UNIVERSIDAD INTERNACIONAL DE LA RIOJA

Universidad Internacional de la Rioja (UNIR)

Escuela Superior de Ingeniería y Tecnología

Master Degree in Massive Data Analytics & Visualization

Big Data and Business Strategy, a Proof of Concept on Toolset Integration

Master Degree Thesis

Presented by: Rodríguez Lago, Odín

Director: Mora Pérez, Dr. Javier

City: Madrid
Date: September 2019

Contents

Abstract

This thesis analyzes the coverage in the academic literature of the integration of Business Strategy techniques and Big Data techniques. The relationships between both fields are also analyzed, and a methodology to integrate them is proposed. Finally, a proof of concept using the proposed methodology is implemented. The systematic search run found one academic publication relevant to the subject of this thesis. The analysis performed regarding both fields, based on several Porter's Monitor Group bankruptcy analyses, identifies two issues with the Business Strategy techniques propitious to be mitigated with Big Data techniques. First, the lack of, or a weak, domain knowledge behind the business strategy techniques used. Second, the inherent cognitive bias introduced by human judgment. Results from the proof of concept implementation show that, following such methodology, a data-driven approach enriches and enhances the outcomes of a business strategy technique.

Keywords: DDSD, Strategic Analysis, Business Strategy, Big Data, Machine Learning, Porter's Five Forces, Decission Support System, Cognitive Bias, Domain Knowledge, Data-Driven

Resumen

Esta tesis analiza la cobertura en la literatura académica de la integración de las técnicas de Estrategia Empresarial y las técnicas de Big Data. Se analizan también las relaciones existentes entre ambos campos, y se propone una metodología para integrarlos. La búsqueda sistemática realizada muestra una única publicación académica relevante al tema de esta tesis. Un análisis de ambos campos, basada en varios análisis de la bancarrota de la compañía "Monitor Group" de Porter, identifica dos problemas principales en las técnicas de estrategia empresarial, susceptibles de ser mitigadas por técnicas de Big Data. Primero, la falta de conocimiento de dominio, o un conocimiento débil, asociado a las técnicas de estrategia empresarial utilizadas. En segundo lugar, el sesgo cognitivo inherente introducido por el juicio humano. Los resultados muestran que, siguiendo dicha metodología, un enfoque basado en datos enriquece y mejora los resultados de una técnica de Estrategia Empresarial.

Palabras clave: DDSD, Análisis Estratégico, Estrategia Empresarial, Big Data, Machine Learning, Cinco Fuerzas de Porter, Sistemas de Apoyo a la toma de Decisiones, Sesgo Cognitivo, Conocimiento de Domino, Orientación a Datos

List of Figures

List of Tables

Chapter 1

Introduction

The main topic of this thesis is the limited coverage of the integration of Business Strategy and Big Data techniques in the academic literature. Publications regarding Big Data and Machine Learning are exponentially increasing every year as shown in figure 1; though, the number of publications showing how to integrate Big Data and Business Strategy techniques is meager.

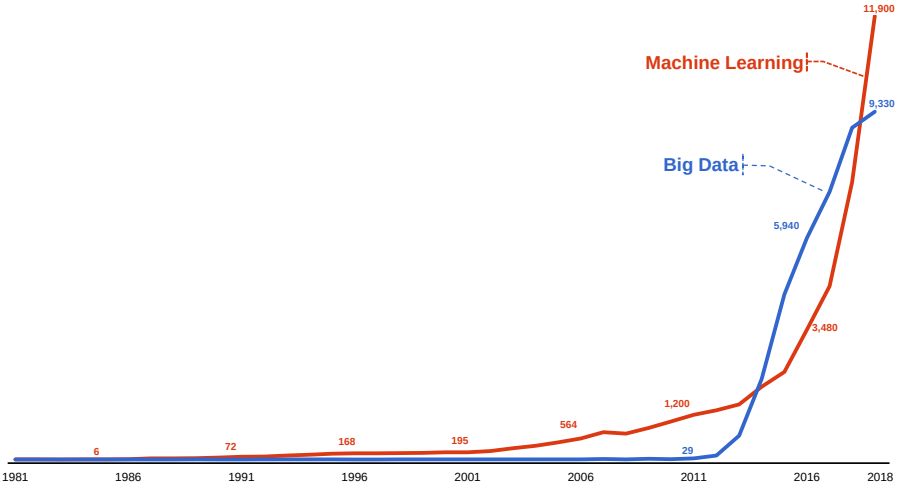


Figure 1: Number of published articles on Machine Learning an Big Data, between 1980 and 2019. Searches containing “Big data” or “Machine learning” in the title of the publication; search engine Google Scholar (Sept. 2019).

In this context, this thesis proposes a methodology to fill this gap; to enrich Business Strategy with Big Data and Machine Learning (section 2.3). A practical example is also developed, following the proposed methodology and within the context of a strategic external analysis.

A second topic, is the aim to improve Business Strategy techniques with a data-driven approach. Indeed, there is a proven causal relationship between data-driven management practices and better performance, as shown in the first large-scale survey, Management and Organizational Practices Survey (MOPS), conducted by U.S. Census Bureau on 2013 (Bloom et al., 2013). Further studies on the same data show a causal relationship between data-driven

decisions and performance increase (Brynjolfsson & McElheran, 2016).

Notice that, since the terms Big Data, Machine Learning, Data Analytics, and Business Intelligence are intertangled and, sometimes interchanged; this thesis uses Big Data as an overarching term for these fields.

The assessment of the feasibility of the integration of both fields is done through two processes. The first one is the analysis of the integration of both fields and a proposal of a novel methodology to integrate both fields (chapter 2). The second process is a Proof of Concept (PoC) based on the proposed methodology. The PoC is developed and documented through the rest of the chapters, following the same sequence and the phases defined in the proposed methodology.

Therefore, the purpose of this thesis is two-fold. First, it assesses the coverage, in the academic literature, of the integration of Business Strategy and Big Data fields. Secondly, it assesses the feasibility of such integration between both fields. Such assessment is done by setting up the following business context: "We are a company in the bar industry, and we want to open a new bar in a specific region, Madrid. Within this region, we want to select a proper location for our business". During the document, the strategic external analysis technique "Porter's Five Forces" is analyzed regarding its integration with Big Data techniques. Such analysis produces a specific strategic goal: "Which location does maximize our chances of success?". The strategic external technique is based on the famous supply-demand model, that acts as the domain knowledge for the "feature selection" phase. This model is integrated with Big Data by proposing two different data tasks: one clustering task, and one classification task. These two tasks produce two data models addressing the initial question posed by the strategic external analysis (which location to select).

Results show that, following the proposed methodology, a data-driven approach enriches and enhances the outcomes of a business strategy technique. It also enhances the technique by providing new multiple different perspectives based on the hidden structure of the data analyzed. Finally, a data-driven approach reduces the uncertainty of business strategy techniques by producing results relevant to the subject analyzed, and assurance supported by the validation of the data models.

Also, during the implementation of the proof of concept, it has been clear that the lack of data, and its quality, are burdens for the proper implementation of Big Data at this level of abstraction. Indeed, it is the most consuming task (data mining). That is a common issue, (Vidgen, Shaw, & Grant, 2017). This limitation is also expected to be less critical as access, and quality of open data sources increase over time.

1.1 Document Structure

This thesis contains, apart from this chapter, three introductory chapters, four chapters implementing the proposed methodology, and two final chapters with results and conclusions:

- Objectives and Methodology. In this chapter, the steps needed to achieve and validate the main purpose of the thesis are described.
- State of the art. In this chapter, the coverage of Big Data and Business Strategy integration is shown. It contains a state of the art section analyzing the coverage of academic publications regarding the integration of both fields.
- Data Driven Strategy Development. A novel methodology, Data-Driven Strategy Development (figure 3), is proposed.
- DDSD Phase One: Strategy Analysis. In this chapter, a Business Strategy technique ("Porter's Five Forces") is analyzed regarding its scope and applicability to Big Data. A business context is built, the location of a new store, and some business objectives are developed based on the external analysisi (Porter's Five Forces).
- DDSD Phase Two: Feature Engineering. This chapter integrates Big Data techniques by analyzig the outcome of the initiation of the external analyisis. Different "meta-features", and tasks will be defined in order to enhance and analyze the defined business objectives.
- DDSD Phase Three: Data Mining & Preparation. In this chapter, based on the meta-features defined, different data sources are analyzed, and several data-sets are proposed for the defined tasks. It also contains the preparation of the data-sets for the models.
- DDSD Phase Four: Model Implementation & Evaluation. In this chapter, multiple models are developed and evaluated. These models are the implementation of the data tasks defined in the featuring engineering phase.
- DDSD assessment. In this chapter, based on the specific goals of the thesis, and with the outcome of the proof of concecpt, the DDSD methodology is assessed.
- Conclusions and future work. This chapter assesses the fitness of the methodology proposed. The assessment is based on the results obtained during the implementation of the PoC; and also on the applicatibility to the business strategy activity selected.

Chapter 2

Objectives and Methodology

In this chapter, the steps needed to achieve and validate the main purpose of the thesis are described. A novel methodology, called Data-Driven Strategy Development (see figure 3), is proposed.

2.1 General Objectives

To fulfill the primary purpose of this thesis, a systematic search is run. This will show the the level of coverage of this thesis subject. To suuport and develop the second goal, there will be a Proof of Concept (PoC) of a selected strategy methodology followed by an assessment of the results. This thesis is a practical guide through the steps needed to empower a strategic activity with practical data analysis. In this respect, there are two questions this thesis answers:

1. “Are Big Data techniques applicable at a strategic level?”, moreover *“Is there any value applying Big Data techniques to strategic methodologies?”*.
2. *“How?”*.

2.2 Specific Objectives

The hypothesis to be proven is that Big Data techniques can enhance Business Strategy techniques. To prove that, an after building a Proof of Concept, a qualitative assessment is conducted to verify the added value over a traditional approach; this assessment is based on the fullfilment fo several objectives:

- To implement an algorithm, or several, answering the questions proposed by the selected strategic methodology.

- To obtain comparable results with the results coming from a classical approach.
- To enhance the strategic methodology
- To reduce the risk, or uncertainty, associated with traditional strategic techniques.

2.3 Work Methodology

This thesis follows several steps to fulfill the general and specific goals. Since there are two main different goals, the thesis can be divided in two parts.

The first part of this thesis, chapter 3 and 4, deals with the theoretical understanding of the subject and its coverage in the academic literature. In this part, the following steps are followed:

- Business Strategy and Big Data analysis: In this step, a brief analysis of both fields is performed. This step introduces the issues in Business Strategy to be addressed by Big Data.
- State of the Art: in this step, a systematic search is run to analyze the coverage in the academic literature of the subject of this thesis: the integration of Business Strategy techniques and Big data Techniques.
- Integration Methodology: after the systematic search, a novel methodology to integrate both fields is developed: “Data Driven Strategy Development” (DDSD). This methodology is used during the practical part of the thesis.

The second part of this thesis, chapter 5 to 8, deals with the practical integration of both fields, following the proposed methodology (DDSD):

- Business context: the first step is to build a business context and initiate a strategic technique to be integrated with Big Data. The business context is the selection of a proper location for a bar in the city town of Madrid. The selected strategic technique is “Porter’s Five Forces”, specifically the “industry rivalry”.
- Data Driven Integration: this step implies to include Big Data techniques into the already initiated strategic technique. This step produces a set of data tasks with different data models, rending specific results answering the issue addressed by the strategic technique (which location is best?)

Chapter 3

State of the art

This chapter introduces the relationship between Big Data and Business Strategy; presenting the first one as mitigation to some of the second one's issues. It will also present a brief State of the Art regarding Big Data as a strategy analysis tool, giving a perspective on how Big Data is used today when it comes to strategy.

3.1 Business Strategy and Big Data

Business strategy is a complex activity, and Porter's Monitor Group bankruptcy in 2012 (Reuters, 2012) is a clear proof of it. One critique, relevant to this thesis, and shared in further analyses of Porter's Monitor Group bankruptcy is the lack of logic (L.G., 2012) (Denning, 2012). A second relevant critique is an excessive focus on financial aspects rather than on customers: *"the value proposition of a supposed sustainable competitive advantage achieved by studying the numbers and the existing structure of the industry became increasingly implausible and irrelevant"* (Denning, 2012).

The lack of logic supporting a model is a common reason for failure, not only in complex systems as corporations but also in the data field. This "logic", domain knowledge as per Armstrong (2001), or theoretical understanding as per Silver (2012), is a key element in any model[1]. As an example, in economics, the increase of data and computer power (Big Data era) "*did not cover for the lack of theoretical understanding about the economy; it only gave economists faster and more elaborate ways to mistake noise for signal*" (Silver, 2012). In complex systems, like corporations and markets, without a theoretical understanding of the field, most likely any pattern found will be noise.

[1] Notice that, strategy implies to predict future events based on incomplete information. Its activities involve the creation of models and forecasting events based on these models.

The corporation-centric approach, instead of a more customer-oriented approach, can be seen as a scope issue due to intrinsic biases. Indeed, it is not trivial to include this "variable" (customers) into a strategy model and try to understand the essential rules that govern their relationships with other variables, i.e., building a theoretical understanding. It might be not possible. Customers' behavior, or any human behavior, in fact, is not governed by pure logic (Akerlof & Shiller, 2010), (Thaler & Ganser, 2015), or (Kahneman, 2011). Though, Big Data can prove insight based on the structure of the data, more than the reasons behind that structure.

This "irrational behavior", a bias, goes beyond the role of customers in a strategy model, it affects the whole business. The deficiency in Porter's models and the fact that they were used for decades are proofs of this "irrational behavior". A famous example is an article by Arnott, Hsu, Kalesnik, and Tindall (2013), where inverse stock portfolio strategies yield same or better results than the original ones, and worst, the random stock selection also yields same results (the famous monkeys throwing darts at Wall Street Journal). Another example focused on strategy decisions is presented by Cooper, Gulen, and Rau (2016). There, it is proven that there is a negative relationship between CEO compensations and future performance (stocks and operational) of a firm.

So, Big Data seems a prospective field for decision making in companies. Indeed, Big Data plays a strategic role in companies (Woerner & Wixom, 2015); but many fail to do so (Meulen, 2016) (Gerdeman, 2017) (Appelbaum, Kogan, Vasarhelyi, & Yan, 2017). The main three challenges, in decreasing importance, companies face when incorporating data-driven processes at management level are (Vidgen, Shaw, & Grant, 2017):

1. Data quality.
2. Big Data and decision making alignment.
3. Big Data and business strategy alignment.

Data quality is the principal concern. Data processing relies on a minimum quality of the data; this issue is present in any automated process and, in Big Data, the impact is even higher (Appelbaum, Kogan, Vasarhelyi, & Yan, 2017). Data shortage, related to data quality, is an added complication. Investments on company-wide digitalization is still low (Weill & Woerner, 2013) (2013), and at a strategic level, data availability is even worst (Creamer & Freund, 2010). So it is arduous, if not impracticable in some cases, to find such data, limiting the applicability of Big Data to strategic techniques. This issue is the principal reason for the loop between phase two and three in DDSD methodology.

It is very indicative that the second and third concerns are related to Big Data alignment at

an operational and strategic level. This shows how difficult is to get meaning out of the data (Meulen, 2016), (Gerdeman, 2017), (Schmarzo, 2017). This is aligned with Armstrong (2001) and Silver (2012), the lack of logic behind any business or strategy model is a real issue.

To see how Big Data techniques can be applied to Business Strategy, it is needed to know the different types of Business Strategy techniques. Based on the type of process they are used for, and regarding Navas López and Guerras Martín (2016), Business Strategy can be organized in three main phases with different activities:

1. Strategy Analysis: apart from the definition of business mission and business goals, in this phase, internal and external analysis activities are developed.
2. Strategy Design: based on the strategic analysis, a strategic plan is developed, giving a set of strategic objectives to monitor and implement.
3. Strategy Implementation and control: in this phase, different processes are defined to achieve strategic objectives and monitor their performance.

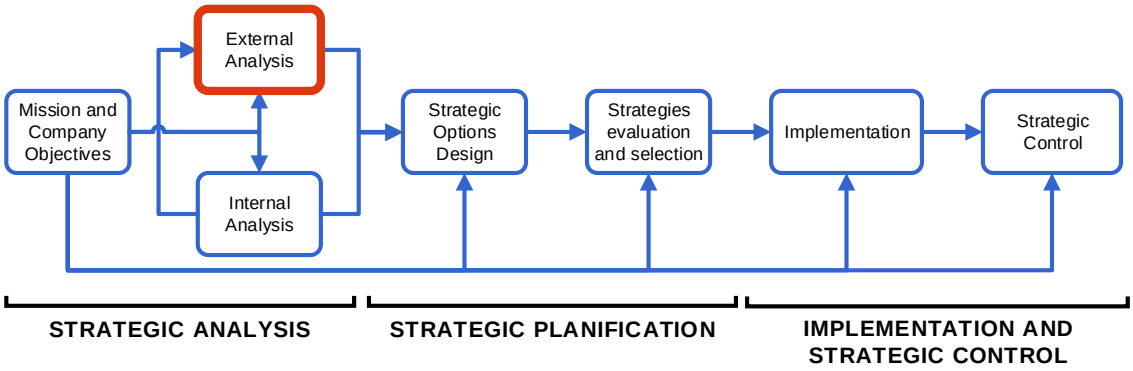


Figure 2: Business strategy activities (Navas López & Guerras Martín, 2016).

The term “Business Strategy techniques” spans multiple keywords, in fact, regarding Cadle, Paul, and Turner (2010), about a hundred different techniques across all the strategic phases. Cadle, Paul, and Turner place some Business Techniques into four specific strategic phases:

1. Strategic Analysis (internal and external): PESTEL (Political, Economic, Social, Technological, Legal and Environment) analysis, and Porter’s Five Forces.
2. Strategic Definition: SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis, and Andsoff’s Matrix.
3. Strategic Implementation: POPIT (People, Organisation, Processes and Information technology) model, Leavitt’s Diamond, McKinsey 7-S.

4. Strategic Monitoring: Balanced Score Card, Critical Success Factors (CFS), and Key Performance Indicators.

Notice that,regarding Navas López and Guerras Martín, Strategic Analysis and Strategic Definition fall inside the same phase: Strategic Analysis. Also, Strategic Planification does not entail any business strategy tool since it is not considered by Cadle, Paul, and Turner. To include a Business Strategy technique in this phase, the Strategy Map tool (Balanced Scorecard tool) is used.

Therefore, Business Strategy techniques can be mapped with the strategic phases as follows:

1. Strategic Analysis: PESTEL, Porter's Five Forces, SWOT analysis, and Andsoff's Matrix.
2. Strategic Planification: Strategy Map.
3. Strategic Implementation and Control: POPIT model, Leavitt's Diamond, McKinsey 7-S, Balanced Score Card, Critical Success Factors (CFS), and Key Performance Indicators.

Big Data techniques can be integrated into any of these three phases, especially in the most operational ones "Implementation and Strategic Control" ((LaValle, Lesser, Shockley, Hopkins, & Kruschwitz, 2011)). The scope of this thesis is the phase "Strategic Analysis"; this is where the strategy is built. The proof of concept focuses on its inner activity "External Analysis". This activity evaluates the business environment. Similar techniques could be applied in the internal analysis activity.

3.2 Field Publications

This section shows a state-of-the-art study based on Kitchenham's "systematic search guideline" (Barbara & Charters, 2007). Search results show that just a few publications are addressing the subject of this thesis: to conjugate both Big Data and Business Strategy techniques. The question to formulate the queries that will render publications on the subject of this thesis is:

How do we integrate Big Data into Business Strategy techniques?

The expected results should form a list of publications, for any year, describing how to leverage Big Data to one or several Business Strategy techniques with implementation details. As commented, in this thesis, Big Data also entails the Machine Learning field; therefore, both terms will be keywords in the searching string. The Strategic Business techniques presented earlier will be used to perform a systematic search.

The intent is to search the meta-data of the publications, so yielded results are relevant to the subject, i.e., title, keywords, and abstract. The main academic search engine used is Google Scholar since, currently, it is the most comprehensive academic search engine (Gusenbauer, 2019). Also, Elsevier and IEEE Xplore have been selected to complement the leading search engine. Google Scholar offers the option of looking into titles and full publication. Elsevier does offer searches in three meta-fields with the following operators: TITLE, ABSTR, and KEY. IEEE Xplore offers searches on all metadata fields (including authors names), but it also adds keywords on top of the author keywords list. All queries of the systematic search are detailed in appendix A.

In order to get only relevant publications, the following rules will be applied to all results:

- Publication must be written in English (or Spanish).
- Publication subject must be Business Strategy.
- Publication subject must be Big Data or Machine Learning.
- Publication must address one or several Business Strategy techniques (directly or indirectly mentioned)
- Publication must address the application or integration of Big Data or Machine Learning to the subject treated practically.

Results of all searches yield a hundred fourteen results (table 1). After filtering those publications with the above rules, there is a total of ten publications relevant to the subject of this thesis (two of them repeated):

	Google Scholar		IEExplore		Elsevier		Total
	Results	Filtered	Results	Filtered	Results	Filtered	Filtered
PESTEL	0	0	0	0	0	0	0
Porter	0	0	33	**1**	1	0	**1**
SWOT	7	0	6	**1**	1	0	**1**
Andoff	0	0	0	0	0	0	0
Strategy Map	0	0	1	0	1	**1**	**1**
POPIT	0	0	0	0	0	0	0
Leavitt	0	0	3	0	0	0	0
McKinsey	1	0	4	0	1	0	0
Scorecard	3	**1**	4	0	4	**3**	**4**
CSF	1	0	3	0	2	0	0
KPI	0	0	17	0	18	**2**	**2**

Table 1: Systematic search results.

Table 1 shows one hundred fourteen results; only nine of them pass the rules described earlier. From the nine filtered results, there are not relevant or meaningful publications on “Strategic Analysis”, and not publications at all on “Strategic Planification” and “Strategic Implementation”. Regarding “Strategic Monitoring”, the publication by Kokina, Pachamanova, and Corbett (2017) is fully relevant, though, it is focused on the strategic control activity. I does not particicape in the conception of the strategy (strategic analysis). It is a practical exercise for students; it provides sales data of a company named “Bombas” and, based on metrics designed in a Balanced Scorecard, students must induct which data should be selected and visualized (through Tableau or Excel) to support the BSC board. This is an example of Big Data integrated into Busines Strategy processes, although in this case it does not identify or define goals or objectives; it supports them by data validation.

Chapter 4

Data Driven Strategy Development

This thesis proposes a novel methodology named “Data Driven Strategy Development” as a framework to integrate Business Strategy techniques and Big Data. The methodology is depicted in figure 3.

