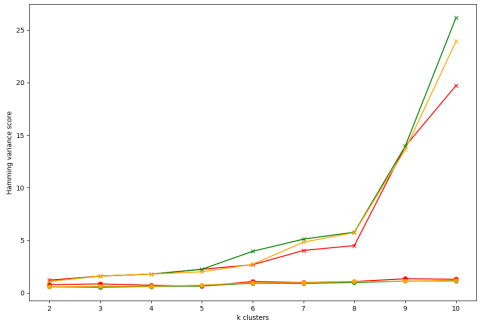
(a) Elbow plot using Silhouette score



(b) Elbow plot using Davies-Bouldin Index



(c) Elbow plot using Calinski-Harabasz Index



(d) Elbow plot using Hamming variance score

Figure 3.16: The Elbow Plots of the different clustering using MCFS method

3.5.2.2 Statistical based : Low Variance

This method sets aside features with too little variance, i.e. features that keep the same value for most data objects in the database. Logically enough, this means that these features do not give much information and are therefore not useful to determine a pattern from the database. Indeed, a low variance for a feature means that the feature is constant throughout the almost the entire database. It therefore should be removed. However, what does 'low' mean? In this paper, the six most relevant features were desired. To determine the minimal value of the variance which is is accepted, a threshold must also be provided. It has been found by trial and error and is 0,215. This simple method works as follows :

1 : Compute the variance for each feature of the database

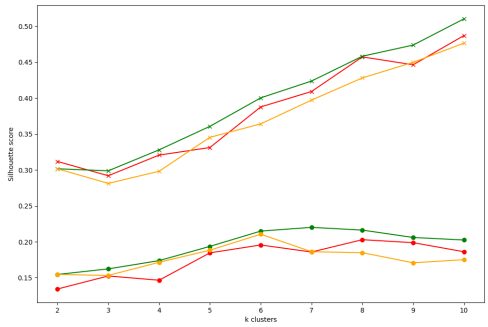
2 : Select the feature whose computed variance is higher than the threshold

The table 3.15 displays the selected features by the Low Variance feature selection method[11].

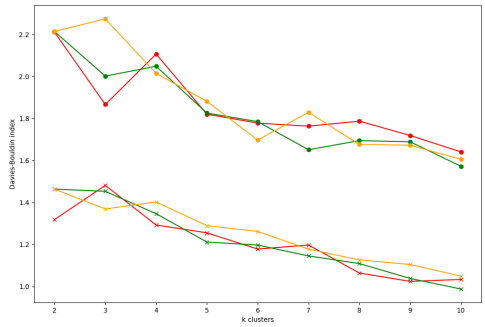
CONDUCT C	CONDUCT P	CONDUCT E
REMEDY 8	REMEDY 11	REMEDY 3

Table 3.15: Selected features with Low Variance

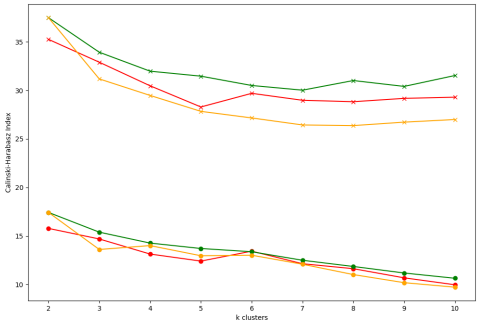
Let's now try to determine the clustering algorithm that gives the best performances with the Low Variance feature selection method. The figure 3.17a and figure 3.17c show that K-Means curves are superior to those of K-Modes and Bisecting K-Means. As for the Davies-Bouldin score in figure 3.17b, K-Means as well as K-Modes curves are still the ones giving the best performances as they are the closest to the x-axis. However the K-Means' curve is more stable. Finally, regarding figure 3.17d with the hamming variance score, K-Means is the one showing the best performances as it has the best scores for all number of clusters except two of of them. In a nutshell, K-Means outperforms the two other clustering clustering methods with Low Variance feature selection technique.



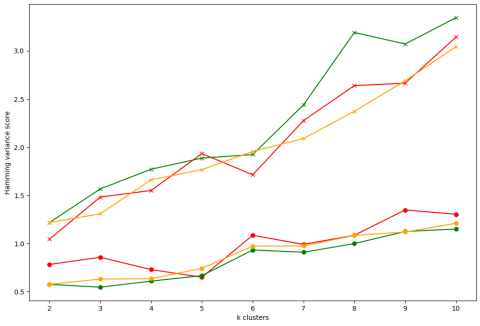
(a) Elbow plot using Silhouette score



(b) Elbow plot using Davies-Bouldin Index



(c) Elbow plot using Calinski-Harabasz Index



(d) Elbow plot using Hamming variance score

Figure 3.17: The Elbow Plots of the different clustering using Low Variance

3.5.3 Feature Selection : Conclusion

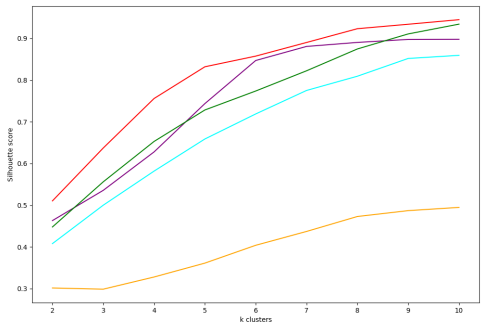
During the analysis, from the five feature selection methods used, one of the clustering methods stood out in terms of performance. Indeed, the K-Means algorithm outperforms Bisecting K-Means and K-Modes, no matter which feature selection method (except one where Bisecting K-Means and K-Means

[11]Implementation implementation of Low Variance from library [32]

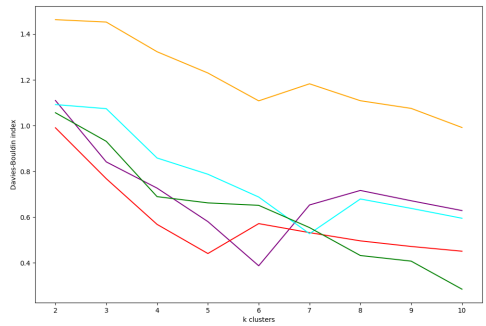
had similar performances) or metric (in most cases) are used. It is therefore time to compare the performances of the K-Means algorithm for the different feature selection methods. This will allow to determine which feature selection method suits the most the database. In figure 3.18 is presented all the K-Means curves with each feature selection technique and for each metric.

In figure 3.18 :

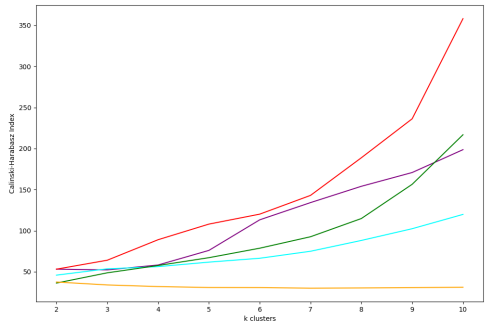
- Purple curve corresponds to K-Means with Laplacian score feature selection method
- Red curve corresponds to K-Means with SPEC feature selection method
- Light blue curve corresponds to K-Means with USFSM
- Green curve corresponds to K-Means with MCFS
- Orange curve corresponds to K-Means with Low Variance feature selection method.



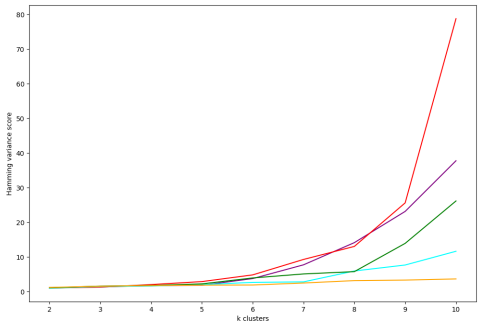
(a) Elbow plot using Silhouette score



(b) Elbow plot using Davies-Bouldin Index



(c) Elbow plot using Calinski-Harabasz Index



(d) Elbow plot using Hamming variance score

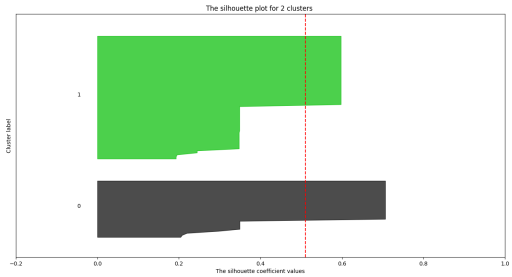
Figure 3.18: The Elbow Plots of the different clustering using Low Variance

Whatever the metric used, it is the spectral decomposition (section 3.5.1.2) feature selection method which is by far the best feature selection method. Indeed, with this method, K-Means has the best performances for almost all metrics. By observing figure 3.18a with the silhouette score, when SPEC feature selection is used, the performances of K-Means exceed all the other clustering methods and this no matter the number of clusters. The exact same analysis can be drawn for figure 3.18c with Calinski-Harabasz index. For Hamming variance score representation, K-Means after SPEC feature selection has the best scores and therefore performances except when the number of cluster is equal

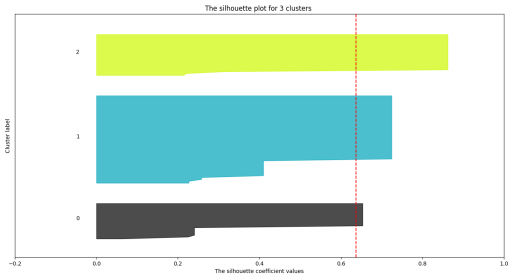
to eight. Finally, when focusing on figure 3.18b with the Davies-Bouldin index, SPEC is best feature technique in terms of performance for $K = [2, 5]$ U 7. Thus, it is not the best feature selection technique for all values of K. However, it is the best method for more than half of the values of K and for other numbers of clusters it is the second best performing feature selection method. In conclusion, with respect to the created database, K-Means after SPEC feature selection is the clustering method that gives the best performances.

Although the clustering and feature selection methods to use has been determined, the number of clusters needs to be checked to see if K=6 is still the best solution. Indeed, the ideal number of clusters may have changed has feature selection has been made. Let's determine the best number of clusters from figure 3.19. As in section 3.4, no useful pattern can derive from a legal perspective if the number of clusters is too small. When looking at figure 3.19h and figure 3.19i, all clusters are not even created. In fact, for figure 3.19h cluster eight is empty and for figure 3.19i clusters eight and nine are empty. In addition to the emptiness of some clusters, others are very small with negative silhouette coefficients. Then, figure 3.19e and figure 3.19f have the exact same issue. Indeed, for those two numbers of clusters, there are large differences in cluster sizes in addition to having clusters whose silhouette coefficients are almost equal to zero. Finally, when analysing figure 3.19d where K equals five, the clusters are quite symmetrical and almost all have the same value for their silhouette coefficient. This conclusion coincides with the analysis drawn from figure 3.18b where K-Means with SPEC feature selection is best performing for five clusters. The other plotted metrics also show slight elbows to the curve.