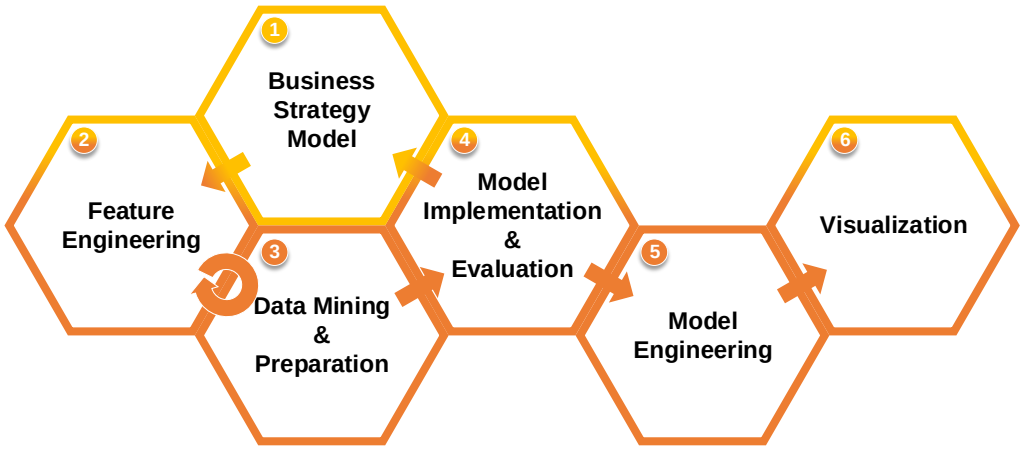


Figure 3: Data Driven Strategy Development. Yellow stands for business layer, orange stands for technical layer.

The proposed methodology, DDSD, is based on the well-known methodology “Cross-Industry Standard Process for Data Mining”, CRISP-DM (Wikipedia, 2019b), depicted in figure 4. The reason to propose a new data mining methodology is that CRISP-DM is not fit for the purpose of this thesis. CRISP-DM is focused on data produced by business processes; it is not meant to improve strategic techniques. It is close to a data-warehouse methodology like Kimball Lifecycle (Kimball, 1996). CRISP-DM starts by analyzing the business needs at an operational level, mainly focused on getting a better insight into the business processes for efficiency improvement. This thesis subject is not the enhancement of business processes but the strategy analysis itself.

A significant commonality between both methodologies is the “research” nature of both: it

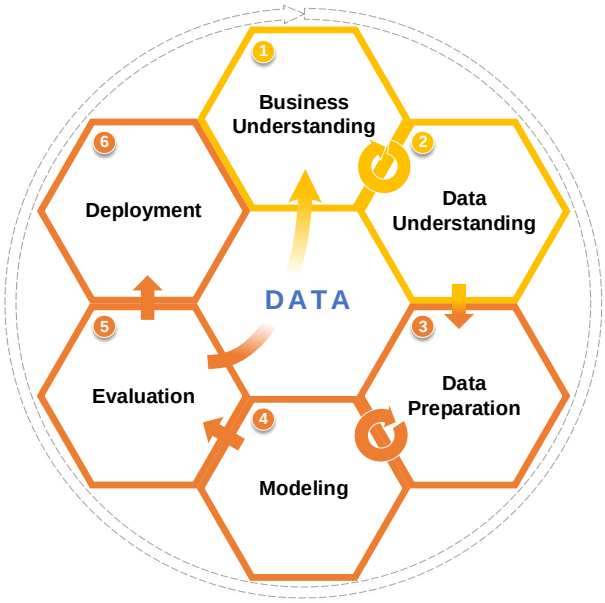


Figure 4: Scheme of the CRISP-DM methodology. Yellow stands for business layer, orange stands for technical layer.

is more a research process than an engineering one. Indeed, the loops between phases show the whole process in CRISP-DM, and the first four phases in DDSD, are highly iterative. Both methodologies use future phases as feedbacks for previous ones. Indeed, these loops happen naturally in every data-driven process. As both diagrams show, there are loopbacks from the model evaluation to the beginning of the process. Though, this is another loop in CRISP-DM, while in DDSD, this phase sets the end fo the iterative process.

A notable difference, between both methodologies, is the first loop between the business field and the data field. In CRISP-DM, business questions will define the data to be processed, but the understanding of this data might trigger a more in-depth insight into the business processes. This loop does not happen in DDSD; the business questions in DDSD are at a higher level of abstraction that the ones in CRISP-DM. The later ones are based on business processes (with data assets already defined). In DDSD, the transition between phase one and phase two implies to transform abstract questions to data questions; there is not a meaningful loopback.

Then, the scope on both methodologies differs. CRISP-DM is centered in the corporations' processes data, while DDSD is open to any external data source. DDSD starts by analyzing the strategy (methodology) without focusing on specific data sources, whereas CRISP-DM, starts with business processes and focuses on data produced by such processes. So there is a clear different starting point: CRISP-DM focuses on improving business processes, and DDSD focuses on improving a strategy methodology. In CRISP-DM phase number two, the data is used based on the analysis of the company's processes and the data they produce. Notice here the level of abstraction is already low since the data-sets to work with are already

apparent. DDSD is heavily based on featuring selection. DDSD phases one and two, colored in yellow, are highly related to the business layer; notice also that DDSD does not mention the term data, yet.

The loop between DDSD phase two and three, not present in CRISP-DM, is another indication of the different scope and nature of both methodologies. Since the real data definition happens in phase three, it might be needed to come back to phase two to translate the business into a different type of data and tasks. This might be caused by the lack of data, its quality, or simply that it is too expensive to gather. Since CRISP-DM works on data produced by business processes, this loopback does not exist. In CRISP-DM, there is no uncertainty in the transition between both phases.

The CRISP-DM phase “Deployment” has been replaced with a new one called “Model Engineering”. As its name indicates, this is the engineering part of the process. In this phase, aspects like quality, performance, usability, and stability are central concerns.

The last DDSD phase, called “Visualization”, is not present in CRISP-DM since it is part of the deployment phase. Its principal goal is to communicate to business a solution to support decision making correctly. The most effective manner of doing so is through data visualization.

Finally, DDSD does not have an external loop, or continuity, between the last and first phase. There are two reasons for this. First, the methodology is meant to be embedded into a higher strategic framework that, when it is necessary or planned, triggers a strategic analysis that will include the utilization of a ”data-driven-enabled” business strategy tool. Second, the loop from phase four to phase one is where the validation, at business level, happens. In short, phase six is the realization of something already validated and ready to be used.

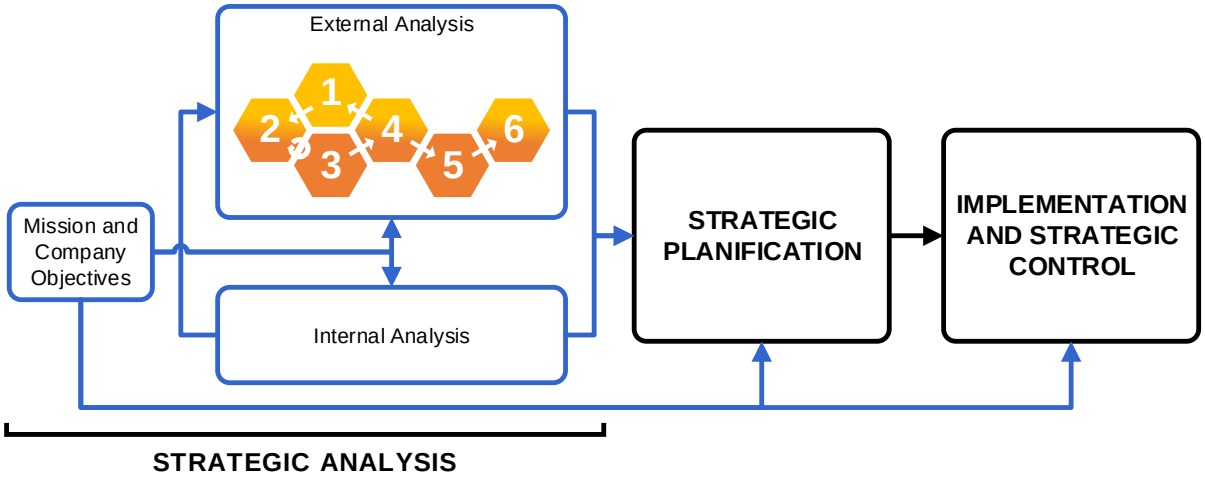


Figure 5: Implementation of the DDSD in this thesis within the business strategy activities; strategic activities from Navas López and Guerras Martín (2016).

As depicted in figure 5, and within the business strategy activities context (figure 2), this

thesis' PoC uses the DDSD methodology as a strategic external analysis technique (Porter's Five Forces).

This thesis follows the DDSD methodology with a chapter per phase, except for the last two ones. This is because the objective of those two phases is to produce a product. This is an academic work with the objective of validating a methodology; there is no goal of implementing a product. As already commented, such validation happens at the end of the DDSS phase four. The nex subsections describe each of the DDSD phases.

4.1 DDSD Phase 1

This is the business understanding phase. In this phase has three principal outcomes: the main strategy goal, the domain knowledge, and the "meta-features".

The strategic goal is the business context that triggers the whole strategy exercise. For this thesis, within the context of setting up a bar in Madrid, the gaol is "Which location does maximize our chances of success?".

The domain knowledge (Wikipedia, 2019c) is the cornerstone on which the Big Data techniques are applied. The strategy activity, external or internal, must be based on a strategy model. In this thesis, it is Porter's Five Forces. This model drives the development of the strategy, and all the Big Data techniques to be integrated are tailored to this model.

The "meta-features" are those characteristics of the strategy model that can be expressed as variables, and captures the logic of the strategy model. These variables, or "meta-features", are the glue between the strategy model and the data models to be developed in further phases.

4.2 DDSD Phase 2

This is the feature engineering (Wikipedia, 2019g) phase. As its name indicates, this is where the features (attributes) of the future machine learning models are defined; it also produces a set of machine learning tasks aligned with the strategic goals defined.

In this phase, the variables describing the logic behind the strategy model are translated into real features. Notice that the data source is not yet mentioned in this phase; though, characteristics of the data source, like availability and quality, might trigger loop-backs from the next phase to this one.

Once the features are defined, several machine learning tasks are defined based on the strategic goals defined. These tasks use the features defined to address the goals that the strategy model tries to achieve.

In this thesis two different tasks, based on machine learning algorithms, are proposed to address the goals of the strategy model; i.e., a clustering task that identifies locations of Madrid based on their demand-supply curve, and a classification task to validate the first task and, at the same time, unveil which characteristics of the locations are responsible for such behaviors.

It is essential in this phase, to consider the implications of data privacy based on the features and the tasks that have been defined; i.e., the general data protection regulation, GDPR (Parlament & Council, 2016). Whit this regards, this thesis does not gather any personal data at all so that the regulation can be disregarded. Personal data means, regarding Parlament and Council (2016), *[...] any information that relates to an identified or identifiable living individual. Different pieces of information, which collected together can lead to the identification of a particular person, also constitute personal data*.

All features (attributes) and related data-sets used in this thesis are related to aggregated public data of different wards of Madrid. Given this granularity, wards, it is nor possible to identify any person in these data-sets.

4.3 DDSD Phase 3

This is the data mining and preparation phase. In this phase, the data-sources containing the needed features, for the different tasks, are selected. Here the data-set, format, and content, are defined and also cleaned (prepared).

Although this is the lex complex phase in the whole process, it is the most time-consuming and, as during the development of the proof of concept shows, it is here where the two main handicaps of the integration between Business Strategy and Big Data appears: data quality and quantity.

4.4 DDSD Phase 4

This is the model implementation and evaluation phase. In this phase, the machine tasks defined earlier are implemented. That yields one or more data models per task.

Once the models are implemented, they are evaluated against the strategic goals defined at the beginning of the process. This evaluation can trigger another iteration of the first four phases, a refinement, or a production readiness planning (phase five and six).

4.5 DDSD Phase 5

This is the model and engineering phase. It implies the full automation of phase three, Data Mining and Preparation. This phase is the one that can be governed by a project management methodology, so a project is triggered to get a real product, mainly to produce an interface to the visualization phase. Notice that the research nature ends with phase four, the inherent uncertainty of previous phases is gone, and now any task could be governed as a pure ICT "project".

In this phase, if relevant, the security aspects related to the data-set must be revisited, especially those concerning data privacy. Techniques of automatic obfuscation and anonymization must put in place since, during previous phases, all the data manipulation is done by ad-hoc and manual processes.

4.6 DDSD Phase 6

This is the visualization phase. This phase has the responsibility to translate the outcome of the processed data faithfully, and the model's outcome in such manner, stakeholders fully understand the meaning of the outcome. The principal reason to decouple phases five and six is that the presentation should not depend on the data model implementation; indeed there might be different representations (views) of multiple models and combinations with existing models. This phase is located between the engineering (orange) and the business levels (yellow). Although it has to be implemented in a productive environment, it does not require strict project management (unlike a data model implementation).

Chapter 5

DDSD Phase One: Strategy Analysis

In the previous chapter, some business strategy techniques were categorized based on which strategy phase they are applied. This thesis focuses on the "Strategic Analysis" phase since it is the most abstract and closest to the concept of business strategy as such. Porter's five forces is an "External Analysis" technique within the "Strategy Analysis" phase (see section 3.1). It is the business strategy technique to be analyzed in this thesis, regarding its applicability to Big Data techniques. Within a business case, an answer to this thesis question *"How do we integrate Big Data into Business Strategy techniques?"* is given by developing an example.

Before applying a business strategy technique, a business goal, in the context of a business case is needed. Therefore, the following context is assumed: "*We are a company in the food and drink industry, and we want to open a new bar in a specific region, Madrid. Within this region, we want to select a proper location for our business*".

With this context, the goal is: *To find a location taht maximizes our chances of success*. This thesis uses Porters' Five forces to do an external analysis of the industry to achieve this goal.

5.1 Business Strategy Technique (Porter's Five Forces)

Porter's Five Forces (Porter, 1989) is an external analysis technique that belongs to the strategic analysis phase (see figure 2). It considers the context where the business is playing (or is going to play) identifying threats that might put at risk the business (the lack of threats would imply opportunities). It is more focused on the business domain than the general context, as PESTEL technique does, for instance. It defines five forces that negatively or positively impact the business; those are shown in figure 6. These five forces yield different questions reflecting on different aspects:

1. Bargaining Power of **Suppliers**: Do suppliers have leverage on us (bargaining power)?

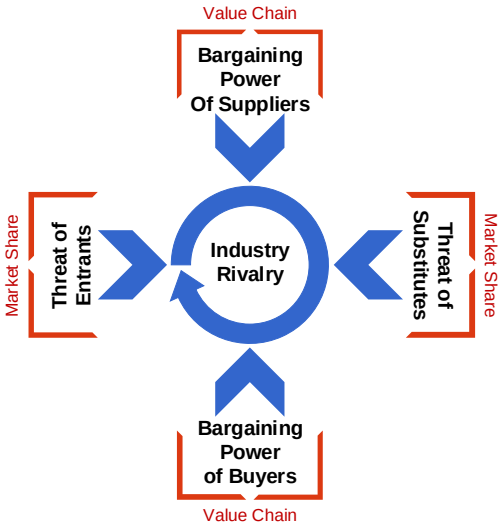


Figure 6: Porter's Five Forces.

2. Bargaining Power of **Buyers**: Is the demand big? Can a buyer easily change its supplier?
3. The threat of **Substitutes**: What are the substitutes to our product/service? How do we compare to those substitutes?
4. The threat of **New Entrants**: Are there barriers to enter into the business domain? Are there other business domains that could enter into this one?
5. Industry **Rivalry**: Is there much competition in the business domain? How are the competitors positioned in the market?

This thesis analyzes the center force "Industry Rivalry". The same analysis for the other forces would lead to complementary machine learning models and visualizations.

In order to integrate this business strategy technique with Big Data techniques, a model representing the domain knowledge is needed, as explained in the previous chapter. Regarding the "Industry rivalry" there is a well-known principle that explains its behavior based on the market price: the Law of Supply.

The Law of Supply (Wikipedia, 2019h) is a fundamental principle of economics theory; it says that the higher the market price of a product is, the more quantity is supplied. This has to be taken on the market perspective, not with just one supplier, so when there is will (in the market) to pay more for a product (or service) there will be more supply in the market because more suppliers will be willing to produce the product or service; the demand (buyers) react in the opposite direction, the higher is the price of a product the lower is their will to buy it, demand and supply curves are the inverse of each other. All this given that all variables affecting price are static. The point where both curves cross each other is the market equilibrium, where the market price moves naturally (Wikipedia, 2019j), this is shown in figure 7.

The supply and demand model has several hard assumptions; one of them is the assumption of a perfect competition market; the second one is that all the other variables are static, i.e. “cæterīs pāribus” (Wikipedia, 2019a). Figure 7 depicts the logic behind this model, decreases in demand (D curve shifts to the left) imply decreases in prices and supply. Likewise, increases in demand (D curve shifts to the right) imply increases in prices and supply.

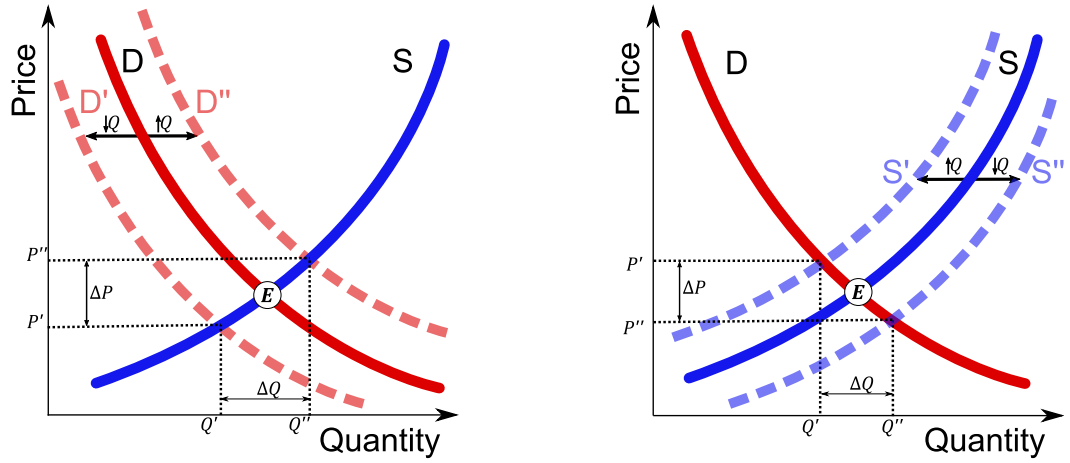


Figure 7: Shifts on Supply or Demand affect Market Price.

These increments (Δ) vary with time, and they depend on the market situation, ideally oscillating between a point of equilibrium (E), where the demand and the supply intersection sets the market price. Given both figures, several variables are conforming the business model. At this stage, these variables are considered “meta-features”, i.e., the real features of each data set will be derived from these meta-features. Those are the market price of the product and its delta (ΔP); where the product is the services provided by the bars, mainly beverage. The increase, or decrease in supply (ΔS); where the supply is the number of bars in the city town of Madrid. Finally, the increase or decrease in demand (ΔD); where the demand is the number of consumers of the services provided by the bars in Madrid.

The law of supply explains how the business domain behaves. It answers the questions: “Is there much competition in the business domain?”, and “How are the competitors positioned in the market?”. Low prices in one market might indicate a highly saturated market, where high prices suggest the opposite. Similarly, a constant increment on supply might indicate low saturation, and a constant decrement on demand will indicate the opposite. Frequent and high oscillations of supply around the equilibrium point might indicate an unstable market, possibly with high competition and low profit. Likewise, low frequency and low magnitude on supply changes shows a stable market, or a market with high barriers of entrance.

Chapter 6

DDSD Phase Two: Feature Engineering

This chapter transforms the outcome of phase one, the business model, and the meta-features, into a data-set and a set of tasks. Feature engineering is applied to the meta-features to define specifics features related to the business model; these features will produce a data-set that will feed one or more models. A set of data tasks are defined to address this goal based on the goals laid out during the strategy analysis, i.e. "Which location does maximize our chances of success?", and "Is there much competition in the business domain?", and "How are the competitors positioned in the market?".

Based on the meta-features defined in previous phase, as shown in table 2, the meta-features are mapped to specific features; i.e., measurable characteristics that may or not be available[1].

It is essential to notice that these definitions imply that all the features depend on time. Although this is not explicit to the demand-curve model, changes on any of the variables imply an increment in time, i.e., a time component.

The table 2 expresses each meta-feature as one or more features. Since it is representing the "industry rivalry", the most important feature is the number of suppliers over time, this will give an idea of the saturation of the market regarding providers. The ΔD can be expressed in population variation, assuming that an increment in population implies an increment in demand. It also can be expressed as an increment in the total turnover of the companies in that area. This is harder to measure (actually to obtain) but more accurate. The variation of the market price, ΔP, is a very common economic variable. However, it has to be constricted to the area and

[1] These features can be derived from an existing data-set or data-source, in case there is such; but always keeping the logic behind the business strategy model defined (demand-curve in this case).

Meta-Feature	Feature	Description
ΔS	Number of suppliers over time	Increment or decrement of limited bars and restaurants in an area during a specific period.
ΔD	Population over time	Increment or decrement of people above 18 years old during a specific period.
	Consumption over time	Consumption of food and beverage in an area during a specific period
ΔP	Market price over time	Increment or decrement of food and beverage prices (consumers' price) in an area during a specific period.
(All)	Location	Geographical unit taken to which other features refer. Granularity: ward or district.
(All)	Period	Time interval taken to measure other features. Granularity: month or year.

Table 2: Mapping between meta-features and features.

the type of product, i.e., food and beverage consumers' price within a ward or district. Location and period are the features defining the scope used to measure all the other features.

6.1 Tasks Identification

With the set of features defined, it is possible to define several tasks to meet the objectives derived from the business strategy technique. Since these tasks are based on machine learning algorithms, below there is a list with six main types of machine learning algorithms (Brownlee, 2013):

- **Classification**: given a set of **classes** the model predicts which class an instance belongs to; similarly, a scoring algorithm will give a set of probabilities of belonging to each class.
- **Regression** (value estimation): the model will **predict** the numerical value of a variable for an instance.
- **Clustering** (unsupervised): the model groups instances based on the similarity of their features, no target attribute; indeed, there is no class defined.
- **Association Rules**: the model extracts rules that will **describe** relationships between features.
- **Artificial Neural Networks** and Deep Neural Networks: models for classification and regression but inspired in biological neurons. Both are used as non-linear models, where the latter one is used when there is missing data on the data-set.

- **Time Series Forecasting**: type of regression (**prediction**) that includes a time component.

The figure 8 defines a set of machine learning tasks within the current strategy analysis (Porter's five forces). These tasks are the ones contributing to enhancing the strategy analysis.