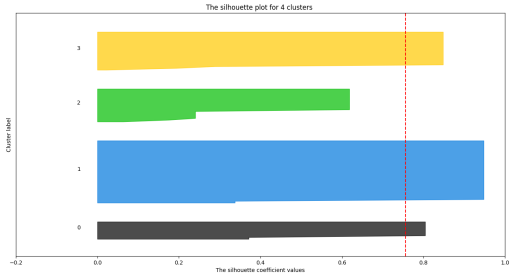
In conclusion, concerning the database used, the ideal is to determine five clusters using the K-Means algorithm after unsupervised spectral feature selection. All this was determined in order to have the most accurate clustering method possible.



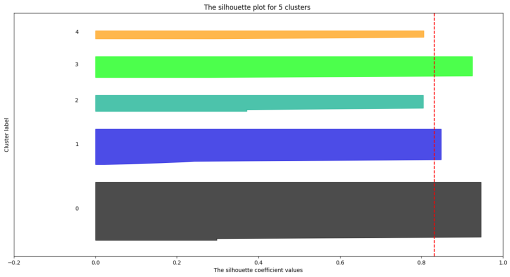
(a) The silhouette plot for the 2 clusters



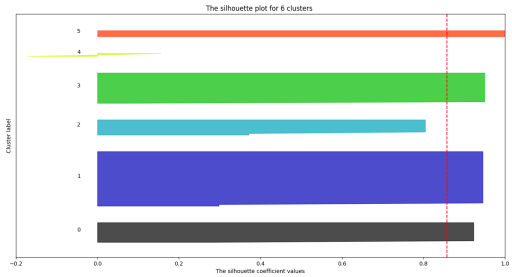
(b) The silhouette plot for 3 clusters



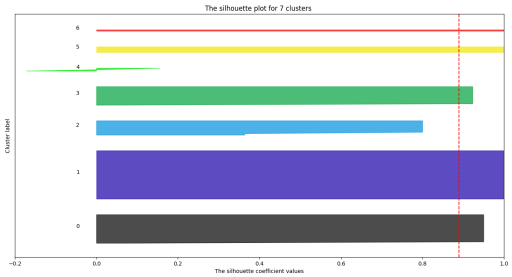
(c) The silhouette plot for 4 clusters



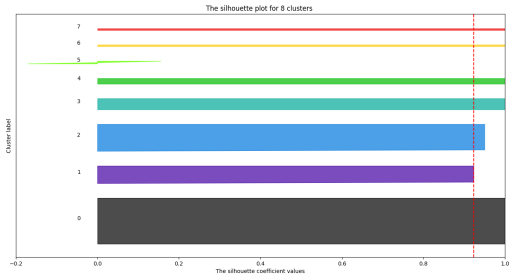
(d) The silhouette plot for 5 clusters



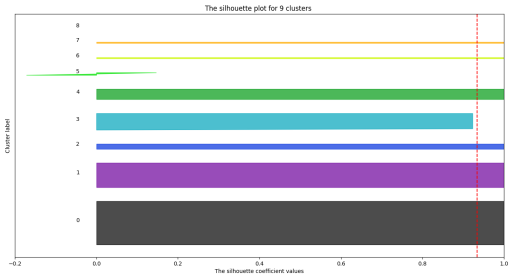
(e) The silhouette plot for 6 clusters



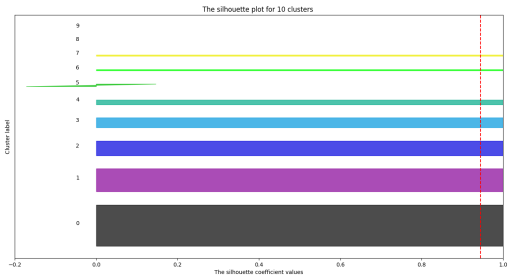
(f) The silhouette plot for 7 clusters



(g) The silhouette plot for 8 clusters



(h) The silhouette plot for 9 clusters



(i) The silhouette plot for 10 clusters

Figure 3.19: The silhouette plot of K-Means method after SPEC for the various clusters

Chapter 4

Antitrust Perspective of the Clusters

Now that we have determined which feature selection to use in order to determine the most relevant features as well as finding the clustering algorithm that gives the best performances, we can compute the clusters. The determined clusters are available on the google sheet[1]. On this google sheet :

- Sheet 1 (DATA FOR US CASES) : The 91 cases with all the information available for every legal case.
- Sheet 2 (Data set): The actual dataset used with only the useful information as features. Indeed, as mentioned in section 3.2, we needed the help of antitrust domain expert to build this database. The expert told us which information was important and which wasn't.
- Sheet 3 (Cluster 0) : This sheet contains all legal cases of cluster 0.
- Sheet 3 (Cluster 1) : This sheet contains all legal cases of cluster 1.
- Sheet 3 (Cluster 2) : This sheet contains all legal cases of cluster 2.
- Sheet 3 (Cluster 3) : This sheet contains all legal cases of cluster 3.
- Sheet 3 (Cluster 4) : This sheet contains all legal cases of cluster 4.

An antitrust perspective of the created clusters will be given in the rest of this chapter[2].

4.1 Cluster 0

This cluster is made up of fourteen antitrust cases. In the entire dataset, there are twenty-four 'others' industry and ten of them are in this cluster. The four other legal cases' industry are one computer, one real estate, one healthcare and one gas&oil. It is important to note that the most recurrent conduct is the "Agreement not to compete" conduct (conduct M). Indeed, conduct M is present for ten cases of the fourteen in cluster zero. Moreover, nine of the ten legal cases with conduct M have 'others' as industry name. All these cases have also similar remedies. Indeed, remedies five, six and seven are always issued. The remedy that is especially important from the three quoted is remedy seven. This remedy is only issued in twenty legal cases amongst which fourteen are in this cluster. In contrast, remedy five and six are issued respectively in 99% and 92% of the legal cases. Furthermore, we can

[1] https://docs.google.com/spreadsheets/d/14nXr8Cm1psosEnEAYy-I62Oje0WGYrVj/edit?usp=sharing&ouid=103570306248967671612&rtpof=true&sd=true

[2] Please note that this legal analysis is based on the knowledge acquired during this thesis. Further analysis will be conducted by an expert in the antitrust field from UPenn University in the weeks following the submission of this thesis

observe that remedy three is used in ten cases amongst which nine have the industry name 'others'. It can also be noted that no antitrust cases are undergoing legal problems due to conducts B, C, D, F, I, K, L, P and Q. The fact that conduct C and P is important as both conducts are adopted in a third of the cases. In addition none of them issued remedies one, eight, nine, twelve, thirteen and fourteen.

Concerning cluster 0, we can conclude that antitrust issues, concerning cases that have been classified in the 'others' industry, are often the agreement not to compete. Moreover, those case solve the issue by always undergoing remedy 7, implementation of an antitrust compliance program and often by being obliged to disclosure information.

In addition, we can note that the Federal Trade Commission did not adopt many conducts per case in this cluster. Indeed, on average a case has 2.57 conducts and 4.13 remedies. This cluster has an average of 1.57 conducts per case. This cluster can thus also be characterized by its small number of conducts per case.

4.2 Cluster 1

This clusters is made up of twenty-three legal cases amongst which eighteen are in the healthcare and pharmaceutical industry. Concerning the conducts, this cluster contains more than half of the cases with conduct E which is price fixing. This is the only useful information that can be drawn from the different conducts. Concerning the remedies, it must be noticed that 100% of the antitrust cases of cluster one have used remedy eight which is contract limitations. The is particularly valuable as only twenty-nine cases issued contract limitations remedy and twenty-three of them are in this cluster.

From this cluster, the algorithm suggests that healthcare industry often raise the antitrust concerns regarding the price fixing and adopt contract limitations as remedy.

4.3 Cluster 2

Cluster two contains eleven anti-competitive legal cases where the most recurrent industry is 'others' with six cases followed by healthcare/pharmaceutical with three. The two other cases are in the transportation and media industry. Concerning the conducts, the FTC adopted conduct L, invitation to collude for all legal cases in cluster two. Moreover, no cases of cluster two had adopted conduct O, concerted practices, which outstanding as a third of the cases of the dataset issued this conduct. Jumping to the remedies, we do not have a lot of information however nine cases amongst the eleven of the cluster adopt remedy 11, limitation in the exchange of information.

The machine learning clustering method suggests , for cluster two, that the Federal Trade Commission often adopts remedy eleven to counter the issue of invitation to collude, conduct L. This makes sense as in order to stop competitors to collude, they should stop exchanging information (remedy eleven).

4.4 Cluster 3

This cluster, which is the biggest, includes thirty-seven cases. A common theme concerning the industry name is that ten of the twelve real estate cases are in this cluster. In addition, concerning the industry, nineteen other cases are related to healthcare. The eight cases left are in the 'others' (five), gas & oil (two) and computer (one) industry. Concerning the conducts, the FTC adopted seventeen times conduct C, refusal to deal, although this conduct was used thirty-one times in the whole dataset. Cluster three contains also twenty-one cases which adopted conduct O (there are twenty-eight in the whole dataset), concerted practices and nineteen adopted conduct P which appeared thirty-two times in the entire database. Regarding the remedies, the obligation to disclosure information was used twenty-two times and remedy two, amendments of the code, appeared twelve times although this remedy only occurred twenty times in the whole dataset.

From this cluster, we can conclude that cases in the real estate industry were in the FTC's sights because of the refusal to deal conduct. Indeed, nine of the real estate cases had conduct C. Moreover, eight of the healthcare cases adopted this conduct too. In addition, nine of the healthcare and all of the real estate cases had conduct O. Furthermore, conduct P is adopted by eight real estate and seven healthcare cases. Thus, the healthcare and the real estate industry undergo the same kind of antitrust conducts. Especially the real estate industry used remedy two. Indeed, this remedy was adopted in nine cases. This leads to think that cases adopting conduct C and O often use remedy two, especially in the real estate industry.

4.5 Cluster 4

Cluster four is the smallest of all. Indeed, it only has six cases among which five are in the healthcare industry and one in others. Except conduct A, exclusionary conduct which appears four times and conduct I, pay for delay, which appears three times, no conduct stands out. Then, regarding the remedies, all antitrust cases adopted remedies seven, implementation of an antitrust compliance program and eight, contract limitations.