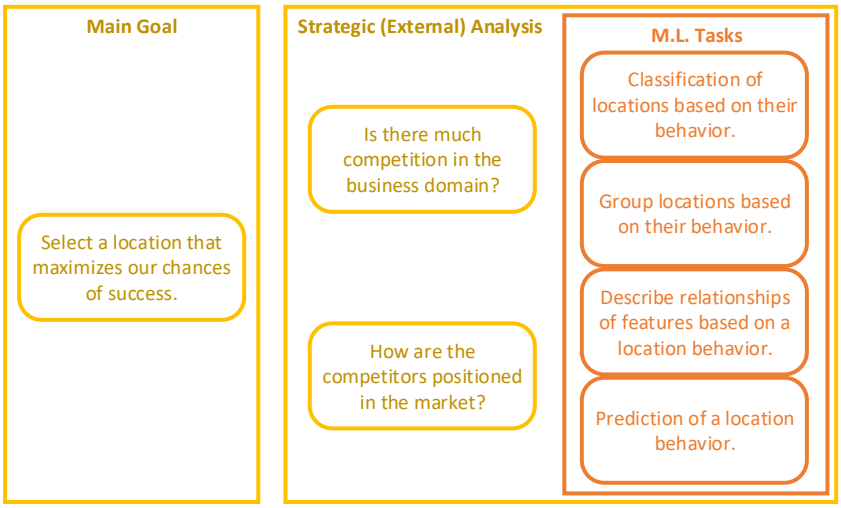


Figure 8: Machine Learning tasks within the business external analysis.

Each of the four tasks defined in figure 8 supports and enhances the current strategic external analysis. Each task leads to one or more data models; these models might work together to produce one visualization, or they might produce their own visualization. Eventually, all tasks together would produce a dashboard tailored for the strategy methodology selected (Porter's 5 Forces).

This thesis covers two of the tasks: a clustering task and a classification one. The first task identifies different types of locations and analyze their supply-demand behaviors. The second task, classification, has a double purpose: it acts as a validation of the clustering tasks, and it provides descriptive information as well. These tasks support the goal shown in figure 8: *Is there much competition in the business domain?*.

The clustering task is an unsupervised learning task (Wikipedia, 2019k), i.e., a learning algorithm that helps to identify patterns in the data without previous knowledge. That means the data is not labeled; indeed, the first exercise is to discover different demand-supply patterns within the 129 wards (codified as time-series). After finding the different clusters, i.e., types of wards based on their supply-demand behavior, it is possible to inspect which wards belong to each cluster. For this thesis, one ward of each cluster, the most representative (closest to the cluster centroid), is selected. Based on these wards, the average turnover, per year and by ward, of all the stores is gathered from the data-source Camertada. With this information, the behavior of the three wards regarding supply-demand, and the economic results (turnover) per

ward are compared to assess which cluster is a better location.

The second exercise goal, classification task, is to collect as many characteristics (static ones) about each ward as possible to provide descriptive information about them. These characteristics compose the predictors of the classification task. The classification algorithm tries to classify each ward in one of the clusters defined in the first task (clustering). So, the idea is to use an unsupervised clustering algorithm to group the wards in different clusters, and after that, to use a classification algorithm with different features, using the clusters assignments as classes. The aim is to see if it is possible to classify back each ward into the correct cluster. This process is similar to the approach followed by Haimowitz and Schwarz (1997) to cluster and predict credit line optimizations. The process is depicted in figure 9 with the two different data-sets.

Cluster data

$\Delta S - \Delta D$	Ward
y	ward 1
y	ward 2
y	ward 3
...	...
y	ward 129

Clusters
C2
C1
C1
...
C3

Classification data

Class	Ward	Feature 1	Feature 2	Feature 3	...	Feature N
C2	ward 1	x	x	x	...	x
C1	ward 2	x	x	x	...	x
C1	ward 3	x	x	x	...	x
...	...	...	...	...	...	...
C3	ward 129	x	x	x	...	x

Figure 9: Cluster and Classfication data sets.

If the classification task succeeds, it is an indication that, among the properties of each ward, some properties contribute to defining the supply-demand behavior (clusters). That acts as a validation of the clustering task. Finally, inspection to the weights of the new features used will show which properties of the ward most likely influence the demand-supply behavior.

Notice the supply-demand behavior is a time-series data-set; it describes each ward based on its historical data regarding the supply-demand relationship. The supervised task uses static data, without a temporal component; it describes each ward based on many different properties (e.g., average education, average incoming, etc.). Both data-sets are entirely different.

The cluster data-set contains the feature derived from the meta-features described in figure 2. The second data set will not include any feature already used in the clustering task; those features will be static ones, properties of each ward.

6.2 Clustering task

Three characteristics will shape the structure of the data-set:

1. The features have a temporal structure; therefore, they are better expressed as time series.

2. There are features with constant values to all samples, e.g., the market price is constant for all the wards.

3. To correctly represent the demand-supply model, the data-set structure must capture the relationships between features.

The first point highlights that each ward's data is a time series. Indeed there are three different time series per ward: ΔS, ΔP, and ΔQ. Therefore, there is one time-series per ward and features, as shown in table 3a.

ward 1...129

Date	ΔD | ΔS | ΔQ
01/01/2000	value 1
01/02/2000	value 2
01/03/2000	value 3
...	...
01/12/2018	value N

(a) Initial [387] data-sets as time series data. One data set per ward and variable.

ward	$\Delta S - \Delta D$	Date
ward1	value 1	01/01/2000
ward1	value 1	01/02/2000
...	...	...
ward1	value N	01/12/2018
ward2	value 1	01/01/2000
ward2	value 1	01/02/2000
...	...	...
ward2	value M	01/12/2018
...	...	...
ward129	value 1	01/01/2000
...	...	...
ward129	value P	01/12/2018

(b) Final data-set with formated time series data for the clustering task.

Table 3: Clustering data-sets.

The second point implies that regardless of the samples (wards), the values of some features are constant. This is because the external analysis is performed in a small area, the city town of Madrid. This is the case of market price and some features for demand. Since the exercise is to cluster the samples (wards), and some features will not discriminate between samples, those features cannot be considered. This type of features (with constant values or close to) have a zero variance or near-zero variance.

Regarding market price, there is no free available data for such a small market price per ward. Indeed, given the small region, it is quite improbable that changes in market price (increase or decrease) would differ in subregions (wards). Therefore, this thesis assumes that market price behavior for food and beverages is constant for all wards.

Regarding demand, it is between the consumption and the population where the real value is. Not all the population want a product (service, in this case, a bar), and more people are willing to consume the product than the actual consumers, but they do not, for whatever the

reasons are (lousy location, price, bad publicity, etc.). Since this thesis does not have access to the consumption data, the population increment variable is selected to represent ΔD. To increase the significance of the variable, only a subset of the population is accounted, those that are more likely to use a bar's services; i.e., those people above 18 years old.

Finally, since the data-set's structure must represent domain knowledge (supply-demand curve), the features will be expressed as the difference between them, i.e., the difference between suppliers increment and demand increment ($\Delta S - \Delta D$). A ratio demand to supply ($\Delta D / \Delta S$) could have been selected, but some wards have very small increments of bars and restaurants in some periods. This will push some values towards infinite, distorting the relationship.

It is essential to understand, within the context of the selected strategy analysis, what the features are describing, i.e., what is the meaning of their different values. There are three possible values for the relationship:

1. Positive $\Delta S - \Delta D$. This value is positive if:
    - Supply has increased more than demand. This is an indication of a (possible) saturated market and, eventually, supply will be corrected.
    - Demand has decreased more than supply. Similarly, this is an indication of (possible) saturated market.
2. Negative $\Delta S - \Delta D$. This value is negative if:
    - Demand has increased more than supply. This is an indication of a (possible) low competition market. Eventually, supply should increase.
    - Supply has decreased more than supply. Possible over-correction of a saturated market. Similarly, this is an indication of a (possible) low competition market.
3. Zero, or near zero, $\Delta S - \Delta D$. The equilibrium point (see figure 7) is reached; both variables increase and decrease proportionally.

Together with the value of the variable selected, it is interesting to analyze its variation during the time, e.g., deep falls or increases might imply unstable markets.

Technically, the task consists of clustering multivariate time series. To do so, the Dynamic Time Wrapping will be used to cluster all the time series. The number of time series to clusters is equal to the number of wards, 129. The final data-set for the clustering task will have the format is shown in the table 3b.

6.3 Classification task

The goal of this task is to classify a new data-set of features (per ward) based on the clusters already defined; the clusters will be the classes of each sample (ward). In this case, there are no specific features to use, since they do not describe any business model. Given a clustering outcome, the aim is to discover features that can help to classify a city ward as member of one specific cluster. The assignment of wards to each cluster will be used as a class, i.e., as the variable to predict. The other features will act as predictors, and each ward will represent a sample; as it is shown in figure 9.

In this respect, the features to compose the data-set for the classification task can be selected from any data source, and the bigger the number, the better. The only condition is to keep the granularity of the clustering task; i.e., the samples must represent wards of the city town of Madrid. For instance, the number of pensioners in a city ward could be a feature of the classification task.

The classification task will be trained to predict to which cluster (class) each sample belongs based on the new features. Once it is done, its performance will be compared against a random classifier (random guessing) using a cumulative gain curve and a cumulative lift curve (Wiki, 2019). The accuracy can also be used; if there are K classes (clusters), a random classifier will yield an accuracy of $1/K$. If the static features collected for the classification data, do not have anything to do with the supply-demand curve of each ward, the performance of the classifier will be similar to a random classifier. Alternatively, if the classifier performs better than a random classifier, the new features have a relationship with the wards' supply-demand behavior.

It is important to test the correlation between the cluster features and the classification features. If the cluster feature, $\Delta S - \Delta D$, is correlated with any of the classification features, the task would not be needed. The correlated (classification) feature would explain the behavior of the supply-demand without the need of any cluster or classification task, i.e., there is a linear relationship between both so there is no need of a machine learning model. The maximum correlation between the cluster feature ($\Delta S - \Delta D$) and any of the classification features for 2018 (last data available) is $r = 0,2287$, the minimum is $r = 0,00117$ and the average is $r = 0,0768$. So the correlation is, at best, weak. The complete list of correlations between cluster and classification data can be found in Appendix B.

After having a data model trained, to know which features are the ones with the stronger relationship with the wards' supply-demand behavior, the weights of each input attribute (features) in the data model are inspected.

Chapter 7

DDSD Phase Three: Data Mining & Preparation

Given that the nature of the main goal implies to get a location within the city town of Madrid, the data source selected is “Banco de Datos de Madrid” (Ayuntamiento de Madrid, 2018a). It is an open statistical data series source. Another data source will be “Camerdata” (Camerdata, 2018), a company owned by Spanish Chamber of Commerce (not open data). The more data sources used, the higher the chances of getting better results. However, given this is an academic work, the thesis will use only those two mentioned.

As explicitly depicted in the DDSD workflow (figure **??**), there is a loop between this phase and the previous one. Indeed, the data source influences feature engineering. The availability of the data and its quality might force some decisions in feature engineering. In this case, as already commented, the variable ΔD (demand) is based on the population’s growth due to limitations on the data sources. Similarly, the granularity selected for this thesis (ward) is based on the data provided by “Banco de Datos de Madrid”. Indeed the second data source offers a similar granularity but with different “labeling”, zip codes instead of wards. This type of mismatches in data sets are widespread and, eventually, has to be sorted out. So, although these two phases have been done following an iterative approach, this thesis describes the outcome sequentially.

7.1 Data mining

7.1.1 Clustering data

The available data for the clustering task (Ayuntamiento de Madrid, 2018a) comes as a spreadsheet (Microsoft Excel sheet). The spreadsheets will be converted into CSV files. As defined in the previous section, there are two features to collect: population growth (representing demand) and bars and restaurants' growth. Notice the restaurants enter into the picture, and the business context is about bars. Again, the data-source has influenced in the feature selection since there is no data with this specific granularity (only bars).

The first feature can be collected from the census data provided in the data source. More specifically in the demography section (C), entry named "*Población mensual por distritos, barrios y secciones*" (monthly population by district, ward, and sections). The data goes from 2003 to 2019, and with three different granularities: districts, wards (within districts), and sections (within wards).

The interface to download the data forces the user to download a file (spreadsheet) per month, so eventually, all of them should be merged (192 spreadsheets). There is a reason for this inconvenience, the structure of the data varies based on changes to the sections and wards (new entries, merges, splits, or renames).

The next variable (bars and restaurants' growth) has a lower frequency and smaller range, with six months periods and data available from 2013 to 2018. In order to consolidate frequency and range in both variables, the census data will be downloaded matching those values: since 2013 and with six month periods. This renders 11 files for the census data.

The census data web interface structures the data by gender, age, and nationality. Since the feature to represent is the demand in bars and restaurants, the data will be filtered by age, from 18 to 85 years old (maximum is a 100 and above). The aim of putting a limit of 85 is to normalize the outcome. Above 85, the data structure of each ward changes if there are people in those ages. A sample of such data is depicted in figure 10.

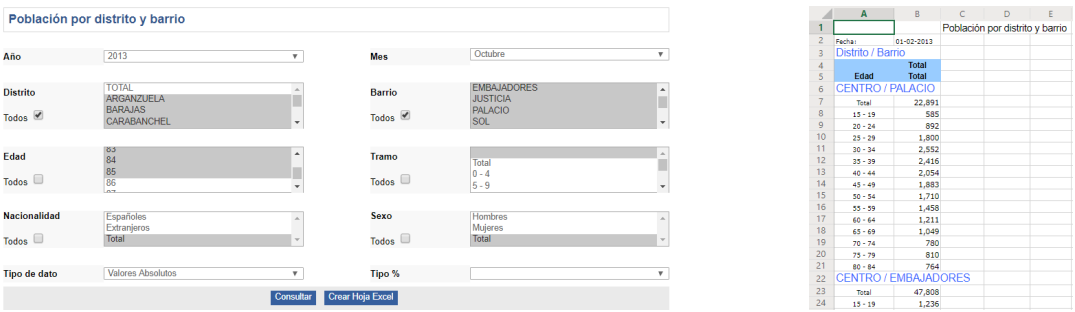


Figure 10: Data source for variable "Demand".

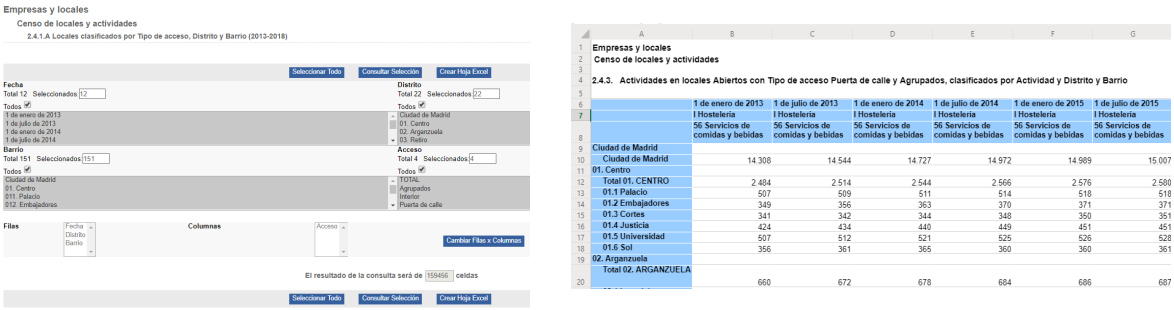


Figure 11: Data source for variable “Supply”.

The second feature is collected from the stores’ census subsection (under companies and stores), from the economic section (D) of the data source, with entry name “*Locales clasificados por Tipo de acceso, Distrito y Barrio (2013-2018)*” (stores classified by type of access, district and ward). As commented, the data has a frequency of 6 month periods and goes from 2013 to 2018, and it has been filtered by type number 56 “Food and beverage services”. In this case, the whole data-set can be downloaded in a single spreadsheet (figure 11).

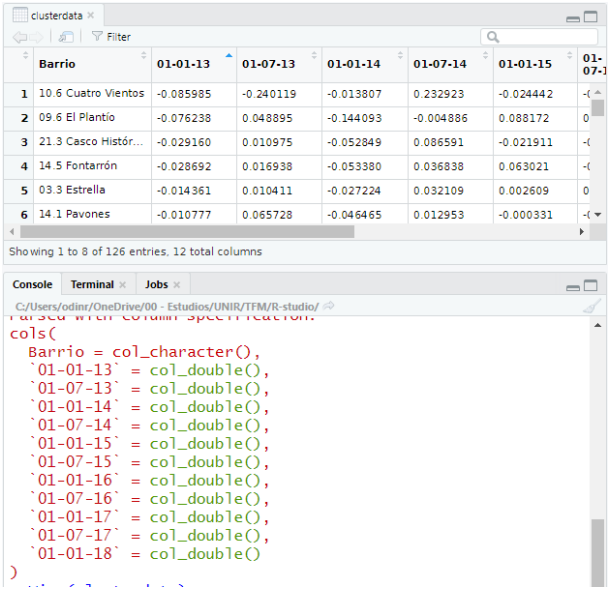


Figure 12: Supply and Demand features (cluster data) loaded as CSV file.

Due to mergers, splits, and new entries, some wards (four of them) have been disregarded to ease the data preparation. Some of the data preparation has been done in Microsoft Excel, specifically combining Demand and Supply into increments (ΔD and ΔS), and finally into its difference (ΔS - ΔD). The resulting CSV file already contains the complete data-set needed for the clustering task.The CSV files is uploaded to RStudio as shown in figure 12. The first column of the data-set (variable cluster data in RStudio) is the name of the wards (label), following the values of the Supply-Demand difference for each period.

As it is evident in figures 10 and figure 11, the structures of both spreadsheets are entirely different. More importantly, the labeling of the wards (samples) does not match in both files.

There are different abbreviations or diacritics, and even different names referring to the same ward (e.g., *"Villaverde Alto"* is mentioned sometimes as *"San Andrés"*, or as *"San Andres"*). This forces to manually identify and match each sample in both data sets before merging them (getting ΔS - ΔD).

7.1.2 Classification Data

As already commented, the classification task depends on the clustering task. The objective of this task is to identify possible properties in the wards that shape the supply-demand behavior. Therefore, the bigger the number of properties (features) included in the data-set, the better. In this task, the data is static, and there is no time series. The data-set would be a snapshot of the wards at a specific time (2018 or 2019). This increases the number of possible data-sets since time-series data is harder to find than static data. The data source for this task is the same as for the clustering task (Ayuntamiento de Madrid, 2018a). Although there are hundreds of data-sets available in that source, only those referring to city wards can be incorporated. The ward is the labeling attribute for both data-sets.

For the classification tasks, a supporting data source is also used "Portal de datos abiertos del Ayuntamiento de Madrid" (Ayuntamiento de Madrid, 2018b). This data source is used to match the wards labeling since the amount of mismatches is higher than in the previous task. The data source also provides an application interface, so it would be possible to get the data, for instance, in JSON format.

The data sets selected, from Ayuntamiento de Madrid (2018b), are the following:

- Residences' average size (m^2) by type of residence, district, and ward. *"Superficie media (m^2) de las viviendas por Tipo de vivienda, Distrito y Barrio"*. Section 4.2.1.3.
- Residences' average declared price ($€/m^2$) and index number by type of residence, district, and ward. *"Precio medio declarado de la vivienda ($€/m^2$) y números índice por Tipo de vivienda, Distrito y Barrio"*. Section 4.2.1.6.
- Residential buildings from the urban cadastre by district and ward ("Inmuebles de uso residencial del catastro urbano por Distrito y Barrio por Año"). Section 5.2.
- Homes by size (n°of members) by district and ward, for each type of home regarding its composition by nationality (Spaniards and foreigners) (since January 1st 2018). *"Hogares por Tamaño (nº de miembros) según Distrito y Barrio para cada Tipo de hogar en relación con su composición por Nacionalidad (españoles y extranjeros) (desde 1 de enero de 2018)"*. Section 1.1.5.1.B.
- Average duration of contracted credit (months) and the number of contracts by type of building, district, and ward. *"Duración media del crédito contratado (meses) y Número de contratos por*

Tipo de bien inmueble, Distrito y Barrio". Section 4.2.1.14.B

- Average age of Spanish and foreign population by districts and wards (2004-2017). *"Edad promedio de la población española y extranjera por Distritos y Barrios (2004-2017)*. Section 6.1.10.A.
- Population in school-age (below 16 years old) classified by nationality, district, and ward. *"Población en edad escolar (menores de 16 años) clasificada por Nacionalidad, Distrito y Barrio (2003-2017)*. Section 6.1.9.A.
- Population above 25 years old classified by level of studies, by district and ward, for each sex (as of January 1st 2018). *"Población de 25 y más años clasificada por Nivel de estudios según Distrito y Barrio, para cada Sexo (desde 1 de enero de 2018)*. Section 1.1.6.1.B.

Each data-set renders one or more features; for instance, the average age data-set is transformed in multiple features creating ranges of ages and sexes. The total amount of features collected is 40. The steps done to integrate all the data is similar to the one done in the clustering data. Since all files are spreadsheets, these transformations have been done with M. Excel. As with the clustering data, many manual steps need to be done to label the data-sets properly. The complete list of features for the classification data set are:

Feature	Description
Superficie (m2)	Area in square meters
Superficie Viviendas (m2)	Average size in square meters of the residences.
Superficie (m2)	Area in square meters
Superficie Viviendas (m2)	Average size in square meters of the residences.
Residencias	Number of residences.
Año construcción (promedio)	Average year of construction of the residences.
Superficie construida total	Total area with buildings.
Superficie construida promedio	Average size of buildings constructions.
Precio Medio Vivienda	Average price of the residences.
Tamaño medio hogar	Average number of family members.
Españoles por hogar	Average number of Spaniards per family.
Extranjeros por hogar	Average number of foreigners per familiy.
Españoles y Extranjeros por hogar	Average number of foreigners and spaniards per familiy
Edad promedio	Average age of the population
Edad promedio españoles	Average age of spaniards population
Edad promedio extranjeros	Average age of foreigners population
Escolares extranjeros	Number of foreign primary school students

Feature	Description
Escolares nacionales	Number of spanish primary school students
No sabe leer ni escribir	Number of iliterate population
Sin estudios	Number of population without primary studies
Primaria incompleta	Number of population with incomplete primary studies
Bachiller elemental, Graduado escolar, E.S.O.	Number of population with Compulsory Secondary Education
Formación Profesional Primer Grado	Number of population with technical studies (first cicle)
Formación Profesional Segundo Grado	Number of population with technical studies (second cicle)
Bachiller superior, B.U.P.	Number of population with pre-university studies.
Otros titulados medios	Number of the population with other pre-university studies
Diplomado Escuela universitaria	Number of the population with university studies (grade)
Arquitecto o Ingeniero Técnico	Number of the population with engineering grades
Licenciado, Arquitecto o Ingeniero	Number of the population with engineering grades and engineering master
Estudios superiores no universitarios	Number of the population with third-level studies (but not university studies)
Doctorado o postgrado	Number of post-graduated
Densidad (Hab./Km2)	Density, habitants per square kilometer
ExtNiños (%)	Percentage of foreign male children (under 18 years old)
ExtNiñas (%)	Percentage of foreign female children (under 18 years old)
NacHom (%)	Percentage of Spanish men (18 years old and above)
NacMuj (%)	Percentage of Spanish women (18 years old and above)
ExtHom (%)	Percentage of foreign men (18 years old and above)
ExtMuj (%)	Percentage of foreign women (18 years old and above)
ExtAncianas (%)	Percentage of senior foreign men (65 years old and above)
ExtAncianos (%)	Percentage of senior foreign women (65 years old and above)
NacAncianas (%)	Percentage of senior Spanish men (65 years old and above)
NacAncianos (%)	Percentage of senior Spanish women (65 years old and above)

Table 4: List of features for the classification data-set.