This cluster may also be characterized by its low number of conducts and its high number of remedies. Indeed, as mentioned in section 4.1, on average a case has 2.57 conducts and 4,13 remedies. Nevertheless, this particular cluster has an average of 1.8 conduct and 5.3 remedies per case.

Finally, this cluster suggests that cases in the healthcare industry adopting conduct A or I always use remedies seven and eight.

4.6 Conclusion

Thanks to this chapter, it is noticeable that the created clusters make sense. Indeed, some specific patterns were found between the conduct adopted and the remedies used. This reassures the goal of the thesis which was to find common patterns between some cases. As shown in the analysis of each cluster, similarities were detected within the five clusters above.

Chapter 5

Conclusion

In this thesis, we studied different clustering and feature selection methods for a topic that has been little discussed in the machine learning world. Indeed, no study except the one of Dr. Massarotto and Pr. Ittoo (Giovanna Massarotto et al., 2021 [26]), to our knowledge, has yet done a deep research from a computational point of view about antitrust laws. One problem that quickly became apparent was that there was no structured database of all the cases concerning antitrust laws in the United States. This was our first priority and hence our first contribution in this thesis.

Moreover, several unsupervised clustering techniques were tested on the novel database. Indeed, our goal was to find recurrent patterns within the database in order to help antitrust judges in their decision-making. It stands out that the best performing clustering algorithm was K-Means with five clusters. Among the clustering algorithms, self-organizing maps appeared to perform much worse than the other three algorithms. Indeed, the results were found to be highly variable from one compilation to another.

Furthermore, we tried to determine the most relevant features using feature selection. Indeed, this was done in order to help judges by outputting the best variables on which they should be focusing. Five methods of unsupervised filter technique have been developed. The SPEC feature selection was the one giving the best results.

This study finally suggests that machine learning algorithms can discover underlying patterns between the antitrust cases. Although the work done in this thesis can not replace antitrust judges, it can help them in their decision making by showing the similarities between legal cases. Artificial intelligence can thus be a valuable tool in this kind of field.

5.1 Limitations and Possible Improvements

The main limitation of this thesis is the small dataset used with only ninety-one legal cases and that every case had to be curated be hand. In the future, the ideal would be to continue to complete this database and train again the machine learning algorithm in order for the predicted clusters to improve.

In addition, many government agencies operate in the same way as the Federal Trade Commission agency in the US. This means that the analysis made in this thesis could be extended to other antitrust agencies in other countries.

Moreover, auto-encoder could also be developed. It works by creating a code from the input (encoder) and then to rebuild the input from this code (decoder). The output of the decoder is not the perfect input of the encoder but is often the most relevant aspects of the input. This could be useful in order to choose the most relevant aspects of the input space.

Furthermore, concerning the feature selection section, unsupervised wrapper feature selection could also be discussed as only uni- and multi-variate filter techniques were developed. Moreover, in this thesis, the number of features has been chosen with respect to the number of features selected in Pr. Ittoo's paper (Massarotto et al, 2021[26]). That said, perhaps it would be interesting to determine for each feature selection technique, which number of features, which may vary from one method to another, is the one that gives the best performances.

Finally, as mentioned in chapter 4, the identified clusters are going to be analysed and validated by an antitrust field expert. If the expert determines that the labels are correctly identified, those labels could be used as reference to train and evaluate supervised learning techniques. This has already been created on the GitHub but not tested as the expert did not give here opinion on the created clusters.

5.2 Personal Opinion

From my point of view, I find this kind of inter-disciplinary thesis very interesting. Indeed, it has allowed me to be interested in subjects that I would not normally research. It also allows me to see concretely how artificial intelligence can be implemented in various fields. Regarding the automation of the decision-making part of an antitrust judge's job, this thesis showed that patterns and similarities could be identified between antitrust cases but a complete automation without a judge seems to be too soon in my opinion. However, we are well on our way to getting as close as possible to the automation goal. Hence, this thesis can pave the way towards automation of antitrust courts but is not the end product for doing so.

Appendix A

Metrics scores of clustering methods after different feature selection methods

Metric	Clustering method	2	3	4	5	6	7	8	9	10
Silhouette	K-Modes	0.46316	0.53569	0.62158	0.72436	0.84698	0.86579	0.87818	0.89742	0.88100
	K-Means	0.46316	0.53569	0.62780	0.74365	0.84691	0.88063	0.890247	0.89747	0.89774
	Bisecting K-M	0.46316	0.49525	0.59104	0.72436	0.84698	0.84497	0.87019	0.87700	0.88282
Davies-Bouldin	K-Modes	1.10968	0.74620	0.66115	0.58539	0.38713	0.41421	0.27643	0.39056	0.47571
	K-Means	1.10968	0.84132	0.72659	0.58029	0.38713	0.65304	0.71650	0.67136	0.62854
	Bisecting K-M	1.10968	0.88726	0.74654	0.56055	0.38628	0.32921	0.40897	0.38918	0.37994
Calinski-Harabasz	K-Modes	53.16706	52.25917	57.91298	66.73278	113.23348	132.98941	151.23444	147.55810	161.09588
	K-Means	53.16706	52.25917	58.22661	75.99317	113.23348	134.21089	154.06029	170.82864	198.66808
	Bisecting K-M	53.16706	44.68890	48.94023	66.95204	112.641856	106.65877	133.56527	143.78266	167.05080
Hamming	K-Modes	1.15092	1.18987	1.43059	1.88255	3.77385	6.85431	8.07003	15.73087	15.67301
	K-Means	1.15092	1.36255	1.74889	1.84536	3.77385	7.74812	14.15022	23.16051	37.77914
	Bisecting K-M	1.15092	1.25747	1.41328	2.24323	3.60660	3.99969	6.84465	8.50708	15.67301