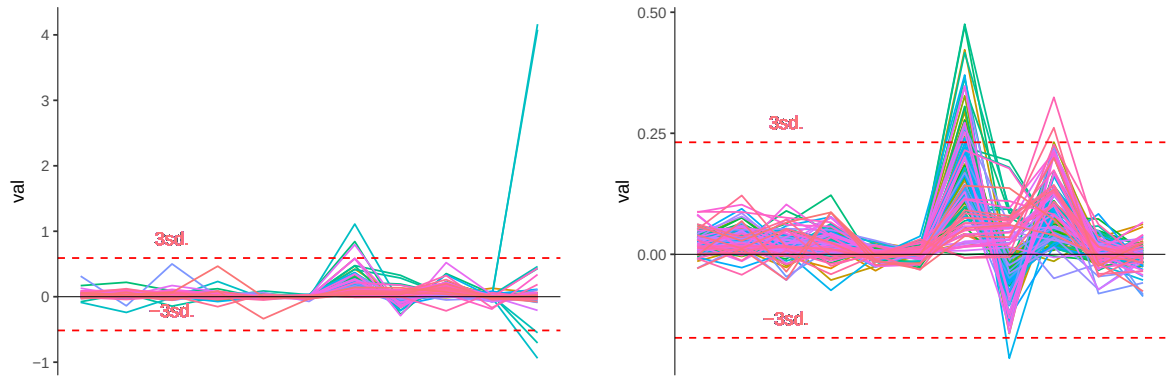
7.2 Data preparation

7.2.1 Clustering data

Once the cluster data-set is in R-Studio, it is possible to inspect and visualize the data. However, before, it is essential to see the type of values in the data-set and if it contains missing values. If there are missing values, either the sample is removed (ward disregarded) or the missing value is replaced. In the latter, with, for instance, the average of the previous and next period.

For the clustering data, there are no missing values (source code extract 3). A visual inspection to the data-set shows that there are time-series, wards, with values far from three times the standard deviation of the whole data-set, as shown in figure 13a (source code extract 4).

It is common practice to get rid of extreme values (outliers) since this type of values is considered noise. But, since these are time-series, it is possible to normalize the affected time-series. Normalization, in this case, means to scale the data in such a way all the values of the different time series can be compared within a similar range. It is possible to use data series tendencies to soft extreme values. In this case, the times-series are normalized using a non-parametric regression with the package *forecast*. Each time-series is normalized (code line 28 in source code extract 4) by using local polynomial regression, a seasonal and trend (STL) decomposition using LOESS (locally estimated scatterplot smoothing). The result is shown in figure 13b.



(a) Raw time series. (b) Normalized time series

Figure 13: Supply Demand time series (129 wards), raw (a) and normalized (b); where $x \in \{Jan.2013, Jan.2018\}$, and $y = \Delta S - \Delta D$. The straight red dotted lines are the values of the third standard deviations, $\pm 3\sigma$.

7.2.2 Classification data

As with the clustering data, once the classification data is uploaded in R-Studio, the very first thing to do is to inspect the data, especially looking for missing values. Again, there are no missing values in the data-set (source code extract 5). Since this data-set contains static data (the year 2018) regarding the different wards, if there were missing values for different wards (samples), an easy approach to solve that would be to get the value from a different year.

Regarding the different ranges of the classification data, it is obvious there are vast differences across all the features. As shown in table 4, the classification data-set contains low values like the number of family members, to high values like the area of wards in square meters. A visualization of the forty features and their distributions across the different wards (samples), figure 14, show the disparity of ranges (x values).

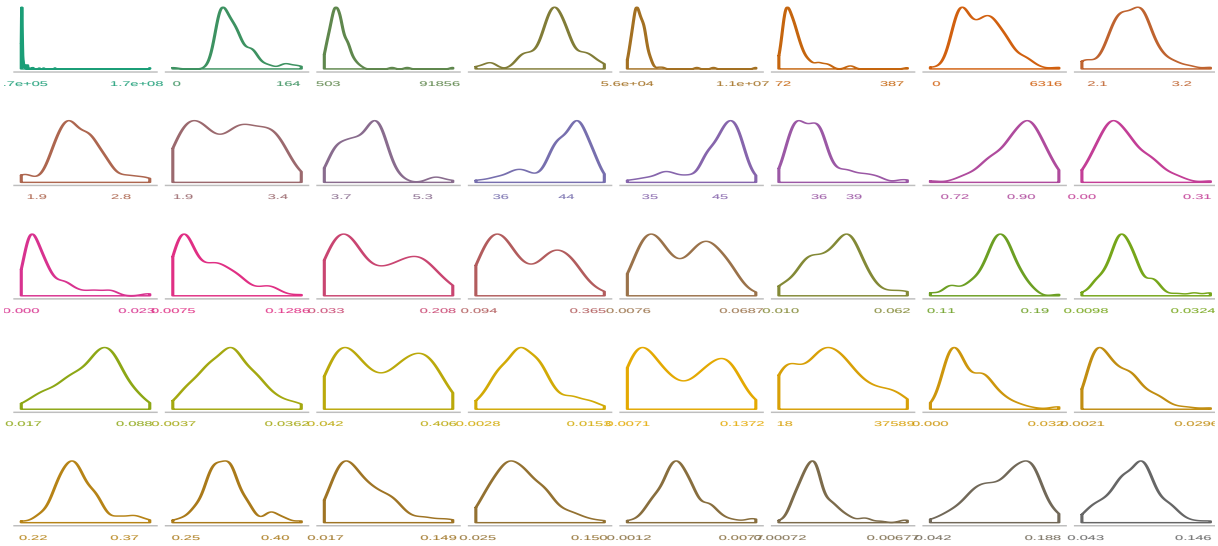


Figure 14: Distributions of the forty features, without normalization. Axis x is the different values of the feature, and axis y is the frequency of those values across wards.

Again, a normalization of the data is required. With such different scales for each feature, the algorithm to measure the distances between them, usually the Euclidean distance, will be distorted. After normalizing the data-set, the distances between different values have a consistent meaning; the ranges of the values of the features across the different samples are the same. Also, all features are centered around zero, with a standard deviation value of one (source code extract 6). A visualization of the forty features and their distributions across the different wards (samples), figure 15, shows the range after the normalization is much narrower (between -5 and +11), and the center (mean) of every feature distribution is zero.

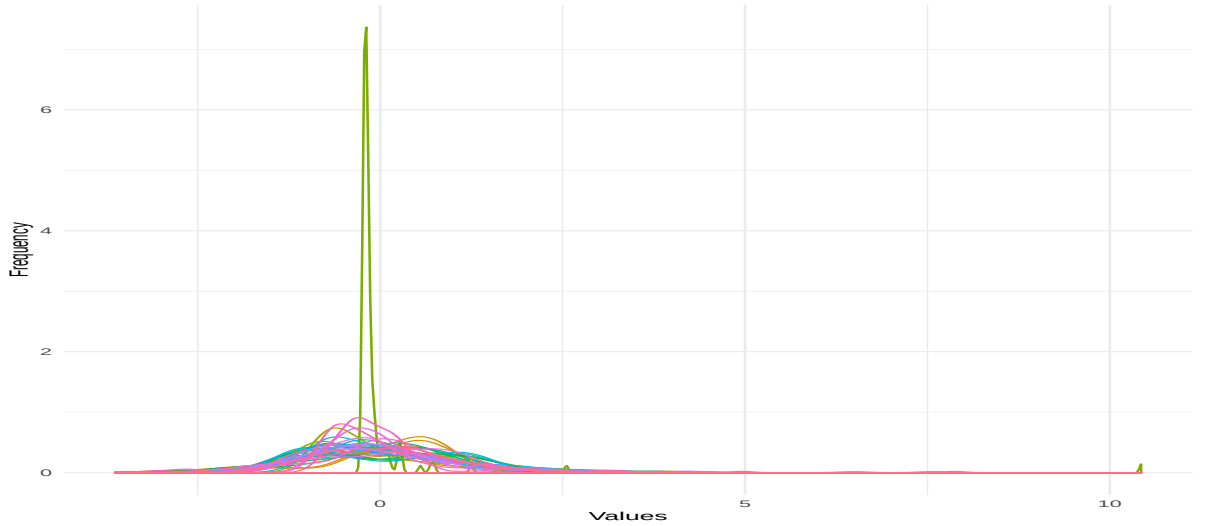


Figure 15: Distributions of the forty features, in one chart and without normalization. Axis x is the different values of all features. Axis y is the frequency of those values accross wards.

It is possible to inspect the same features' distributions, but with one chart per feature, this is shown in figure 16, where it is easy to see the distributions of the features are not affected after the normalization, just the ranges).

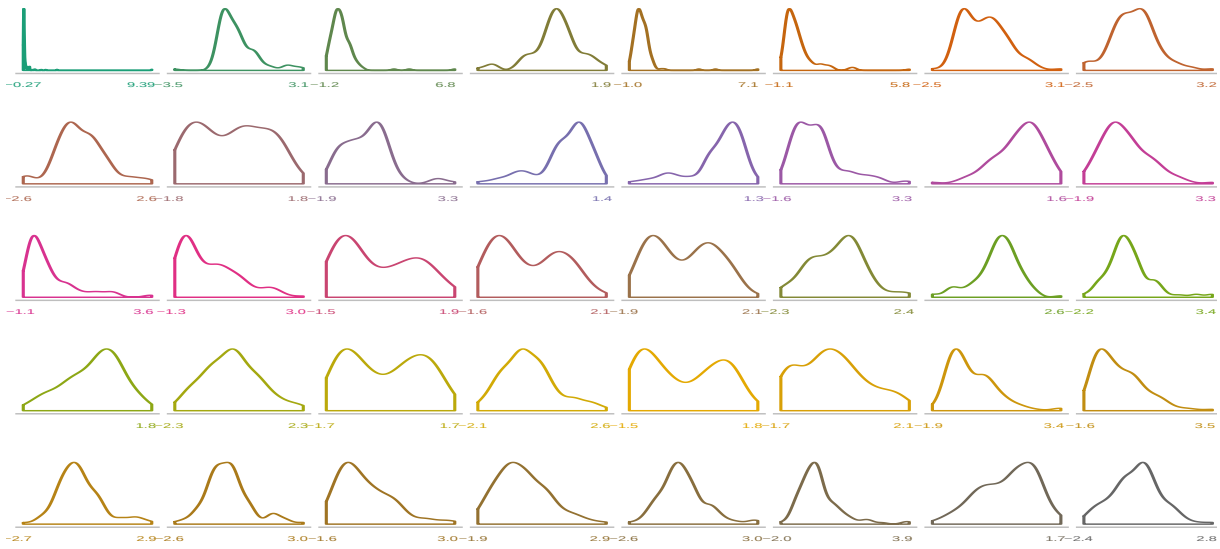


Figure 16: Distributions of the forty features, with normalization. Axis x is the different values of the feature, axis y is the frequency of those values across wards.

The last check to the data-set is to inspect for zero, or near-zero, variance features; since those features do not add any value, just noise. From previous figures, it is clear that all features do not have a near-zero variance given their kurtosis (how thin the curve is). Just the first feature might have issues. To be sure, with the package "caret" and its function *nearZeroVar*, it is possible to get all the features of a data.frame with near zero, or zero, variances effortlessly. In this case, none of the features have either zero or near-zero variance (source code extract 7).

Chapter 8

DDSD Phase Four: Model Implementation & Evaluation

After gathering and preparing the data-sets for both tasks, clustering, and classification, the next phase in the DDSD deals with the implementation and evaluation of the models for each task.

8.1 Clustering model

The clustering task implements an unsupervised clustering algorithm; as defined in section 6.1; the goal is to cluster the wards based on their supply-demand behavior similarities. Commonly, a clustering task consists of grouping (static) properties of a set of samples (wards in this case) using distance metrics (e.g., Euclidean) between their properties (attributes/features). This is shown in figure 17.

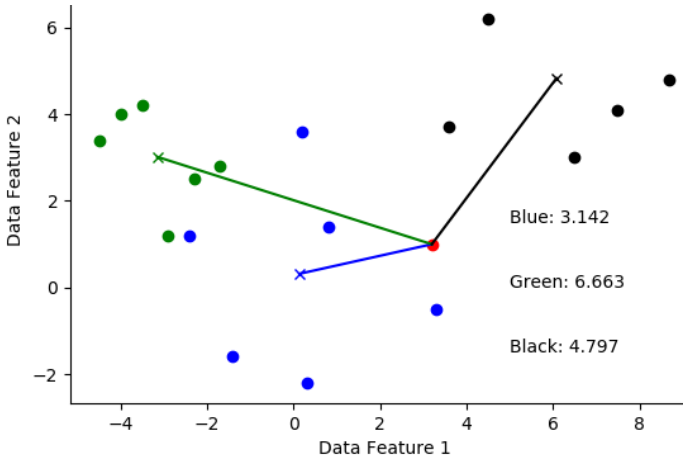


Figure 17: Euclidean distance in a classical clustering task, from (Anagolum, 2019). Each point represents a sample. Colors indicate the different clusters (blue, black, and green). Axis represent two different features.

In a classical clustering model, as shown in the previous figure, with two different features (axis) and three clusters (colored in blue, green and black), each sample (point) is assigned to a cluster based on the distance to the other samples, the most common distance to use is the Euclidean distance (Wikipedia, 2019f).

That would be the case if the clustering data-set were as the classification data-set, with unrelated features. However, in this case, the clustering data-set has just one feature (supply-demand) with a time component; i.e., each sample is a time series, not a single point (value) per feature. The measure the distance (similarity) between time series, there are two approaches:

1. Euclidean distance between periods. Here the distance between each time series is computed by getting the distance between every single period of the time series. For example, the distance between the supply-demand of ward 1 to the supply-demand of ward 2, could be the sum of the Euclidean distances between each period from Jan. 2013 to Jan. 2018. This is shown in the left chart of figure 18.

2. Dynamic Time Warping (DTW) (Wikipedia, 2019d). Here the time component of the time series is accounted, and the periods to compare are selected based on the similarity of the time series between two or more periods (time window). This is shown in the right chart of figure 18.

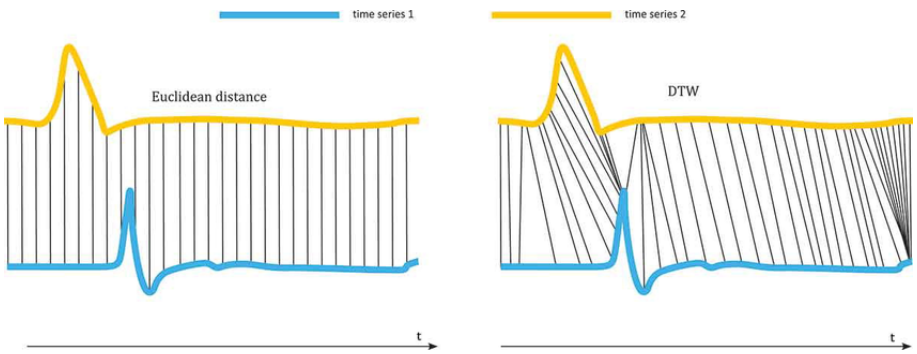


Figure 18: Representation of the Euclidean distance (left chart) and DTW (right chart), from Giorgino et al. (2009). Periods (axis x) of two time-series (yellow and blue) are compared based on their distances with two different algorithms.

Therefore, the distance to use between supply-demand samples in this clustering exercise is the DTW. Since this is an unsupervised algorithm, before starting the clustering task, the optimal number of clusters needs to be calculated. To get such an optimal number of clusters, multiple clustering models with a different number of clusters are run (source code extract 9). Then, different metrics, producing a single value (index), regarding how well each cluster is defined, are applied to each configuration. These indexes give an estimation of the adequacy of each clusterization based on how many clusters have been defined. The indexes used in

this thesis are calculated based on the cluster validity index (CVI) function, from the R package DTW (Giorgino et al., 2009)):

- Silhouette index (Rousseeuw, 1987).
- COP index (Arbelaitz, Gurrutxaga, Muguerza, PéRez, & Perona, 2013).
- Dunn index (Arbelaitz, Gurrutxaga, Muguerza, PéRez, & Perona, 2013).
- Davies-Bouldin index (Arbelaitz, Gurrutxaga, Muguerza, PéRez, & Perona, 2013).
- Modified Davies-Bouldin index (Kim & Ramakrishna, 2005).
- Calinski-Harabasz index (Arbelaitz, Gurrutxaga, Muguerza, PéRez, & Perona, 2013).
- Score Function (Saitta, Raphael, & Smith, 2007).

Silhouette index, for instance, measures the similarity of a sample (using a given distance) to the cluster it is assigned, known as cohesion, compared to other clusters, known as separation. Each index is applied to nine different configurations, ranging from two to ten clusters. Since the data-set represents the supply-demand behavior of wards within Madrid, no more than ten clusters are expected. So each index can be represented as a function of the number of clusters configured in the clusterization; i.e., $f_i(K) = Y_K$, where $i \in$ {Silhoutte,COP,Dunn,Davies-Bouldin, Mod.Davies-Bouldin, Calinski-Harabasz, Score Function} and Y_K is the index value for the configuration K, where $K \in \{2..10\}$. The results are shown in figure 19, where each index is represented with a chart (source code extract 9). The optimal number of clusters is found by

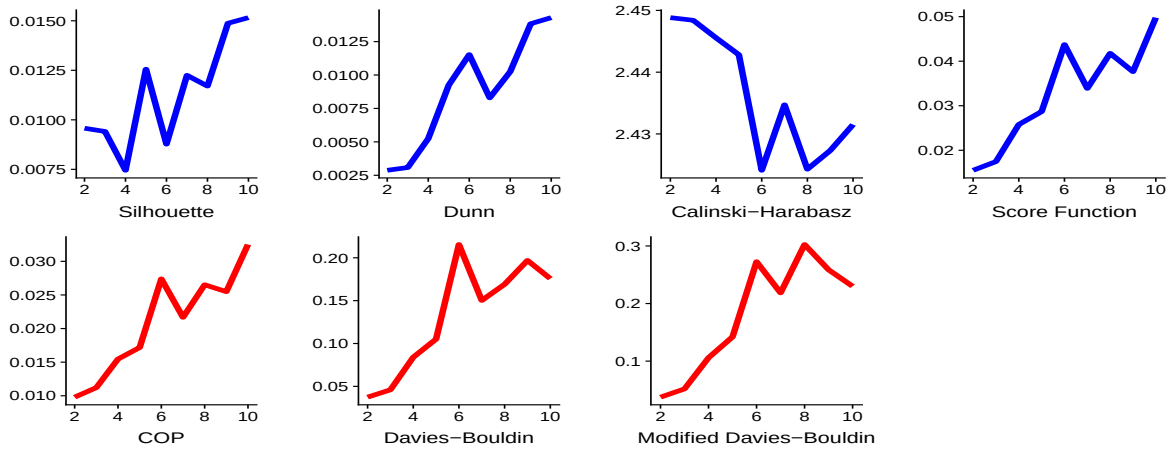


Figure 19: Validity indexes for nine different clusterizations, where axis x is the number of clusters (K=2 to K=10), and axis y is the index value for each validity index.

maximizing or minimizing the functions represented in the charts above; this depends on the index. The most straightforward manner of doing this is to inspect the functions visually for each

index, instead of getting a polynomial regression of each index and, afterward, maximizing or minimizing each function (derivative).

Given the clusterization is grouping the different wards based on their supply-demand curve, a small number of clusters should be expected. Indeed, the lower the value of K, the better. The number of samples for the classification task is precisely 128 (number of wards). Since the clusters assignation is used as the class for the classification task. A high number of classes implies there will be fewer samples per class to do the training and validation.

In figure 19, the blue charts should be maximized and the red ones minimized. For the validity indexes to be maximized, values K=3 (Silhouette and Calinski-Harabasz) and K=6 (Dunn and Score Function) are good candidates for the number of clusters. For the validity indexes to be minimized, the lower the value of K, the better. In K=6 all the indexes have an explicit maximum, so K=6 is disregarded. Therefore, the number of clusters selected for this thesis is three (K=3).

Unsupervised clustering is a stochastic process (Wikipedia, 2019i). The resulting clustering assignments of clustering task varies randomly over time. This is due to the composition of the clusters, and it depends on the selection of initial centroids. Those are randomly selected at the beginning of the process. To overcome that randomness, there is the possibility of setting an initial seed to the random generator, so the clustering process in no longer stochastic; i.e., the same initial centroids will always be selected. Though, this would not give any guarantee of consistent results; i.e., there is no guarantee that the cluster assignments are better with one seed than with a different seed. Therefore, this thesis uses an iterative approach similar to the ensemble weather forecasting method (Wikipedia, 2019e). It is a type of Montecarlo analysis. A high number of clusterizations (a thousand) are calculated, aggregating the results to have an ”average” model. Each clusterization produces different cluster assignments for the samples (wards), the idea is to average this thousand assignments, so the result is, hopefully, stable and coherent; i.e., running several times the thousand clustering tasks, the different averages results will be the same or very close. The ensemble clustering process (ideal[1]) is shown in figure 20.

[1] In the real ensemble process the models are aggregated within the clustering process, since the identification of each cluster is needed.