Table A.1: Score of different clustering algorithm using the Laplacian score feature selection technique

Metric	Clustering method	2	3	4	5	6	7	8	9	10
Silhouette	K-Modes	0.51041	0.62692	0.74839	0.83246	0.86337	0.89547	0.91719	0.93378	0.94477
	K-Means	0.51041	0.63719	0.75582	0.83187	0.85742	0.89011	0.92311	0.93378	0.94477
	Bisecting K-M	0.51041	0.62692	0.74839	0.83187	0.85249	0.88470	0.91769	0.92818	0.93916
Davies-Bouldin	K-Modes	0.99031	0.72865	0.57240	0.44065	0.55540	0.30329	0.26854	0.42351	0.35276
	K-Means	0.99031	0.76788	0.56810	0.44065	0.57144	0.53210	0.496031	0.47157	0.45085
	Bisecting K-M	0.99031	0.72865	0.51847	0.44065	0.48587	0.44638	0.40857	0.39125	0.36373
Calinski-Harabasz	K-Modes	53.15131	61.80752	89.00905	108.02351	112.56054	136.68640	188.80219	229.19077	410.62323
	K-Means	53.15131	64.04652	89.00905	108.02351	120.14788	142.99721	188.80219	236.24797	358.17032
	Bisecting K-M	53.15131	61.80752	85.01104	108.02351	120.14788	142.99721	188.80219	236.24797	358.17032
Hamming	K-Modes	1.05004	1.22500	1.71842	2.763841	4.89381	3.66332	17.74476	21.52334	79.35819
	K-Means	1.05004	1.29646	2.04664	2.90709	4.84193	9.29840	13.05861	25.62228	78.78602
	Bisecting K-M	1.05004	1.22500	1.60731	2.76384	4.87650	9.36862	13.15841	25.81877	79.35819

Table A.2: Score of different clustering algorithm using SPEC feature selection technique

Metric	Clustering method	2	3	4	5	6	7	8	9	10
Silhouette	K-Modes	0.3853	0.4952	0.5693	0.6501	0.7056	0.7428	0.8028	0.8222	0.8665
	K-Means	0.2887	0.3215	0.3478	0.3798	0.4161	0.4337	0.4699	0.4804	0.5072
	Bisecting K-M	4080	0.5001	0.5524	0.6501	0.7175	0.7428	0.7619	0.8130	0.8671
Davies-Bouldin	K-Modes	1.3362	1.0740	0.9092	0.7873	0.6078	0.6327	0.5273	0.4490	0.5712
	K-Means	1.0919	1.0740	0.8583	0.7873	0.6877	0.5274	0.6791	0.6381	0.5949
	Bisecting K-M	1.0919	1.0740	0.8593	0.6983	0.6480	0.7426	0.7508	0.6471	0.5937
Calinski-Harabasz	K-Modes	44.3825	53.5705	55.4991	60.1826	55.9202	73.0319	78.9261	81.2773	115.9777
	K-Means	45.8803	53.5705	56.2231	61.7225	66.43293	74.9521	88.1190	102.4118	119.8324
	Bisecting K-M	45.8803	53.5705	52.7435	60.1826	65.6401	66.4327	70.5722	87.3929	115.1574
Hamming	K-Modes	1.1574	1.4759	1.5862	1.9006	2.0452	2.8670	2.1490	4.8123	3.9703
	K-Means	1.0069	1.4759	1.6365	2.0467	2.6760	2.8270	5.9630	7.6933	11.6571
	Bisecting K-M	1.0069	1.4759	1.5542	1.7683	2.3266	3.2284	5.2178	6.3570	9.2303

Table A.3: Score of different clustering algorithm using USFSM technique

Metric	Clustering method	2	3	4	5	6	7	8	9	10
Silhouette	K-Modes	0.4432	0.5561	0.6565	0.7175	0.7740	0.8096	0.8599	0.9132	0.9343
	K-Means	0.4479	0.5561	0.6529	0.7283	0.7737	0.8222	0.8745	0.9106	0.9340
	Bisecting K-M	.0.4479	0.5561	0.6529	0.6908	0.7345	0.8238	0.8745	0.9106	0.9202
Davies-Bouldin	K-Modes	1.3179	1.4681	1.3300	1.3259	1.2293	1.1923	1.1033	1.0471	0.9388
	K-Means	1.46303	1.4526	1.3228	1.2299	1.1081	1.1826	1.10874	1.0751	0.9916
	Bisecting K-M	1.4630	1.3686	1.4020	1.2889	1.2614	1.1771	1.1262	1.1042	1.0482
Calinski-Harabasz	K-Modes	35.6571	48.7013	54.77123	67.3763	69.2473	90.0276	114.8120	153.8437	193.6991
	K-Means	36.2184	48.7013	57.5941	67.1416	78.6761	92.6317	114.8120	156.4843	216.8311
	Bisecting K-M	36.2184	48.7013	57.5941	57.0823	64.4443	90.7979	113.6749	154.2158	185.1597
Hamming	K-Modes	1.2217	0.9312	0.6893	0.6319	0.6591	0.5540	0.3399	0.4856	0.3786
	K-Means	1.0561	0.9312	0.6893	0.6620	0.6514	0.5540	0.4320	0.4074	0.2848
	Bisecting K-M	1.0561	0.9312	0.6893	0.6147	0.5815	0.5225	0.4155	0.3928	0.3743

Table A.4: Score of different clustering algorithm using the MCFS technique

Metric	Clustering method	2	3	4	5	6	7	8	9	10
Silhouette	K-Modes	0.3117	0.2943	0.3163	0.3660	0.3974	0.4152	0.4456	0.4522	0.5029
	K-Means	0.3015	0.2987	0.3279	0.3612	0.4039	0.4371	0.4729	0.4870	0.4947
	Bisecting K-M	.0.3015	0.2812	0.2981	0.3453	0.3641	0.3971	0.4280	0.4497	0.4764
Davies-Bouldin	K-Modes	1.3179	1.4681	1.3300	1.3259	1.2293	1.1923	1.1033	1.0471	0.9388
	K-Means	1.4630	1.4526	1.3228	1.2299	1.1081	1.1826	1.1087	1.0751	0.9916
	Bisecting K-M	1.4630	1.3686	1.4020	1.2889	1.2614	1.1771	1.1262	1.1042	1.0482
Calinski-Harabasz	K-Modes	35.2675	33.0325	31.2457	29.0208	29.7156	28.4406	28.4210	28.3048	28.6851
	K-Means	37.5025	33.9402	31.9704	30.8321	30.7310	30.0070	30.2896	30.6643	31.1613
	Bisecting K-M	37.5025	31.1900	29.4861	27.8435	27.1648	26.4411	26.3766	26.7326	27.0031
Hamming	K-Modes	1.0444	1.4817	1.6126	1.6895	1.8267	2.4891	2.7143	2.4634	2.8008
	K-Means	1.2175	1.56657	1.7711	1.8865	1.9233	2.4980	3.1908	3.3411	3.6763
	Bisecting K-M	1.2175	1.3083	1.6610	1.7669	1.9553	2.0897	2.3724	2.6887	3.0446

Table A.5: Score of different clustering algorithm using Low Variance feature selection technique

Bibliography

[1] B. De Moor A. Daemen. "Development of a kernel function for clinical data, in: Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society". In: *IEEE* (2009).

[2] Angèle CHristin et al. "Courts and Predictive Algorithms". In: (2015).

[3] Anil Chaturvedi et al. "K-modes Clustering". In: (2001).

[4] Axel Gautier et al. "AI algorithms, price discrimination and collusion: a technological, economic and legal perspective". In: (2020).

[5] B. Efron T. et al. "Least angle regression". In: *Annals of Statistics* (2004).

[6] Lars Buitinck et al. *API design for machine learning software: experiences from the*. 2013. URL: https://scikit-learn.org/stable/.

[7] Maxime C. Cohen et al. "The Use of AI in Legal Systems: Determining Independent Contractor vs. Employee Status". In: *SSRN* (2022).

[8] Tushar Kansal et al. "Customer Segmentation using K-means Clustering". In: *IEEE* (2019).

[9] Ujjwal Maulik et al. "Performance Evaluation of Some Clustering Algorithms and Validity Indices". In: (2002).

[10] Utsav Sharma et al. "Analysis of Customer Segmentation Clustering Techniques". In: (2022).

[11] Yawei Ahmadi et al. "Large-scale k-means clustering via variance reduction". In: (2018).

[12] Yiheng Chen et al. "The Comparison of SOM and K-means for Text Clustering". In: (2010).

[13] Liu Hd Alelyani S Tang J. "Feature selection for clustering: a review". In: (2013).

[14] Adrienne Brackey. "Analysis of Racial Bias in Northpointe's COMPAS Algorithm". In: (2019).

[15] Danny Butvinik. *Feature Selection – All You Ever Wanted To Know*. URL: https://www.kdnuggets.com/2021/06/feature-selection-overview.html.

[16] Tess et al Cersonsky. In: (2022).

[17] *Cluster with Self-Organizing Map Neural Network*. 2022. URL: https://nl.mathworks.com/help/deeplearning/ug/cluster-with-self-organizing-map-neural-network.html;jsessionid=f3e328c76950d1f017384120cf53.

[18] Mark Coeckelbergh. "AI ethics". In: (2020).

[19] *Competition Law*. URL: https://en.wikipedia.org/wiki/Competition_law.

[20] Yadong Cui. "Artificial Intelligence and Judicial Modernization". In: *Shanghai People's Publishing House* (2020).

[21] Mahyuddin Daud. "ARTIFICIAL INTELLIGENCE IN THE MALAYSIAN LEGAL SYSTEM: ISSUES, CHALLENGES AND WAY FORWARD". In: *INSAF - THE JOURNAL OF THE MALAYSIAN BAR* (2022).

[22] Bernard Desgraupes. “Clustering Indices”. In: (2017).

[23] Afrizal Firdaus. *Bisecting K-Means Clustering*. 2020. URL: https://medium.com/@afrizalfir/bisecting-kmeans-clustering-5bc17603b8a2.

[24] Avi Goldfarb and Florenta Teodoridis. *Why is AI adoption in health care lagging?* URL: https://www.brookings.edu/research/why-is-ai-adoption-in-health-care-lagging/.