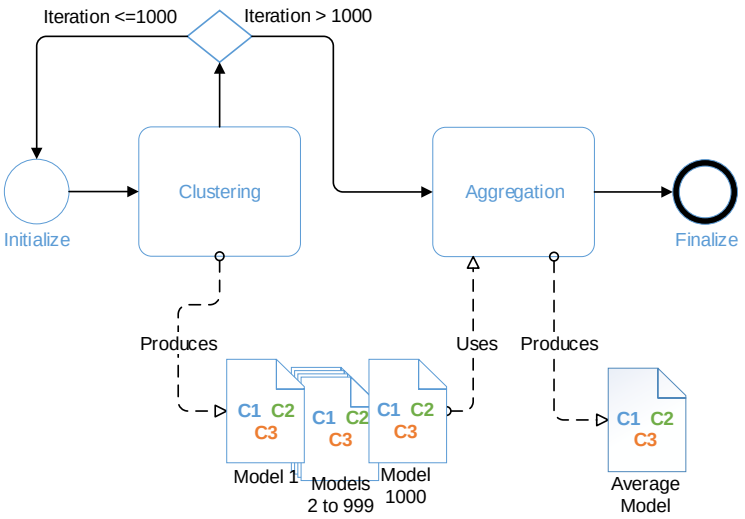


Figure 20: Ensemble clustering process. A thousand clustering tasks are performed, producing a thousand models that are aggregated, to finally get an average model.

In the ensemble clustering, each model produced during each iteration has two essential components that differentiate the model from the rest: the clusters assignments, i.e. which sample belongs to which cluster (C1, C2 or C3); and the centroids of each cluster, i.e., the time-series of each cluster around each sample of the cluster is aggregated. Therefore, to aggregate a thousand models (results), it is needed to aggregate the sample assignments of each cluster and the centroids of each cluster. However, the stochastic nature of the clustering process introduces an issue: it is not possible to identify any of the three clusters (C1, C2, and C3) across different models. Indeed, the clusters of one model are different from the clusters of any other model, in samples assignments and centroids. Therefore, it is not possible to get an average cluster 1, an average cluster 2, and an average cluster 3 since they cannot be identified. This is shown in figure 21.

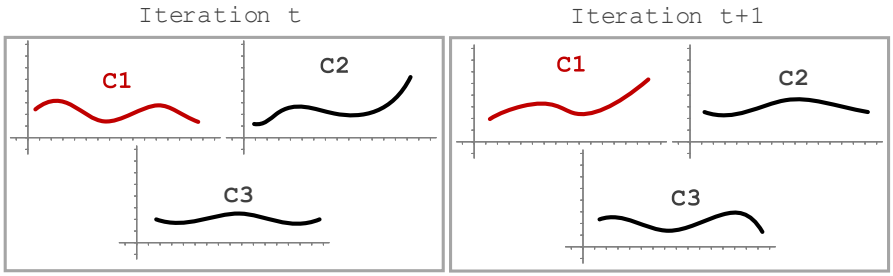


Figure 21: Two clustering iterations (t and t+1), produces three different centroids for clusters C1, C2, C3. Notice the first cluster centroid (C1) in the first iteration (t), is totally different to the first cluster centroid (C1) in the next iteration (t+1).

The figure 21 shows the issue with the centroids of each cluster. The same issue applies to the assignments of samples to each cluster. Each sample (ward) will be assigned to different clusters (C1, C2, or C3) for each iteration.

To identify the three different clusters across models (different iterations), the ensemble process is based on two assumptions:

1. In each iteration, the probability for a sample (ward) to fall in one of the three clusters is higher than in any of the other two.

2. Between two iterations, the similarity between cluster CX of one model (iteration t) with cluster CX' of a second model, is higher than the similarity with the other two clusters of the second model.

Based on those two assumptions, to identify each cluster, the centroids of each cluster in one iteration are compared with the centroids of the cluster in the previous iteration. Each cluster (in iteration t+1) is identified as C1, C2 or C3 based on its similarity to the previous iteration clusters' centroids (C1, C2, and C3 in iteration t). Then, immediately, both models (t and t+1) are aggregated so that the resulting average model will be the basis for the comparison in the next iteration. The comparison, similarity, is made with the DTW function to get a distance matrix between all the centroids (time series). The idea is depicted in figure 22.

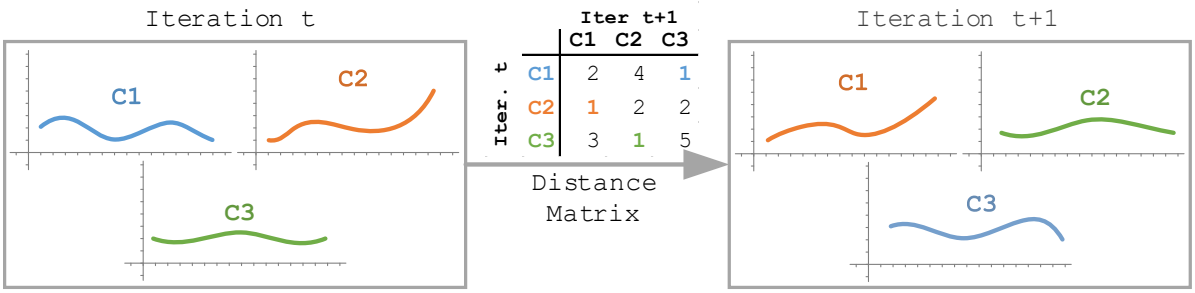


Figure 22: Distance matrix based on dynamic time warping to identify clusters in different iterations. Notice that based on the DTW matrix, cluster C1 in iteration t has been identified as C3 in iteration t+1.

Once the clusters are identified, it is possible to average the two properties of each model: centroids and samples assignments. The centroids are be merged (averaged) using a weighted mean, so in the first two iterations, the means of the centroids of each iteration weight the same, 50%. Also, the result is an average centroids C1, C2, and C3 to be used in the next iteration as a basis. In the n^{th} iteration, the centroids of the model weights $1/n$%, since the average centroids already have the aggregation of n-1 iterations. Regarding the samples assignments to each cluster, in each iteration, each sample (ward) will account for how many times it has been assigned to each cluster (C1, C2, or C3). At the end of the process, each ward will be assigned to the cluster that it has been assigned more times. The complete source code is available in extract 11.

The centroids for each cluster obtained in the thousand iterations can be visualized in figure 23. Each chart contains one thousand centroids for each cluster (C1, C2, and C3). The black

line shows the resulting aggregated centroid for each cluster. Also, in the title of each chart, there is the number of wards assigned to each cluster.

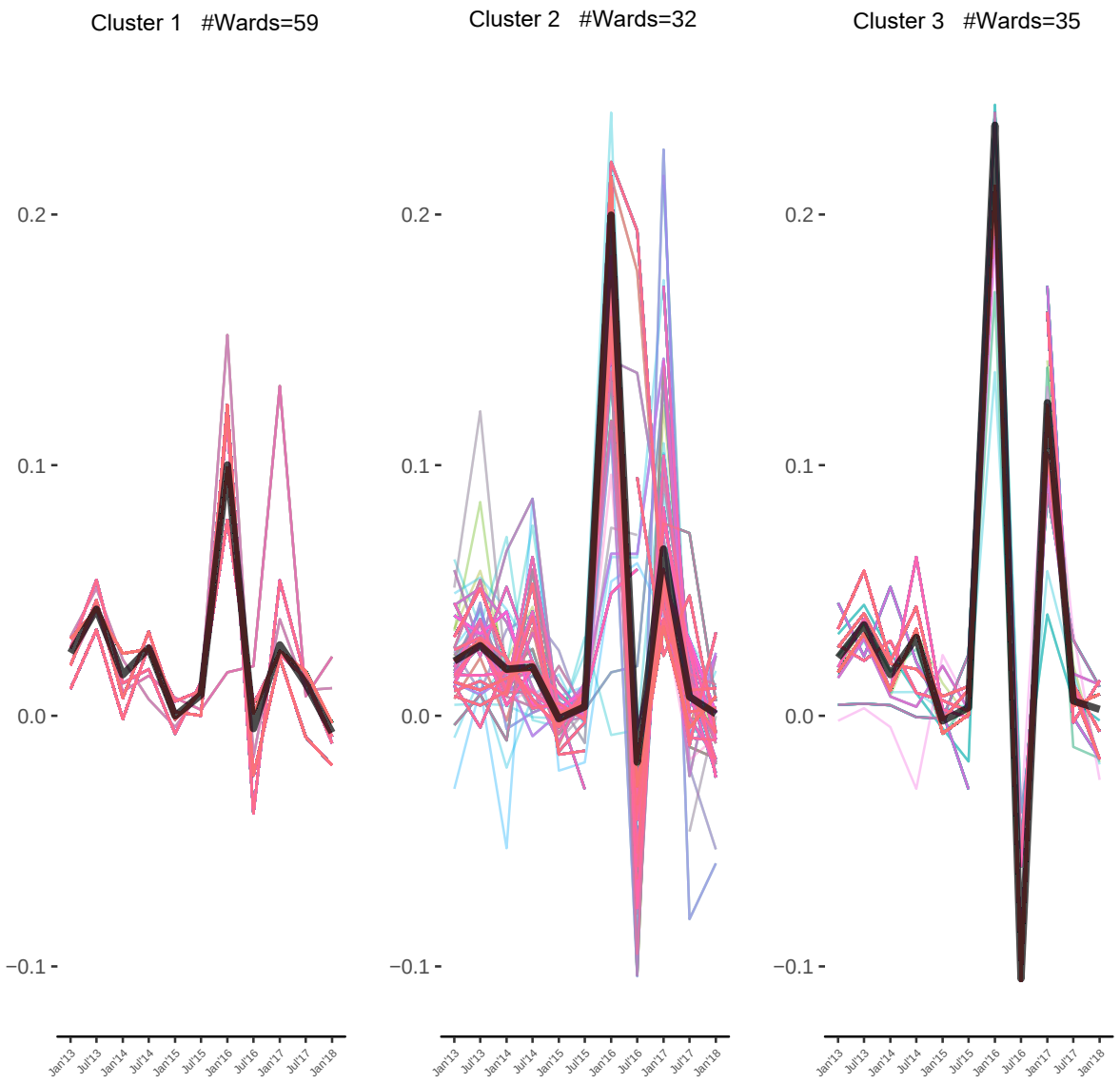


Figure 23: Averaged centroids (black lines) for a thousand clustering iterations (coloured lines) with the number of wards assigned to each cluster: Cluster 1, Cluster 2, and Cluster 3. Axi x represents time, and axis y represents the values of the demand-supply feature.

A visual inspection shows the three clusters have similar supply-demand reactions. Though, the magnitudes of these reactions differ. Cluster 1 has softer reactions than the other two. It looks like cluster 3 overcompensates periods with low supply more abruptly than the other two. Based on the three charts, there are some immediate conclusions:

- The supply-demand behavior in all ward have very similar reactions; they differ though in the magnitude of these reactions.
- There are three different groups of wards, differentiated mainly on the magnitude of their

reactions to changes. Initially, this could be linked to the size of the ward's population, i.e., its capacity to absorb changes. However, the correlation between the supply-demand values and the population size in each ward is very weak ($r = 0.0905$[2]).

- Given the stability of the group of wards in cluster one, it seems these locations are better than those assigned to the other clusters.

The whole process has been run four times to test the stability of the clusters assignments. As expected, in each iteration the identification of the clusters varies[3], but their assignments (wards) are stable. Given the three clusters centroids, it is possible to get the distances of each ward to each cluster. A DTW matrix is calculated based on all the wards' supply-demand curves and the three clusters centroids (source code in extract 12). Table 5 shows the first wards based on their distances to Cluster 1. They are sorted by distance (similarity) to Cluster 1 centroids; with small variations, during the four iterations, the assignments are the same.

Iter. 1	Iter. 2	Iter. 3	Iter. 4
Abrantes	Abrantes	Abrantes	Abrantes
Aluche	Canillas	Aluche	Canillas
Canillas	Aluche	Canillas	Aluche
Lucero	Lucero	Lucero	Lucero
San Isidro	Las Águilas	San Isidro	Las Águilas
Las Águilas	San Isidro	Las Águilas	Buenavista
Buenavista	Buenavista	Pinar del Rey	Pinar del Rey
Pinar del Rey	Pinar del Rey	Buenavista	San Isidro
Peñagrande	Peñagrande	Peñagrande	Peñagrande
Comillas	Comillas	La Paz	Comillas
Cuatro Vientos	La Paz	Comillas	Cuatro Vientos

Table 5: Cluster 1 assignments (truncated) through four iterations (4000 models) of the ensemble clusterization.

The most "representative" ward for Cluster 1 is Abrantes (closest to Cluster 1 as shown in the previous table). Similarly, for Cluster 2 is "Cuatro caminos", and for Cluster 3 is Universidad.

8.1.1 Clustering Evaluation

The main objective of the task is to select a location based on the "rivalry" of the market, modeled by the supply-demand curve (domain knowledge); i.e., to discriminate each supply-demand group based on its economic impact to the future location. All the wards have been assigned to

[2]This is the averaged correlation between both variables in all periods of the time series (between 2013 and 2018). With a minimum of $r = 0.0007$ and a maximum of $r = 0.1726$. The complete table of correlations can be found in Appendix B

[3]Through iteration 1 to 4, cluster one was named as:Cluster 1, then Cluster 1 again, then Cluster 3, then Cluster 2. Similarly, for Cluster 2 and 3.

three different groups and, regarding the algorithm, these groups have different supply-demand behaviors. Based on the law of supply and demand, this should economically impact the market in each location. To validate that, the turnover of the companies in each of the three groups is analyzed. If these three different supply-demand behaviors exist, they should also affect the turnovers of the companies in such groups.

Based on the three "representative" wards, a table with the average of the turnover for all limited companies with NACE code "beverage and food", since 2011, and o each of the selected wards. This data, from the second data source (Camerdata[4]), is based on zip codes, so there is not a one to one relationship between zip codes and wards. Therefore, the resulting zipcodes representing each cluster, have been selected based on the wards map available in Estadística (2016). The resulting three zip codes are:

- Zip code 28047 for Cluster 1. This zipcode contains Aluche, Lucero, and Los Cármenes. There was not a good zip code selection for the ward Abrantes; it shares the ZIP code with other two wards that belong to different clusters.
- Zip code 28004 for Cluster 2. This contains Universidad and Justicia.
- Zip code 28020 for Cluster 3. Thi contains Cuatro Caminos and Castillejos.

Te figure 24 shows the turnovers for the three Zip codes.

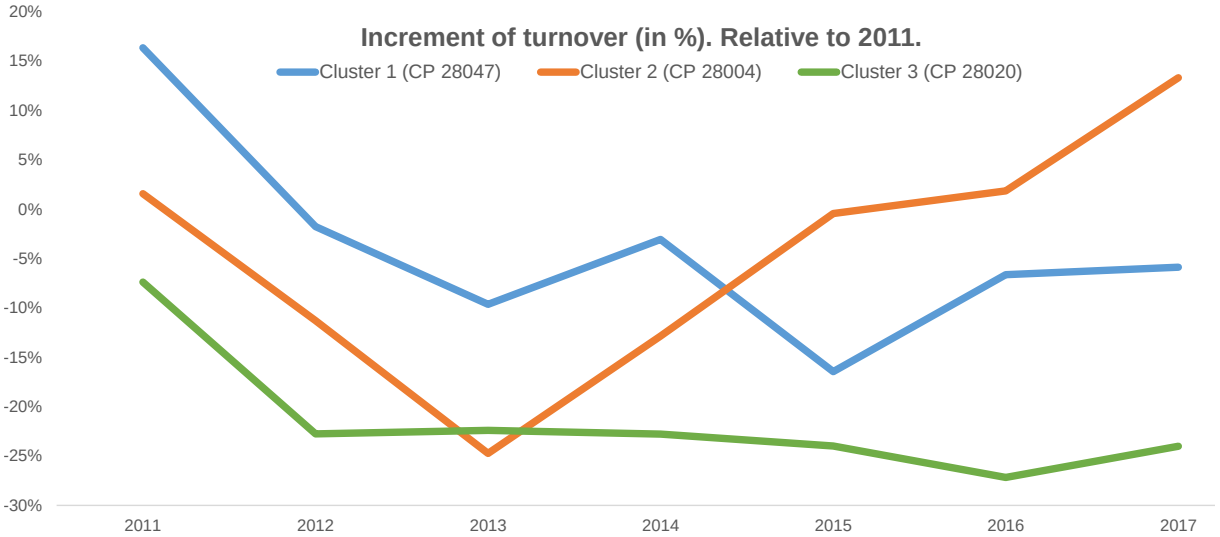


Figure 24: Average turnover incremement per year and per ZIP code, for all limited companies with NACE code "beverage and food". The increment is relative to year 2010.

Indeed, it seems the third zip code (representing Cluster 3) is not a good selection. The significant changes in their supply-demand curve might have something to do with such steady

[4]This data-set is not open, as already commented, so it is not possible to get the data for all the wards, it is simply too costly.

decrement in its turnover. Zip code 28047 (Cluster 1) is the one with higher turnovers, at least until mid-2014, then Zipcode 28004 (Cluster 2) recovers much faster than Zipcode 280047 (Cluster 1) and surpasses it.

Therefore, although cluster one is the most stable zone regarding supply-demand, it seems that is too stable, cluster two has a positive increment with since 2015.

8.2 Classiffication models

The classification task implements several algorithms on the classification data-set prepared in phase three. The principal goal is to evaluate if any of the characteristics of each ward for this year (new features) has any influence in the wards' supply-demand behavior. If that is the case, it would be possible to predict the class of a city ward (cluster defined previously) based in the new features.

In this respect, as long as any algorithm produces better results than a random classifier, it would indicate there is a relationship between the wards' characteristics (classification data-set) and the wards' supply-demand behavior (clustering data-set). Ultimately, the list of weights in the classifier model will reveal which of the classification features are responsible for such a relationship, if any.

To implement classification models, the R package "*caret*" (Kuhn, 2008) is used. It contains about 250 different machine learning algorithms. A set of different types of classifiers are selected for this task: K-nearest neighbor, classification and regression tree (CART), and a Random Forest. The *caret* package acts as a framework that encapsulates many algorithms in R for machine learning from many different libraries; it offers a common interface to use all those algorithms with a nice and rich workflow. Though, with so many dependencies and with such a rich and collaborative language, R, there are integration issues. To overcome them, the first requirement is to have the data-set as "*data.frames*" (no data.table, no matrix,etc.). Another requirement is to use short alphanumeric names for the predictors (lines 16 and 17). Also, the class column must be factors, not numbers (lines 20 and 21). With the algorithms used, these three measures allow using the package without any problem. The needed process can be found in source code extract 13.

The classification data-set contains 43 variables and 126 samples (wards). The first variable is the name of the wards; this will not be needed in the modeling. The last variable is the cluster assignments done in the clusterization task; this is the class of the data-set. The rest of the variables are the predictors; these are the features detailed in table 4.

Since this task is based on supervised classification algorithms, the data-set must be di-

vided into a training data-set and a validation data-set. The first one will be used to train the models and the second one to check the real performance, i.e., each model will be validated against data never seen during the training. After running the classification models with different configurations, it is clear that the amount of samples is not large[5].

So a trade-off between assurance (validation) and performance (training) must be done. The higher the partition for the training, the more chances to get better performance. Conversely, the lower the partition for the training, the more chances to get higher assurance during the validation. Since the number of samples is low, any reduction of the validation data increases the chances of getting imbalanced training data. In this case, with this data-set, a partition that has proven to yield good results while keeping a minimum assurance is a 70%-30% partition. So 30% of the data-set will be used to validate the models; this implies around ten instances per class. As commented, notice the number of samples is not assigned equally to each cluster (class), since the class represents the cluster assignation of the previous exercise. The amount of samples per class is around 30, as shown during the clustering task in figure 23, and without the validation data, this figure drops to roughly 20 samples per class. Figure 25 shows the distribution of the training data-set, respect to two of the forty different features. Notice the low number of samples for class (cluster) number 3 (green circles).Also, notice there are gaps between some values in the distributions of the variables. With a low number of samples, when randomly splitting the data-set in two, the resulting sets might miss samples covering specific ranges, like it is the case here.

To increase the amount of data during the training phase, and mitigate the imbalanced data effects on training, the training data-set is oversampled; i.e., synthetic samples are produced in the data-set. This is done by applying the synthetic minority oversampling technique algorithm, SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002).The parameter "*perc.over*" controls the number of synthetics samples; it sets the maximum ratio of synthetic instances of the minority class against the majority class. The figure selected is 300; it is the one that yields the most balanced data-set. After the oversampling, the amount of available training data increases (obviously, the validation data has not been oversampled). This is shown in figure 26; notice class two and three (green andn orange circles) are increased more than class one (blue), this is obvious by inspecting both charts.

[5] In chapter "10", there are some comments regarding different strategies to tackle the same problem with more samples, i.e., getting each ward represented by more than one time series.

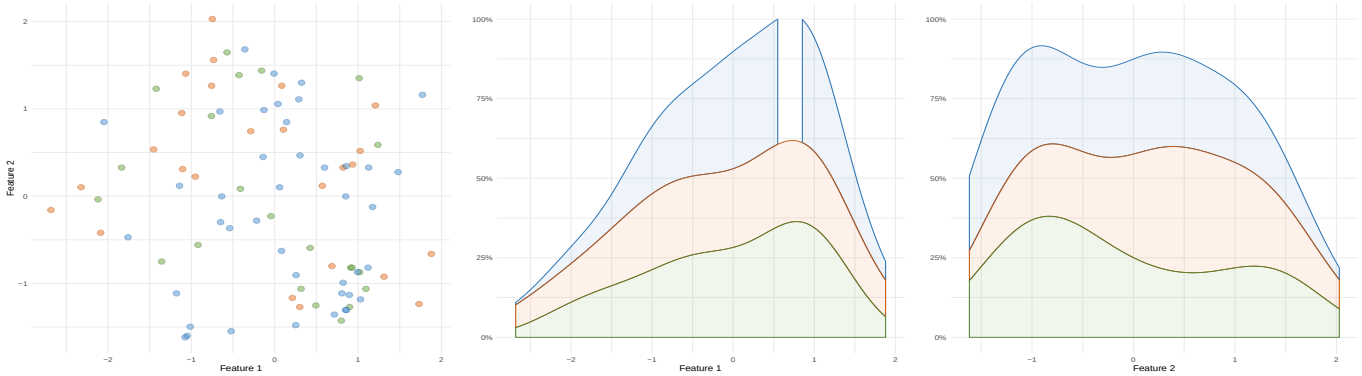


Figure 25: Scatter plot representing original training data-set samples for two features (left chart). Also the marginal distributions (stacked) for each feature (center and right chart). Blue circles are samples of class 1, orange circles are samples of class 2, and green circles are samples of class 3. Source code in extract 20.