[25] Deng Cai Chiyuan Zhang Xiaofei He. “Unsupervised Feature Selection for Multi-Cluster Data”. In: *State Key Lab of CADCG* (2010).

[26] Giovanna Massarotto Ashwin Ittoo. “Gleaning Insight from Antitrust Cases Using Machine Learning”. In: *Standford Computationnal Antitrust* (2021).

[27] Philip Treleaven Jeremy Barnett. “Algorithmic Dispute Resolution—The Automation of Professional Dispute Resolution Using AI and Blockchain Technologies”. In: *The Computed Journal* (2018).

[28] Shehroz S Khan and Shri Kant. “Computation of Initial K-modes Clustering Algorithm using Evidence Accumulation”. In: ().

[29] Ajitesh Kumar. “Correlation Concepts, Matrix Heatmap using Seaborn”. In: (2022).

[30] Dhairya Kumar. *Introduction to Data Preprocessing in Machine Learning*. URL: https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d.

[31] Walsh Leila. “More than 20 cities and states adopt risk assessment tool to help judges decide which defendants to detain prior to trial”. In: *Laura and John Arnold foundation* (2015).

[32] Jundong Li et al. “Feature selection: A data perspective”. In: *ACM Computing Surveys (CSUR)* (2018).

[33] Motoda H Liu H. “Feature selection for knowledge discovery and data mining”. In: *IEEE Sensors Journal* (1998).

[34] Ulrike von Luxburgd. “A tutorial on spectral clustering”. In: (2007).

[35] *Machine Intelligence - Lecture 7 (Clustering, k-means, SOM)*. 2019. URL: https://www.youtube.com/watch?v=lFbxTlD5R98.

[36] Jyoti Prakash Maheswari. *Breaking the curse of small datasets in Machine Learning*. URL: https://towardsdatascience.com/breaking-the-curse-of-small-datasets-in-machine-learning-part-1-36f28b0c044d.

[37] Md. Abdul Malek. “Criminal courts’ artificial intelligence: the way it reinforces bias and discrimination”. In: (2022).

[38] *Manipulation des données avec Pandas*. URL: https://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/fr_Tanagra_Data_Manipulation_Pandas.pdf.

[39] Saúl Solorio-Fernández1 · J. Ariel Carrasco-Oca1 · José Fco. Martínez-Trinidad. “A review of unsupervised feature selection methods”. In: (2019).

[40] Giovanna Massarotto. “Antitrust Settlements : How a simple agreement can drive the economy”. In: (2019).

[41] Giovanna Massarotto. *Giovanna Massarotto*. URL: https://giovannamassarotto.com/bio/.

[42] Charles Nicholas and Ketan Rajshekhar Shahapure. “Cluster Quality Analysis Using Silhouette Score”. In: *IEEE* (2020).

[43] Xiaofei He Deng Cai Partha Niyogi. "Laplacian Score for Feature Selection". In: (2005).

[44] Nicolas Petit. "Antitrust and Artificial Intelligence: A Research Agenda". In: (2017).

[45] Nikita Pilnenskiy and Ivan Smetanniko. "Feature Selection Algorithms as One of the PythonData Analytical Tools". In: *Future Internet* (2020).

[46] Novianti Puspitasari. "Customer segmentation using bisecting k-means algorithm based on recency, frequency, and monetary (RFM) model". In: (2020).

[47] C. Kamath R. Ponmalai. "Self-Organizing Maps and Their Applications to Data Analysis". In: (2019).

[48] rileypsmith. *sklearn-som 1.1.0*. 2021. URL: https://pypi.org/project/sklearn-som/.

[49] Darío García-García; Raúl Santos-Rodríguez. "Spectral Clustering and Feature Selection for Microarray Data". In: *IEEE* (2009).

[50] Tirthajyoti Sarkar. *Clustering metrics better than the elbow-method*. 2019. URL: https://towardsdatascience.com/clustering-metrics-better-than-the-elbow-method-6926e1f723a6.

[51] J. Ariel Carrasco-Ochoa Saúl Solorio-Fernández JoséFco. Martínez-Trinidad. "A new Unsupervised Spectral Feature Selection Method for mixed data: A filter approach". In: *IEEE Sensors Journal* (2017).

[52] *Selecting the number of clusters with silhouette analysis on KMeans clustering*. URL: https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html#sphx-glr-auto-examples-cluster-plot-kmeans-silhouette-analysis-py.

[53] sharadarao1999. *Bisecting K-Means Algorithm*. 2020. URL: https://www.geeksforgeeks.org/bisecting-k-means-algorithm-introduction/.

[54] Oskar Skoglung. "Finding co-workers with similar competencies through data clustering". In: (2022).

[55] Diego Unzueta. *Unsupervised Learning: K-Means Clustering*. 2022. URL: https://towardsdatascience.com/unsupervised-learning-k-means-clustering-6fd72393573c.

[56] Nelis J. de Vos. *kmodes categorical clustering library*. URL: https://github.com/nicodv/kmodes.

[57] Michael Waskom. *seaborn: statistical data visualization*. URL: https://seaborn.pydata.org.

[58] Michael Waskom. *Self-Organizing Maps*. 2018. URL: https://medium.com/@abhinavr8/self-organizing-maps-ff5853a118d4.

[59] Huan Liu Zheng Zhao. "Spectral Feature Selection for Supervised and Unsupervised Learning". In: *IEEE Sensors Journal* (2007).