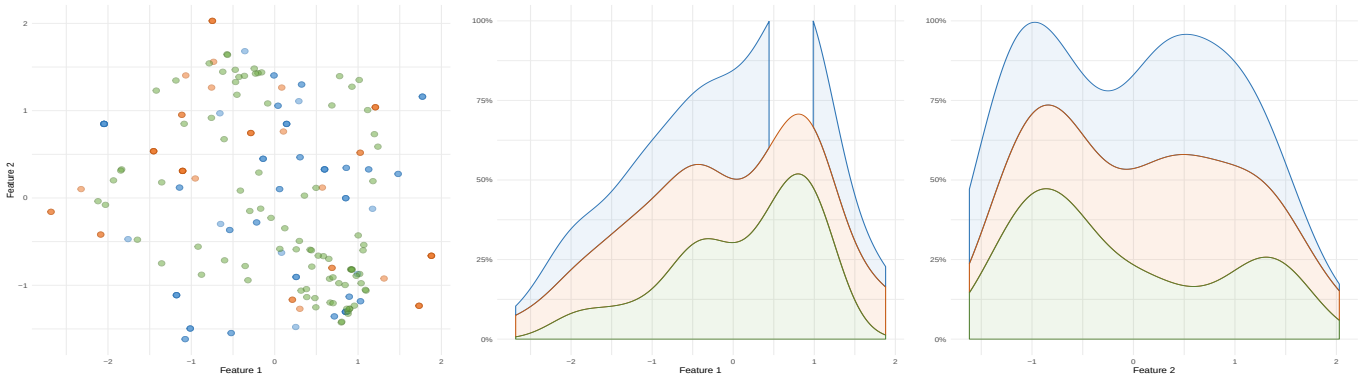


Figure 26: Scatter plot representing the training data-set, after SMOTE oversampling, for two features (left chart). Also the marginal distributions (stacked) for each feature (center and right chart). Blue circles are samples of class 1, orange circles are samples of class 2, and green circles are samples of class 3. Source code in extract 20.

The validation dataset is a 30% of the original data-set, obviously it is not oversampled. The number of samples per class is much lower than in the training data-set, as it is evident in figure 27.

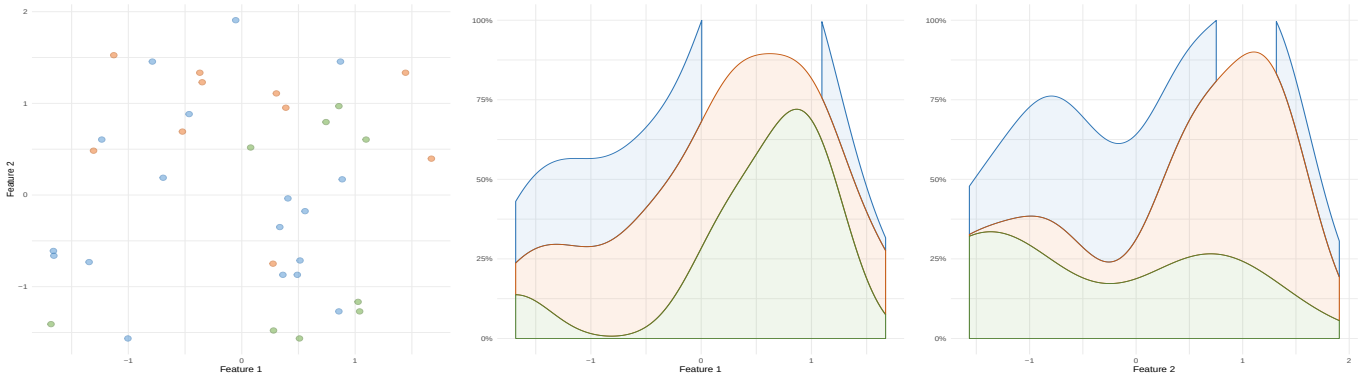


Figure 27: Scatter plot representing the validation data-set samples for two features (left chart). Also the marginal distributions (stacked) for each feature (center and right chart). Blue circles are samples of class 1, orange circles are samples of class 2, and green circles are samples of class 3. Source code in extract 20.

As commented, the package Caret has a convenient way of training models; below there is the training configuration for all the different algorithms. This configuration, training control variable, is passed to every algorithm during training initialization as a parameter. The configuration is the following:

```
36 > #Sets the training type: a repeated k-fold cross-validation.
37 > #With 20 repetitions and 3 folds (2 for training and 1 for test)
38 > K <- 3; rep <- 20
39 > C <- which(names(training)=='cluster')
40 > trainControl <- trainControl(method="repeatedcv", number=K,repeats=rep,
41 >                               savePredictions="final",
42 >                               classProbs=TRUE,
43 >                               verbose=FALSE,
44 >                               allowParallel=TRUE,
45 >                               index=createMultiFolds(training[,C],k = K,times = rep))
```

Source Code Extract 1: Configuration for each classification model.

As shown above, in this thesis, the training configuration is a "repeated k-fold cross-validation" with three folds and twenty repetitions. The k-fold cross validation training works by dividing the training set in a k-number of partitions. One of the partitions is selected to act as test data; the others are shuffled and used for training. In each iteration, the model is trained against the K-1 folds. When it is finished, the performance is tested against the testing fold (the one selected beforehand). This process iterates over all the k-folds, so in each iteration, the testing fold is a different one. This thesis uses a repeated version of the k-fold cross validation; it repeats the entire process as many times as requested, in this case, twenty times. The whole process is shown in figure 28. Notice the training set is randomly reshuffled in each repetition (done in source line 51).

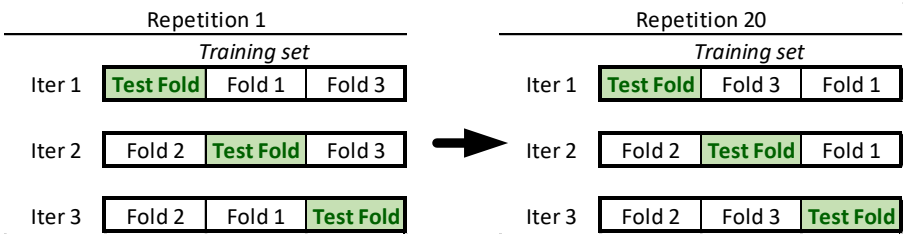


Figure 28: Repeated K-fold cross validation process with 3 folds and 20 repetitions.

For this thesis, there are just three folds due to the low number of real instances; selecting a higher number of folds would imply that the testing fold might not have real samples of some classes. To compensate for the low number of iterations (three), it is repeated twenty times. Notice that the synthetic samples are there to reduce the effect that an imbalance data-set has over the training process. If one of the classes has much more samples than other ones, during the training process, the algorithm will learn better this class than the others. With synthetic

samples, the performance of the algorithm might not increase, but minority classes have more chances to be learned by the algorithm.

The whole training process is available in source code extract 16

8.2.1 Classification evaluation

With the three models already trained, they can be used to predict the classes of the validation data set. These predictions and the original classes (validation data) can be used to get the performance information of the model (source code extract 17); i.e., the confussino matrix per model.

The results for the k-nearest neighbour model follows are shown in figure 29:

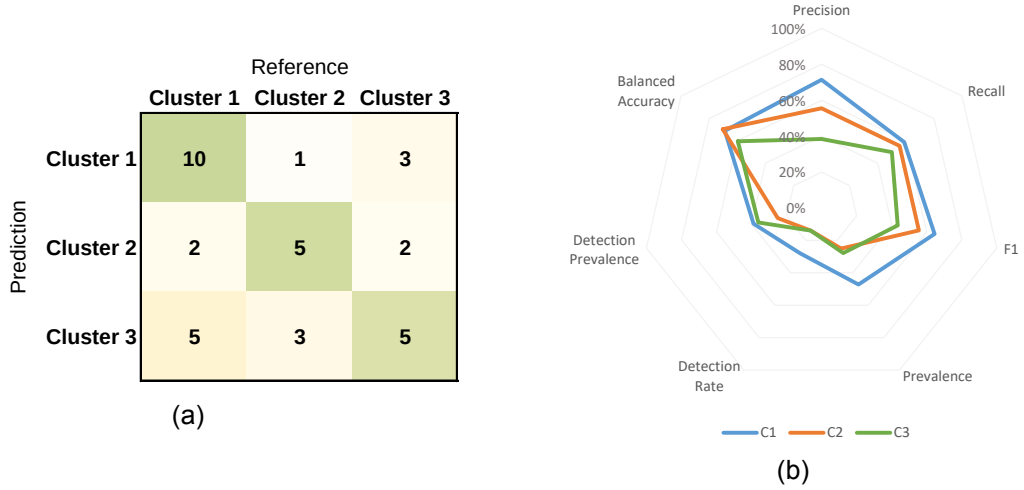


Figure 29: Confussion matrix (a) and performance statistics or KNN model.

The accuracy for every class is above 33%, so the model performs better than a random classifier for any class. Though, the model is clearly biased towards the first cluster. The results for the CART algorithm are shown in figure 30

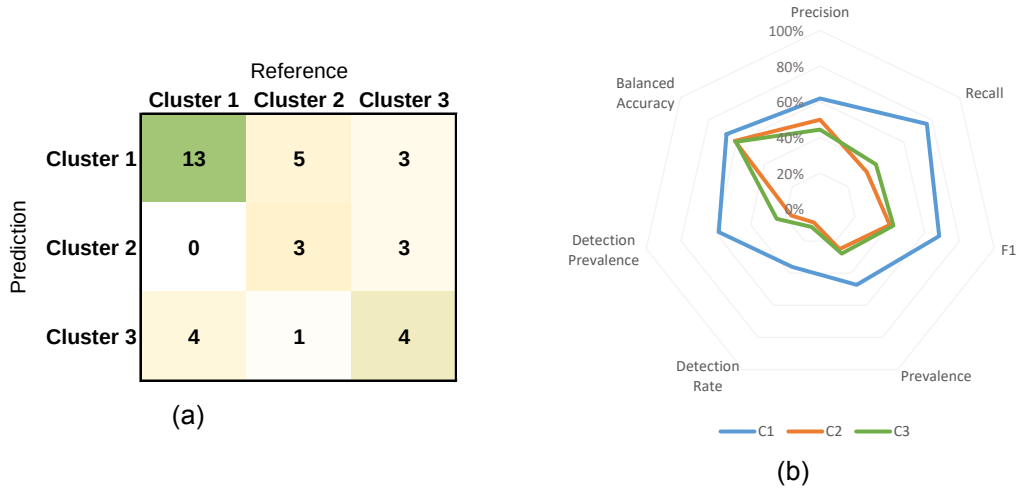


Figure 30: Confussion matrix (a) and performance statistics or CART model.

The CART algorithm performs similarly as the KNN but also increasing the performance in cluster three. The results for the algorithm Random Forest (RF) are shown in figure 31:

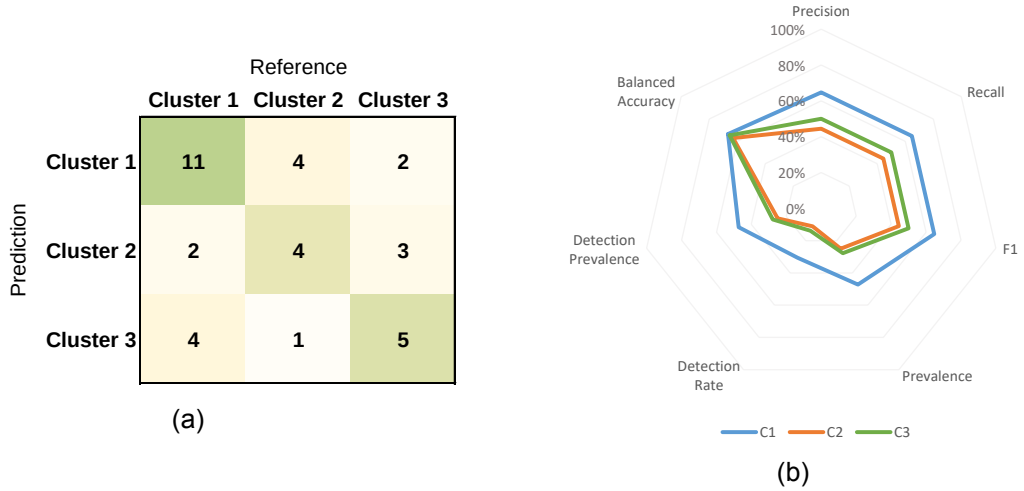


Figure 31: Confussion matrix (a) and performance statistics or RF model.

The RF algorithm is the most balanced of the three, increasing the precision of class three up to 50%. It is possible to train the RF model with binary data, i.e., samples belonging to two clusters. With a binary model, it is possible to plot the gain, and the lift curve (Wiki, 2019) and directly compare the performance of the model with a random classifier. The code that iterates over the three classes of the samples and trains three RF binary models is available in the extract 18. Figures 32, 33 and 34 shows that any binary model based on the Random Forest algorithm can perform better than a random classifier; even with two classes instead of three (original number of clusters found by the first task), this is done by aggregating two of the three classes in one, resulting in just two classes.

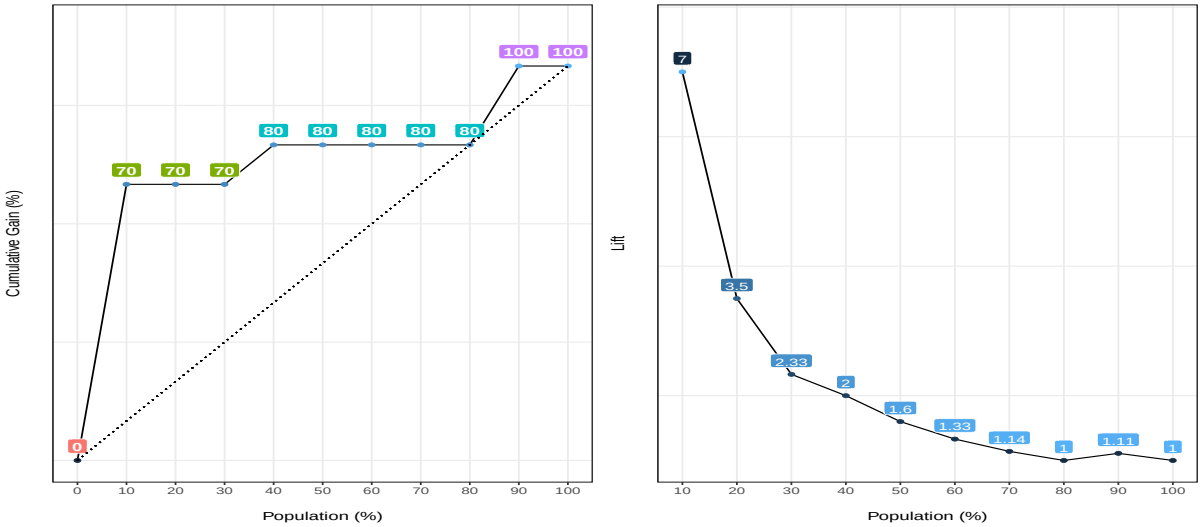


Figure 32: Cumulative gain curve and fit curve for RF binary model, where class 1 = cluster 1, and class 2 = cluster 2 and cluster 3.

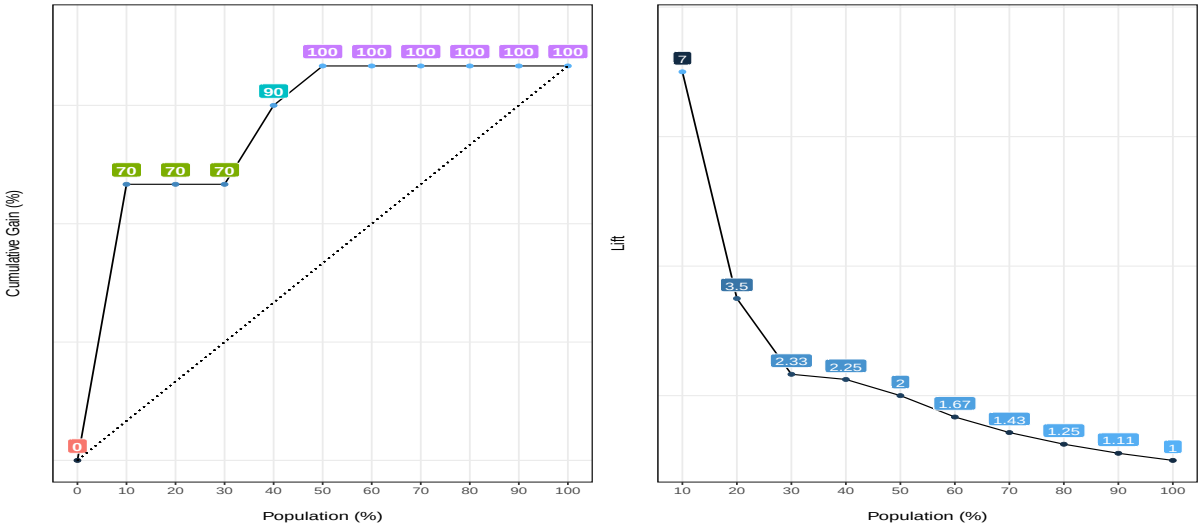


Figure 33: Cumulative gain curve and fit curve for RF binary model, where class 1 = cluster 2, and class 2 = cluster 1 and cluster 3.

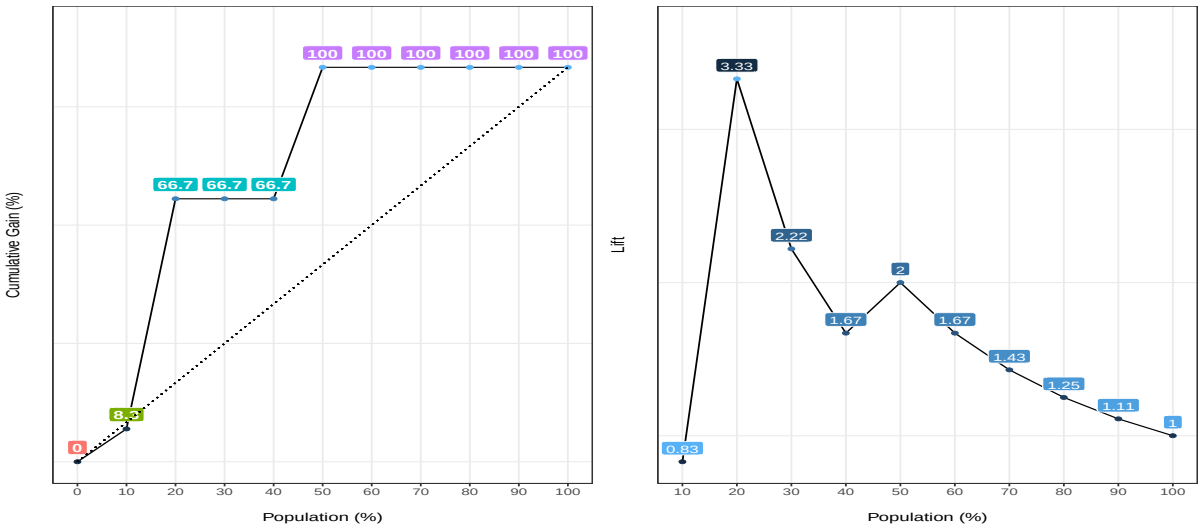


Figure 34: Cumulative gain curve and fit curve for RF binary model, where class 1 = cluster 3, and class 2 = cluster 1 and cluster 2.

Finally, to get descriptive information as commented in phase two, it is possible to inspect which features are the most important ones for each model; i.e., what are the weights of each input variable in the model. Below, only the top ten variables are shown and sorted by importance.

```
[1] "Españoles por hogar"          "Tamaño medio hogar"
[3] "Extranjeros por hogar"        "Densidad (Hab./Km2)"
[5] "Estudios superiores no univ." "Residencias"
[7] "Superficie construida total"  "NacMuj"
[9] "Superficie (m2)"              "titulados medios"
```

Table 6: Top ten imporant predictors for the Random Forest model. Source code available in extract 19

```
[1] "Españoles por hogar"          "Densidad (Hab./Km2)"
[3] "Extranjeros por hogar"        "Precio Medio Vivienda"
[5] "Tamaño medio hogar"           "Superficie construida total"
[7] "Año construcción(promedio)"   "Estudios superiores no univ."
[9] "Superficie (m2)"              "Residencias"
```

Table 7: Top ten imporant predictors for the CART model. Source code available in extract 19

```
[1] "Superficie (m2)"                  "Precio Medio Vivienda"
[3] "Superficie Viviendas(m2)"         "Españoles y Extranjeros por hogar"
[5] "Edad promedio"                    "Extranjeros por hogar"
[7] "Españoles por hogar"              "Superficie construida media"
[9] "Año construcción(promedio)"       "Tamaño medio hogar"
```

Table 8: Top ten imporant predictors for the KNN model. Source code available in extract 19

There are interesting findings based on the most important attributes in the classification models.

- After all, population size has, indirectly, something to say in the supply-demand curve behavior. The density of the ward and the area of the ward are important characteristics, not the highest ones, but they do help to define the classes.
- It is clear that the size and composition of the family household are important.
- Type of the residencies also matter. Their size and year of construction appear in all the results.
- Lastly, and quite interesting, the size of the population without a university degree also matters, i.e., people with a high school diploma ("*bachillerato*" in Spain).

There are two options after getting to the end of phase four. The first option is to incorporate the models and results in the company's IT assets and produce a visualization to finish the strategic external analysis. This will be done in phase five by launching a project to move the data and models to production systems. Moreover, in phase six, by producing the corresponding visualization. The visualization would contain the attributes mentioned, or an aggregation of them; also their cluster assignment. The visualization must provide an easy way to understand these properties for each ward, so the decision to select one location is taken with better information.

The second option, and in this case the right option, is to iterate again from phase one with this new information. In this case, the business strategy technique will be re-assessed so, based on the supply-demand curve, new indicators of "rivalry" might be included. The new analysis should account for the wards' characteristics found: type and size of the family household, the density of the ward, population's level of studies, and the type of residences built. With

these new variables, a new strategic external analysis can be performed, more focused and connected to the area that is under analysis.

The PoC proves that it is possible to implement an algorithm that answers, or even enhances, the questions derived from the strategic methodology. Indeed, phase two of the DDSD methodology translates such high-level questions into information (meta-features) and tasks to work on such information. Phase two, feature engineering, stresses the importance of keeping the strategic logic (domain knowledge) by selecting and transforming the appropriate information into data and tasks. Here, the supply-demand curve was translated into a data-set linked to different tasks targeted to achieve the goal set by the strategy: to locate an advantageous location by inspecting the location's supply-domain behavior. The PoC identified three different groups of locations (within Madrid) with different types of supply-demand behaviors, it also provided several characteristics of the city ward that have a proven relationship with the identified supply-demand behaviors.

Chapter 9

DDSD assessment

Based on the specific objectives of this thesis (section 2.3). The following sections compose a qualitative assessment of the methodology implemented through the proof of concept.

9.1 Implementation of an algorithm, or several, that answers the questions proposed by the selected strategic methodology.

The results of the PoC developed in this thesis following the novel methodology proposed (section 8.2.1), show that it is possible. Indeed, two different machine learning tasks were defined and implemented to tackle the specific goal defined during the initiation of the strategic external analysis. Both tasks used different data-sets, and same target classes, so that the second task could validate the first one. The performance of the classification task validates the unsupervised clustering task and the whole process itself. Also, the second data-set used in the clustering task, from the data-source Camerdata, indicates that the supply-demand patterns found during the unsupervised clustering, have an economic impact on businesses based on their location; i.e., depending on the cluster (the group of wards) they are assigned.

9.2 Comparison of the obtained results with the results coming from a classical approach

The results obtained by the PoC are easily comparable with possible results coming from a classical approach. Indeed, it is easy to see the added value. Given the small size of the area under study, there are not many options for the analysis of the market's rivalry (under the same budget). So, based on the external analysis used, market rivalry, a classical approach could have been an inspection of the ratio "bars & restaurants to population", for each zone of

Madrid. This ratio would be used together with the analysis of other factors, like the threat of entrants, substitutes, etc. With a data-driven approach and the same available information, the produced outcome is much more prolific and meaningful. Just with one iteration of the process, the locations can be grouped by supply-demand behavior based on common characteristics of the locations that shape that behavior. Indeed, the resulting characteristics found, type and size of the family household, the density of the ward, population's level of studies, or the type of residences built, open new possibilities for defining a new external strategic analysis. In short, there is more and better information to get an educated a decision.

9.3 Enhancement of the strategic methodology

The data-driven approach shapes and enhances strategic methodology. The first iteration of DDSD process unveils relationships between the wards' properties and the demand-supply behavior utterly unknown during the definition of external strategic analysis. For instance, the analysis of the possible relationship between the number of bars and restaurants with the number of undergraduates in an area. The data-driven approach opens endless possibilities to enhance a strategic analysis, in this case, the external analysis "market rivalry".

9.4 Reduction of the risk, or the uncertainty associated with traditional strategic methodologies.

The uncertainty, coupled with generalist processes like the strategic ones, is reduced. The data-driven process yields results pertinent to the data provided. Although the data models are based on the logic of the domain knowledge (law of supply and demand), their validation can provide a certain grade of assurance on the validity of the results. Indeed, in this PoC, the validation of the data models implies that there are three different supply-demand behaviors for Madrid. This does not mean this is true for other regions of Spain or countries, but the DDSD provides a certain level of assurance that this is the case for the selected area.

Chapter 10

Conclusions and future work

This thesis shows a successful integration of a Business Strategy technique with Big Data techniques. During the analysis of both fields, section 3.1, two obstacles related to Business Strategy techniques (based on Monitors Group's bankruptcy analysis) are described: the lack, or poor, domain knowledge in the business model; and cognitive bias during the implementation of the models.

The methodology proposed mitigates these two obstacles by providing, with a structured manner, the incorporation of data-driven techniques. Indeed, the data structures discovered with the PoC unveiled relationships in the target subject of the business model that, initially, were hidden. This is the case of the relationship of the average level of studies in a city ward and the supply-demand behavior of its bars (and restaurants).

A third obstacle, also mentioned during the analysis of both fields, is the negative impact of data quality in the implementation of data-driven processes at management level (Vidgen, Shaw, & Grant, 2017). The implementation of the PoC highlight this obstacle. The lack of relevant data forced the implementation to take several assumptions and countermeasures. One of them is assuming that the food and beverage industry represents the bars accurately. There is no data, in the data source selected, with enough granularity to select data exclusively relevant to bars, and not restaurants. The inclusion of synthetic samples in the classification task is another proof of the lack of quality of the data-sets.

As future work, it is clear that to get better data, a more thorough and extensive phase three is needed. Indeed, this is a bottleneck for the whole process. One possible remediation to this, is the implementation, as a data service in the company, of a data-lake with the scope already set in, principally, external data sources that might contain useful data. Within the context of this thesis, a data-lake containing historical data of all bars created, and ceased, during the last years, labeled with geolocation data, would increase not only the amount of data but also

its quality. That is possible with web scrapping (Wikipedia, 2019l) techniques in official public documents, but the amount of work involved might be more significant than implementing the rest of the DDSD phases combined.

Bibliography

Akerlof, George A and Robert J Shiller (2010). *Animal spirits: How human psychology drives the economy, and why it matters for global capitalism*. Princeton university press.

Anagolum, Sashwat (2019). *Quantum machine learning: distance estimation for k-means clustering*. URL: https://is.gd/dBuOCo.

Appelbaum, Deniz et al. (2017). "Impact of business analytics and enterprise systems on managerial accounting". In: *International Journal of Accounting Information Systems* 25, pp. 29–44.

Arbelaitz, Olatz et al. (2013). "An extensive comparative study of cluster validity indices". In: *Pattern Recognition* 46.1, pp. 243–256.

Armstrong, Jon Scott (2001). *Principles of forecasting: a handbook for researchers and practitioners*. Vol. 30. Springer Science & Business Media. ISBN: 978-0792374015.

Arnott, Robert D et al. (2013). "The Surprising Alpha From Malkiel'sMonkey and Upside-Down Strategies". In: *The Journal of Portfolio Management* 39.4, pp. 91–105.

Ayuntamiento de Madrid (2018a). *Banco de datos: Consulta de series estadísticas*. URL: https://is.gd/NvHxSs.

— (2018b). *Portal de datos abiertos del Ayuntamiento de Madrid*. URL: https://is.gd/4ECQR6.

Barbara, Kitchenham and Stuart Charters (2007). "Guidelines for performing Systematic Literature Reviews in Software Engineering". In: 2.

Bloom, N et al. (2013). *Management in America. US Census Bureau Center for Economic Studies, Paper No*. Tech. rep. CES-WP-13-01. URL: https://is.gd/ygx37p.

Brownlee, Jason (2013). *A Tour of The Most Popular Machine Learning Algorithms*. URL: https://is.gd/OXGNab.

Brynjolfsson, Erik and Kristina McElheran (2016). "Data in Action: Data-Driven Decision Making in US Manufacturing". In: URL: https://is.gd/7eC80q.

Cadle, James, Debra Paul, and Paul Turner (2010). *Business analysis techniques: 72 essential tools for success*. BCS, The Chartered Institute.

Camerdata (2018). *Base de datos de empresas Españolas*. URL: https://is.gd/vKjiig.

Chawla, Nitesh V et al. (2002). "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16, pp. 321–357.

Cooper, Michael J, Huseyin Gulen, and P Raghavendra Rau (2016). "Performance for pay? The relation between CEO incentive compensation and future stock price performance". In: *The Relation Between CEO Incentive Compensation and Future Stock Price Performance (November 1, 2016)*.

Creamer, Germán and Yoav Freund (2010). "Learning a board Balanced Scorecard to improve corporate performance". In: *Decision Support Systems* 49.4, pp. 365–385.

Denning, Steve (2012). "What Killed Michael Porter's Monitor Group? The One Force That Really Matters". In: *Forbes*. URL: https://is.gd/hJoaoO.

Estadística, Instituto de (2016). *Códigos postales, distritos y barrios del municipio de Madrid*. URL: https://is.gd/tqwG94.

Gerdeman, Dina (2017). "Companies Love Big Data But Lack the Strategy To Use It Effectively". In: *Harvard Business School Working Knowledge*. URL: https://is.gd/3Lv1gI.

Giorgino, Toni et al. (2009). "Computing and visualizing dynamic time warping alignments in R: the dtw package". In: *Journal of statistical Software* 31.7, pp. 1–24.

Gusenbauer, Michael (2019). "Google Scholar to overshadow them all? Comparing the sizes of 12 academic search engines and bibliographic databases". In: *Scientometrics* 118.1, pp. 177–214. ISSN: 1588-2861. URL: https://is.gd/5whSEM.

Haimowitz, Ira J and Henry Schwarz (1997). "Clustering and prediction for credit line optimization". In: *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pp. 29–33.

Kahneman, Daniel (2011). *Thinking, fast and slow*. Macmillan. ISBN: 978-0374275631.

Kim, Minho and RS Ramakrishna (2005). "New indices for cluster validity assessment". In: *Pattern Recognition Letters* 26.15, pp. 2353–2363.

Kimball, Ralph (1996). *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*. New York, NY, USA: John Wiley & Sons, Inc. ISBN: 0-471-15337-0.

Kokina, Julia, Dessislava Pachamanova, and Andrew Corbett (2017). "The role of data visualization and analytics in performance management: Guiding entrepreneurial growth decisions". In: *Journal of Accounting Education* 38, pp. 50–62.

Kuhn, Max (2008). “Building Predictive Models in R Using the caret Package”. In: *Journal of Statistical Software, Articles* 28.5, pp. 1–26. ISSN: 1548-7660. URL: https://is.gd/Adu2gI.

LaValle, Steve et al. (2011). “Big data, analytics and the path from insights to value”. In: *MIT sloan management review* 52.2, p. 21. URL: https://is.gd/HQthUQ.

L.G. (2012). “Monitor’s end”. In: *The Economist*. URL: https://is.gd/mhlviZ.

Meulen, R van der (2016). “Gartner survey reveals investment in Big Data is up but fewer organizations plan to invest”. In: *Retrieved on December* 18, p. 2016. URL: https://is.gd/iIxIUP.

Navas López, José Emilio and Luis Ángel Guerras Martín (2016). *Fundamentos de dirección estratégica de la empresa*. 2nd ed. Civitas.

Parlament, European and European Council (2016). *General Data Protection Regulation*. URL: https://is.gd/gEA8RS.

Porter, Michael E (1989). “How competitive forces shape strategy”. In: *Readings in strategic management*. Springer, pp. 133–143. URL: https://is.gd/SMdbpR.

Reuters (2012). “UPDATE 1-Monitor Company files for Chapter 11; Deloitte to buy assets”. In: *Reuters*. URL: https://is.gd/546WAd.

Rousseeuw, Peter J (1987). “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of computational and applied mathematics* 20, pp. 53–65.

Saitta, Sandro, Benny Raphael, and Ian FC Smith (2007). “A bounded index for cluster validity”. In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, pp. 174–187.

Schmarzo, Bill (2017). “5 Steps to Building a Big Data Business Strategy”. In: *Dell EMC InFocus*. URL: https://is.gd/a259H9.

Silver, Nate (2012). *The signal and the noise : why so many predictions fail– but some don’t*. New York: Penguin Press. ISBN: 978-0141975658.

Thaler, Richard H and LJ Ganser (2015). *Misbehaving: The making of behavioral economics*. WW Norton New York, NY.

Vidgen, Richard, Sarah Shaw, and David B Grant (2017). “Management challenges in creating value from business analytics”. In: *European Journal of Operational Research* 261.2, pp. 626–639. URL: https://is.gd/HfbgXh.

Weill, P and SL Woerner (2013). “The Next-Generation Enterprise: Thriving in an increasingly digital ecosystem”. In: *Center for Information Systems Research, Sloan School of Management, Cambridge, MA: Massachusetts Institute of Technology. Research Briefing* 13.4.

Wiki, ML (2019). *Cumulative Gain Chart*. URL: https://is.gd/vI5Adu.

Wikipedia (2019a). *Ceteris paribus*. URL: https://is.gd/GE5uBD.

— (2019b). *Cross-industry standard process for data mining*. URL: https://is.gd/DMcSjT.

— (2019c). *Domain knowledge*. URL: https://is.gd/mhsHUq.

— (2019d). *Dynamic time warping*. URL: https://is.gd/UJ8vV2.

— (2019e). *Ensemble forecasting*. URL: https://is.gd/E2m2vy.

— (2019f). *Euclidean distance*. URL: https://is.gd/OsimIB.

— (2019g). *Feature engineering*. URL: https://is.gd/nChL8l.

— (2019h). *Law of supply*. URL: https://is.gd/yQW1kT.

— (2019i). *Stochastic process*. URL: https://is.gd/k8Zlhq.

— (2019j). *Supply and demand*. URL: https://is.gd/FtF9LF.

— (2019k). *Unsupervised learning*. URL: https://is.gd/Hx9Br4.

— (2019l). *Web scraping*. URL: https://is.gd/FX2UyK.

Woerner, Stephanie L and Barbara H Wixom (2015). “Big data: extending the business strategy toolbox”. In: *Journal of Information Technology* 30.1, pp. 60–62.

Appendices

Appendix A

Systematic search

Following, all queries performed in the systematic search [1]:

Table 9: Systematic search queries

PESTEL	
Google Scholar	allintitle: PESTEL ''Machine Learning'' allintitle: PESTEL ''Big Data''
IEEXplore	((''Big Data'') OR (''Machine Learning'')) AND (PESTEL))
Elsevier	TITLE-ABSTR-KEY(''Machine Learning'') AND TITLE-ABSTR-KEY(PESTEL) TITLE-ABSTR-KEY(''Big Data'') AND TITLE-ABSTR-KEY(PESTEL)

Porter (*Porter's Five Forces*)	
Google Scholar	allintitle: Porter ''Machine Learning'' allintitle: Porter ''Big Data''
IEEXplore	((''Big Data'') OR (''Machine Learning'')) AND (Porter))
Elsevier	TITLE-ABSTR-KEY(''Machine Learning'') AND TITLE-ABSTR-KEY(Porter) TITLE-ABSTR-KEY(''Big Data'') AND TITLE-ABSTR-KEY(Porter)

SWOT	
Google Scholar	allintitle: SWOT ''Machine Learning'' allintitle: SWOT ''Big Data''
IEEXplore	((''Big Data'') OR (''Machine Learning'')) AND (SWOT))
Elsevier	TITLE-ABSTR-KEY(''Machine Learning'') AND TITLE-ABSTR-KEY(SWOT) TITLE-ABSTR-KEY(''Big Data'') AND TITLE-ABSTR-KEY(SWOT)

[1] Notice some keywords have been abbreviated in order to avoid false negatives in the searches, but at the expense of getting more false positives

Andoff	
Google Scholar	allintitle: Andoff ''Machine Learning'' allintitle: Andoff ''Big Data''
IEEXplore	((''Big Data'') OR (''Machine Learning'')) AND (Andoff))
Elsevier	TITLE-ABSTR-KEY(''Machine Learning'') AND TITLE-ABSTR-KEY(Andoff) TITLE-ABSTR-KEY(''Big Data'') AND TITLE-ABSTR-KEY(Andoff)

Strategy Map	
Google Scholar	allintitle: ''Strategy Map'' ''Machine Learning'' allintitle: ''Strategy Map'' ''Big Data''
IEEXplore	((''Big Data'') OR (''Machine Learning'')) AND (''Strategy Map''))
Elsevier	TITLE-ABSTR-KEY(''Machine Learning'') AND TITLE-ABSTR-KEY(''Strategy Map'') TITLE-ABSTR-KEY(''Big Data'') AND TITLE-ABSTR-KEY(''Strategy Map'')

POPIT (*POPIT model*)	
Google Scholar	allintitle: ''POPIT'' ''Machine Learning'' allintitle: ''POPIT'' ''Big Data''
IEEXplore	((''Big Data'') OR (''Machine Learning'')) AND (''POPIT'')
Elsevier	TITLE-ABSTR-KEY(''Machine Learning'') AND TITLE-ABSTR-KEY(''POPIT'') TITLE-ABSTR-KEY(''Big Data'') AND TITLE-ABSTR-KEY(''POPIT'')

Leavitt (*Leavitt's Diamond*)	
Google Scholar	allintitle: ''Leavitt'' ''Machine Learning'' allintitle: ''Leavitt'' ''Big Data''
IEEXplore	((''Big Data'') OR (''Machine Learning'')) AND (''Leavitt'')
Elsevier	TITLE-ABSTR-KEY(''Machine Learning'') AND TITLE-ABSTR-KEY(''Leavitt'') TITLE-ABSTR-KEY(''Big Data'') AND TITLE-ABSTR-KEY(''Leavitt'')

McKinsey (*McKinsey 7-S*)	
Google Scholar	allintitle: ''McKinsey'' ''Machine Learning'' allintitle: ''McKinsey'' ''Big Data''
IEEXplore	((''Big Data'') OR (''Machine Learning'')) AND (''McKinsey'')
Elsevier	TITLE-ABSTR-KEY(''Machine Learning'') AND TITLE-ABSTR-KEY(''McKinsey'') TITLE-ABSTR-KEY(''Big Data'') AND TITLE-ABSTR-KEY(''McKinsey'')

Balanced Scorecard	
Google Scholar	allintitle: ''Balanced Scorecard'' ''Machine Learning'' allintitle: ''Balanced Scorecard'' ''Big Data''
IEEXplore	(((''Big Data'') OR (''Machine Learning'')) AND (''Balanced Scorecard'')
Elsevier	TITLE-ABSTR-KEY(''Machine Learning'') AND TITLE-ABSTR-KEY(''Balanced Scorecard'') TITLE-ABSTR-KEY(''Big Data'') AND TITLE-ABSTR-KEY(''Balanced Scorecard'')

Critical Success Factor	
Google Scholar	allintitle: ''Critical Success Factor'' ''Machine Learning'' allintitle: ''Critical Success Factor'' ''Big Data''
IEEXplore	(((''Big Data'') OR (''Machine Learning'')) AND (''Critical Success Factor'')
Elsevier	TITLE-ABSTR-KEY(''Machine Learning'') AND TITLE-ABSTR-KEY(''Critical Success Factor'') TITLE-ABSTR-KEY(''Big Data'') AND TITLE-ABSTR-KEY(''Critical Success Factor'')

Key Performance Indicator	
Google Scholar	allintitle: ''Key Performance Indicator'' ''Machine Learning'' allintitle: ''Key Performance Indicator'' ''Big Data''
IEEXplore	(((''Big Data'') OR (''Machine Learning'')) AND (''Key Performance Indicator'')
Elsevier	TITLE-ABSTR-KEY(''Machine Learning'') AND TITLE-ABSTR-KEY(''Key Performance Indicator'') TITLE-ABSTR-KEY(''Big Data'') AND TITLE-ABSTR-KEY(''Key Performance Indicator'')

Appendix B

Clustering features' correlations

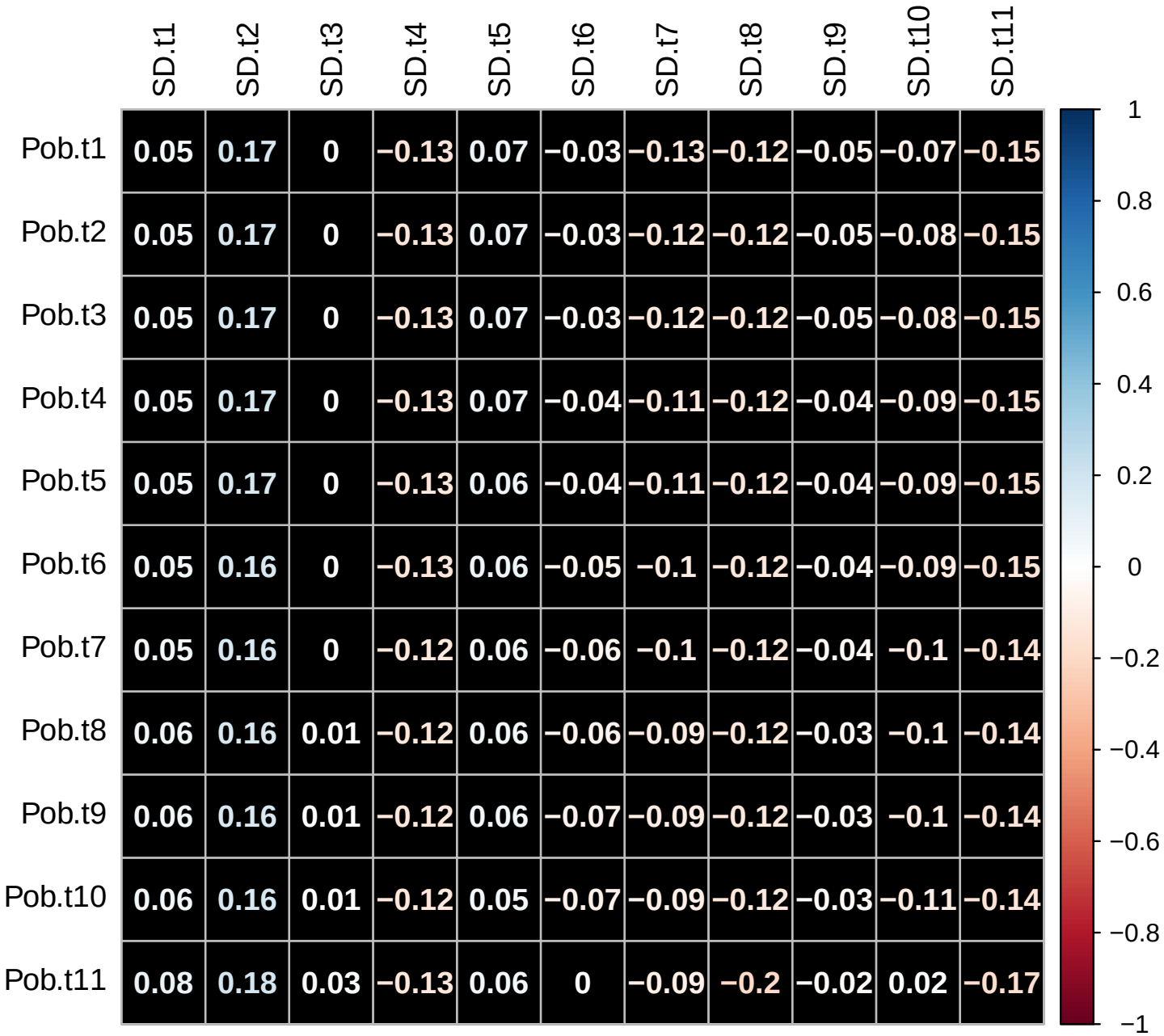


Figure 35: Correlations between supply-demand periods (SD-t1..11), and population (Pob-t1..t11), where $t \in \{Jan2013, Jul2013 ... Jan2018\}$. Source code in extract 8.

Appendix C

Source Code

C.1 Data Preparation

```
> #Loads and insspect cluster data
> library(readr);library(parallel);library(doParallel)
> cl <- makePSOCKcluster(detectCores());registerDoParallel(cl)
>
> clusterdata <- read_csv("./TFM/contenido/datos/clusterdata.csv")
> npredict <- length(clusterdata)-1
> names(clusterdata) <- c("Barrio",1:npredict)
> sink('./TFM/contenido/codigo/output/clustData.txt');
> summary(clusterdata);sink()
```

```
   Barrio                1                  2                  3
Length:126         Min.   :-0.08599   Min.   :-0.24012   Min.   :-0.144093
Class :character   1st Qu.: 0.01182   1st Qu.: 0.01340   1st Qu.: 0.004894
Mode  :character   Median : 0.02016   Median : 0.03152   Median : 0.014498
                   Mean   : 0.02538   Mean   : 0.02998   Mean   : 0.021155
                   3rd Qu.: 0.03243   3rd Qu.: 0.04525   3rd Qu.: 0.027152
                   Max.   : 0.31428   Max.   : 0.21990   Max.   : 0.500922
      4                  5                  6                  7
Min.   :-0.151952   Min.   :-0.3365800   Min.   :-0.072182   Min.   :-0.007671
1st Qu.: 0.008119   1st Qu.:-0.0047935   1st Qu.:-0.002915   1st Qu.: 0.080409
Median : 0.020647   Median : 0.0009245   Median : 0.002371   Median : 0.139726
Mean   : 0.027259   Mean   :-0.0007872   Mean   : 0.001376   Mean   : 0.180074
3rd Qu.: 0.039944   3rd Qu.: 0.0064958   3rd Qu.: 0.007805   3rd Qu.: 0.215600
Max.   : 0.466664   Max.   : 0.0881720   Max.   : 0.037335   Max.   : 1.110170
      8                  9                  10                 11
Min.   :-0.29018   Min.   :-0.21588   Min.   :-0.188854   Min.   :-0.940118
1st Qu.:-0.07232   1st Qu.: 0.03803   1st Qu.:-0.008803   1st Qu.:-0.015714
Median :-0.02810   Median : 0.07243   Median : 0.006659   Median :-0.001632
Mean   :-0.02278   Mean   : 0.09079   Mean   : 0.004393   Mean   : 0.060605
3rd Qu.: 0.02336   3rd Qu.: 0.13055   3rd Qu.: 0.021987   3rd Qu.: 0.013668
Max.   : 0.32698   Max.   : 0.52099   Max.   : 0.129561   Max.   : 4.163204
```

Source Code Extract 2: Preview of the clustering data values.

```
> #Missing Values
> sink('./TFM/contenido/codigo/output/missVal.txt');
> which(is.na(clusterdata)==TRUE);sink()
```

```
integer(0)
```

Source Code Extract 3: Missing values in the clustering data

```
> #Plotting raw time series data
> library(tidyr);library(dplyr);library(ggplot2)
>
> clusterdata_long <- gather(clusterdata,period,val,2:12)
> clusterdata_long$period <- sapply(clusterdata_long$period,as.numeric)
> upp_stdvar <- mean(clusterdata_long$val) + 3*sd(clusterdata_long$val)
> low_stdvar <- mean(clusterdata_long$val) - 3*sd(clusterdata_long$val)
>
> plot1 <- clusterdata_long %>% ggplot(aes(x=period,y=val,colour=Barrio)) + geom_line() +
    ↪  geom_hline(yintercept = c(upp_stdvar,low_stdvar), color = "red", linetype ="dashed")+
    ↪  geom_hline(yintercept = 0, color = "black",size=0.2) + theme(legend.position = "none"
    ↪  ,axis.text.x=element_blank(),axis.ticks.x=element_blank(),panel.grid.major = element_blank
    ↪  (),panel.grid.minor = element_blank(),panel.background = element_blank(),axis.line.y =
    ↪  element_line(color="black", size = 0.1),axis.title.x = element_blank()) + geom_text(aes( x
    ↪  =3, y=(upp_stdvar), label = "3sd.", vjust = -1), size = 4) + geom_text(aes( x=3, y=
    ↪  (low_stdvar), label = "-3sd.", vjust = -1), size = 4)
> ggsave("supplyDemanDataSeries.pdf", plot = plot1, device = "pdf", path=
    ↪  "./TFM/contenido/imagenes",dpi=600)
>
> #Cleaning outliers for time series (package forecast)
> library(forecast)
> clusterdata[,-1] <- sapply(clusterdata[,-1], tsclean)
> clusterdata_long <- gather(clusterdata,period,val,2:12)
> clusterdata_long$period <- sapply(clusterdata_long$period,as.numeric)
> upp_stdvar <- mean(clusterdata_long$val) + 3*sd(clusterdata_long$val)
> low_stdvar <- mean(clusterdata_long$val) - 3*sd(clusterdata_long$val)
>
> plot2 <- clusterdata_long %>% ggplot(aes(x=period,y=val,colour=Barrio)) + geom_line() +
    ↪  geom_hline(yintercept = c(upp_stdvar,low_stdvar), color = "red", linetype ="dashed")+
    ↪  geom_hline(yintercept = 0, color = "black",size=0.2) +   theme(legend.position = "none"
    ↪  ,axis.text.x=element_blank(),axis.ticks.x=element_blank(),panel.grid.major = element_blank
    ↪  (),panel.grid.minor = element_blank(),panel.background = element_blank(),axis.line.y =
    ↪  element_line(color="black", size = 0.1),axis.title.x = element_blank()) + geom_text(aes( x
    ↪  =3, y=(upp_stdvar), label = "3sd.", vjust = -1), size = 4) + geom_text(aes( x=3, y=
    ↪  (low_stdvar), label = "-3sd.", vjust = -1), size = 4)
> ggsave("supplyDemanDataSeriesNorm.pdf", plot = plot2, device = "pdf", path=
    ↪  "./TFM/contenido/imagenes",dpi=600)
```

Source Code Extract 4: Visualization of the clustering data.

```
> #Load classification data and convert to data.frame
> #(caret package needs data.frame objects)
> classdata <- read_csv("./TFM/contenido/datos/classdata.csv")
> classdata <- as.data.frame(classdata)
> #Missing values
> sink('./TFM/contenido/codigo/output/missVal2.txt')
> which(is.na(classdata)==TRUE);sink()
```

```
integer(0)
```

Source Code Extract 5: Missing values in the classification data.

```
> #Scaling and centering data
> classdata[,-1] <- scale(classdata[,-1],center = TRUE, scale = TRUE)
> sink('./TFM/contenido/codigo/output/scaledClassData.txt')
> head(describe(classdata,skew = FALSE,omit=TRUE)[,c(1,3,4,7)]); sink()
```

```
                          vars mean sd range
Superficie (m2)              2    0  1 10.68
Superficie Viviendas(m2)     3    0  1  6.61
Residencias                  4    0  1  8.61
Año construcción(promedio)   5    0  1  5.37
Superficie construida total  6    0  1  8.81
Superficie construida media  7    0  1  7.53
```

Source Code Extract 6: Classification data-set after normalization (scaling and centering). Only first six (out of forty) variables shown.

```
> #Check for constant feautures (near zero variance)
> library(caret)
> nz <- nearZeroVar(classdata[,-1],saveMetrics = TRUE)
> sink('./TFM/contenido/codigo/output/nz.txt')
> which(nz[,3]==TRUE | nz[,4]==TRUE); sink() #Any zero or near zero?
```

```
integer(0)
```

Source Code Extract 7: Zero, or near zero, variance attributes in classification data.

```
> #Correlations
> library(corrplo)
> corr <- read.csv('./TFM/contenido/datos/Pob-DemandSup.csv')
> M <- cor(corr)[-c(1:11),-c(12:22)]
> pdf(file = "./TFM/contenido/imagenes/corrplot.pdf")
> corrplot(M, tl.col = "black",method='number',bg='black')
> dev.off()
```

Source Code Extract 8: Correaltions between Population and Supply-Demand.

C.2 Clustering task

```
> #### Clustering #####
> #Optimal number of clusters
> library(dtwclust);library(parallel);library(doParallel);library(dplyr);library(tidyr)
> #Around 300MB per cpu-kernel (task)
> cl <- makePSOCKcluster(detectCores())
> registerDoParallel(cl)
>
> kmax=10  #Max number of cluster to generate.
> window = 3L #Time window size to compare between time series (DTW)
> clusterdata <- readRDS('./TFM/contenido/datos/clusterdata.rds')
> npredict <- length(clusterdata)-1
>
> clusters <-  tsclust(clusterdata[,-1],type="partitional",centroid="pam", k=2L:kmax, distance =
    ↪  "dtw2",args = tsclust_args(dist = list(window.size = window)))
> cvis <- sapply(clusters, cvi,type="internal")
> sink('./TFM/contenido/codigo/output/cvis.txt')
> cvis; sink()
```

```
              [,1]       [,2]       [,3]        [,4]        [,5]        [,6]
Sil      0.3339872  0.2812678  0.1681663  0.12477461  0.11343660  0.08811310
SF       0.5568160  0.5114286  0.4708386  0.45733102  0.42553928  0.38581059
CH     107.6495161 62.7994773 39.4749810 35.71959054 30.21752287 25.38438886
DB       1.1631345  1.3432993  1.7987190  1.89569867  1.78526599  2.00749408
DBstar   1.1631345  1.3942372  2.0499474  2.50272205  2.28698953  2.89413220
D        0.1223506  0.1223506  0.1181293  0.08903741  0.09124802  0.09315421
COP      0.3483763  0.3216721  0.3004398  0.27926434  0.25983392  0.25924713
              [,7]        [,8]       [,9]
Sil      0.1560663  0.11604912  0.1008240
SF       0.3837939  0.33989390  0.3649288
CH      22.0046414 20.31039896 16.7271912
DB       1.3780134  1.53221041  1.6008030
DBstar   1.9343415  2.36471831  2.8915435
D        0.1239482  0.08591414  0.1010299
COP      0.2413590  0.25111010  0.2410971
```

Source Code Extract 9: CVI for nine different clusterization of the clustering data-set, with K=2 to K=10.

Notice in the code above, the library “doParallel” and “parallel” are used. This decreased drastically the time needed to perform all the clusterizations, especially during the ensemble clustering (extract 11). The amount of parallel processes is calculated based on the number of cores. Also, the amount of memory needed to parallelize the task depends on the memory used for each parallel task that, in turn, depends on the size of the dataset and the algorithm used[1].

[1] For this thesis, 24 CPU cores and 21GB RAM were used

```
> #Plot results.
> #To maximize: Silhouette, Dunn, Calinski-Harabasz, Score Function
> #To minimize: COP, Davies-Bouldin, Modified Davies-Bouldin
> library(ggplot2);library(cowplot);library(scales);library(RColorBrewer)
>
> cvis_long <- as.data.frame(cvis)
> names(cvis_long) <- c(rep(1:(kmax-1),1))
> cvis_long <- scale(cvis_long,center=FALSE)
> cvis_long <- as.data.frame(cvis_long)
> cvis_long <- gather(cvis_long,K,val)
> cvis_long$K <- as.numeric(cvis_long$K)+1
> cvis_long$index <- rep(rownames(cvis),(kmax-1))
> ps <- list()
> to_max <- data.frame('Sil'='Silhouette','D'='Dunn','CH'='Calinski-Harabasz','SF'=
    ↪  'Score Function')
> for(i in 1:4) ps[[i]] <- cvis_long %>% filter(index == names(to_max)[i]) %>% ggplot(aes(x=K,y
    ↪  =val)) + geom_line(show.legend=FALSE,color='blue',size=2) + ylab('') + xlab(as.character
    ↪  (to_max[[i]]))+ xlim(c(2,10))
> to_min <- data.frame('COP'='COP','DB'='Davies-Bouldin','DBstar'='Modified Davies-Bouldin')
> for(i in 1:3) ps[[i+4]] <- cvis_long %>% filter(index == names(to_min)[i]) %>% ggplot(aes(x=K,y
    ↪  =val)) + geom_line(show.legend=FALSE,color='red',size=2) +ylab('') + xlab(as.character
    ↪  (to_min[[i]])) + xlim(c(2,10))
> gridplot <- plot_grid(plotlist=ps,ncol=4)
> ggsave("CVIs.pdf",plot=gridplot,device='pdf',path='./TFM/contenido/imagenes',dpi=400)
```

Source Code Extract 10: Plotting the CVI result.

```
> ### Ensemble clustering for 1000 iterations
> K=3 #Number of clusters.
> N=1000  #Iterations to be performed
>
> #First iteration sets the initial cluster assignations and the centroids
> clusteringA <- tsclust(clusterdata[,-1],type="partitional",centroid="pam", k=K, distance ="dtw2"
    ↪   ,args = tsclust_args(dist = list(window.size = window)))
> CA <- as.data.frame(clusteringA@centroids[1:K])
> names(CA)<-paste("a",1:K,sep="")
>
> #All centroids of the iterations.
> centroids   <- list(k1=data.frame(),k2=data.frame(),k3=data.frame())
> for(j in 1:K) centroids[[j]] <-  rbind(centroids[[j]],t(clusteringA@centroids[[j]]))
>
> n <- length(rownames(clusterdata))
> counter <- as.data.frame(matrix(rep(0,n*K),n))
> names(counter) <- paste("C",1:K,sep="")
>
> #Counting cluster assignation.
> for(i in 1:n) counter[i,clusteringA@cluster[i]] <- +1
> for(i in 1:N){
>         clusteringB <- tsclust(clusterdata[,-1],type="partitional",centroid="pam", k=K, distance
            ↪   ="dtw2",args = tsclust_args(dist = list(window.size = window)))
>         CB <- as.data.frame(clusteringB@centroids[1:K])
>         names(CB)<-paste("b",1:K,sep="")
>         #Who is who? Similarity matrix
>         distances <- dtwDist(mx = rbind(t(CA)[1:K,],t(CB)[1:K,]),distance="dwt")
>         distances <- distances[1:K,(K+1):(2*K)]
>         mapping <- c(1:K)
>         for(j in 1:K){
>                 map <- which(distances==min(distances),arr.ind = TRUE)
>                 mapping[map[1]] <- map[2]
>                 distances[map[1],] <- Inf
>                 distances[,map[2]] <- Inf
>         }
>         #Centroids saved for plotting later.
>         for(j in 1:K) centroids[[j]] <- rbind(centroids[[j]],t(clusteringB@centroids[[mapping[[j
            ↪   ]]]]))
>
>         #Nighborhoods clusters assignments counter
>         for(j in 1:n)
>                 counter[j,mapping[[clusteringB@cluster[j]]]] <- counter[j,mapping[[clusteringB
                    ↪   @cluster[j]]]]+1
>         #Recalculate the averaged centroid (weighted average)
>         for(c in 1:K)
>                 for(r in 1:npredict)
>                         CA[r,c] <- weighted.mean(c(CA[r,c],CB[r,mapping[[c]]]),c(i,1))
> }
> #Cluster assignation. Each sample is assigned to the cluster it has been assigned more times.
> clusterdata$cluster <- rep(0,n)
> for(r in 1:n)
>         clusterdata[r,which(names(clusterdata)=="cluster")] <- which(counter[r,]==max(counter[r
            ↪   ,]))[1]
> saveRDS(clusterdata,'./TFM/contenido/datos/clustresults.rds')
> saveRDS(centroids,'./TFM/contenido/datos/centroids.rds')
> saveRDS(counter,'./TFM/contenido/datos/counter.rds')
> #Stop parallelization
> stopCluster(cl)
> env <- foreach:::.foreachGlobals
> rm(list=ls(name=env), pos=env)
```

Source Code Extract 11: Clustering esemble process.

```
> #Check wards closest to centroids
> distances <- rbind(clusterdata[,-1],avgcentroids$a1) %>% rbind(avgcentroids$a2) %>% rbind
    ↪  (avgcentroids$a3)
> rownames(distances) <- t(cbind(t(clusterdata$Barrio),"ClusterC1","ClusterC2","ClusterC3"))
> distances <- dtwDist(mx = distances,distance="dwt") %>% as.data.frame() %>% select('ClusterC1',
    ↪  'ClusterC2','ClusterC3')
> distances$barrio <- rownames(distances)
> distances <- distances %>% arrange(barrio) %>% filter(!barrio %in% c('ClusterC1','ClusterC2',
    ↪  'ClusterC3'))
> saveRDS(distances,'./TFM/contenido/datos/distances.rds')
```

Source Code Extract 12: Distance matrix between wards' supply-demand curves and the three clusters.

C.3 Classification task

```
> ### Classification ####
> library(caret); library(randomForest); library(e1071); library(parallel); library(doParallel);
    ↪  library(dplyr)
> #Needs around 300MB RAM per core
> cl <- makePSOCKcluster(detectCores());registerDoParallel(cl)
>
> classdata <- readRDS('./TFM/contenido/datos/classdata.rds')
> clusterdata <- readRDS('./TFM/contenido/datos/clustresults.rds')
>
> #Names of the predictors (wards) are changed to avoid issues with caret package.
> variables<-sapply(names(classdata), function(x) paste("Var",which(colnames(classdata)==x),sep=""
    ↪  ))
> varnames <- names(classdata); names(classdata) <-variables
>
> #Values of the cluster must be factors (not number) to avoid issues with caret package.
> classdata$cluster <- sapply(clusterdata$cluster, function(x) paste("Cluster",x,sep=""))
> classdata$cluster <- as.factor(classdata$cluster)
> sink('./TFM/contenido/codigo/output/classvars.txt')
> str(classdata); sink()
```

```
'data.frame':        126 obs. of  42 variables:
 $ Var1   : chr  "01.1 Palacio" "01.2 Embajadores" "01.3 Cortes" "01.4 Justicia" ...
 $ Var2   : num  -0.177 -0.202 -0.227 -0.219 -0.207 ...
 $ Var3   : num  -0.624 -1.157 0.142 0.188 -0.79 ...
 $ Var4   : num  0.0165 1.0132 -0.5497 -0.221 0.5851 ...
 $ Var5   : num  -2.64 -2.7 -2.94 -2.58 -2.52 ...
 $ Var6   : num  -0.0195 0.4183 -0.4377 -0.0358 0.2991 ...
 $ Var7   : num  -0.2535 -0.7483 0.0558 0.262 -0.5009 ...
 $ Var8   : num  1.121 0.595 1.14 1.998 1.132 ...
 $ Var9   : num  -2 -1.89 -2.26 -2.12 -2.19 ...
 $ Var10  : num  -1.88 -2.25 -2.29 -2.03 -2.21 ...
```

Source Code Extract 13: Preparing classification datasets for R package Caret.

```
> #This creates a training set with 70% training data and 30% validation data.
> indices <- createDataPartition(classdata$cluster, p=0.7, list= FALSE, times = 1)
> training <- as.data.frame(classdata[indices,-1])
> validation <- as.data.frame(classdata[-indices,-1])
>
> #Resulting datasets
> sink('./TFM/contenido/codigo/output/trainingdata.txt')
> cbind(data.frame(validation=table(validation  %>% select(cluster))),data.frame(training=table
    ↪  (training %>% select(cluster))))
> sink()
```

Source Code Extract 14: Training and validation data-sets.

```
> #Oversample training
> library(DMwR)
> training <- SMOTE(cluster ~ .,training,k=5,perc.over=300)
> sink('./TFM/contenido/codigo/output/balancedData.txt')
> cbind(data.frame(validation=table(validation  %>% select(cluster))),data.frame(training=table
    ↪  (training %>% select(cluster))))
> sink()
```

Source Code Extract 15: Training and validation data-sets after oversampling.

```
> #Models training
> model.knn<- train(x = training[,-C], y = training$cluster,
>                   method="knn",
>                   trControl=trainControl,
>                   tuneLength=10,
>                   metric = "ROC")
> model.rf<- train(x = training[,-C], y = training$cluster,
>                  method="rf",
>                  metric="ROC",
>                  trControl=trainControl,
>                  tuneLength = 10,
>                  na.action=na.omit)
> model.cart<- train(x = training[,-C], y = training$cluster,
>                    method="rpart",
>                    trControl=trainControl,
>                    metric = "ROC",
>                    na.action=na.omit)
> #Stop cluster
> stopCluster(cl)
> env <- foreach:::.foreachGlobals
> rm(list=ls(name=env), pos=env)
> #Save results and data
> saveRDS(training, './TFM/contenido/datos/training.rds')
> saveRDS(validation,'./TFM/contenido/datos/validation.rds')
> saveRDS(model.knn,'./TFM/contenido/datos/knn.rds')
> saveRDS(model.rf,'./TFM/contenido/datos/rf.rds')
> saveRDS(model.cart,'./TFM/contenido/datos/cart.rds')
```

Source Code Extract 16: Training process for the three different classification algorithms.

```
> #Validation
> predictionKNN <- predict(model.knn,validation)
> predictionRF <- predict(model.rf,validation)
> predictionCART <- predict(model.cart, validation)
>
> results.KNN <- confusionMatrix(predictionKNN, validation$cluster, mode = "prec_recall")
> results.RF <- confusionMatrix(predictionRF, validation$cluster, mode = "prec_recall")
> results.CART <- confusionMatrix(predictionCART, validation$cluster, mode = "prec_recall")
> sink('./TFM/contenido/codigo/output/results.KNN.txt'); results.KNN; sink()
> sink('./TFM/contenido/codigo/output/results.RF.txt'); results.RF; sink()
> sink('./TFM/contenido/codigo/output/results.CART.txt'); results.CART; sink()
```

Source Code Extract 17: Predictions and confussion matrix for the three classification algorithms.

```
> #Gain charts for binary RF model
> library(gtools)
> library(funModeling)
> for(i in 1:3){
>         data <- classdata %>% filter(cluster!=paste('Cluster',i,sep=''))
>         data$cluster <- sapply(data$cluster, function(x) as.character(x))
>         data$cluster <- as.factor(data$cluster)
>         indices <- createDataPartition( data$cluster, p=0.7, list= FALSE, times = 1)
>         training <- as.data.frame(data[indices,-1])
>         validation <- as.data.frame(data[-indices,-1])
>         training <- SMOTE(cluster ~ .,training,k=5,perc.under=300)
>         trainControl <- trainControl(method="repeatedcv", number=K,repeats=rep,savePredictions=
            ↪  "final",classProbs=TRUE,verbose=FALSE,allowParallel=TRUE,index=createMultiFolds
            ↪  (training[,C],k = K,times = rep))
>         binarymodel.rf<- train(x = training[,-C], y = training$cluster,method="rf",metric="ROC"
            ↪  ,trControl=trainControl,tuneLength = 10,na.action=na.omit)
>         predictionRF <- predict(model.rf,validation)
>         gaindata <- validation
>         gaindata$score <-predictionRF %>% as.numeric()
>         gaindata$cluster <- gaindata$cluster %>% as.numeric()
>         pdf(paste('./TFM/contenido/imagenes/class',i,'GainCurve.pdf',sep=''),width = 10)
>         gain_lift(data=gaindata,score='score',target='cluster')
>         dev.off()
> }
```

Source Code Extract 18: Transformation of the data-set with three classes into a binary data-set. Also training of three RF binary claffifiers.

```
> #Inspect features importance
> #Random Forest model
> vars <- data.frame(var=rownames(varImp(model.rf)[[1]]),varImp(model.rf)[[1]])
> varindx <- sapply((vars %>% arrange(desc(Overall)))[1:10,]$var,function(x) gsub("\\D*(\\d*)",
    ↪  "\\1",x) %>% as.numeric())
> sink('./TFM/contenido/codigo/output/varimp-rf.txt')
> varnames[varindx];sink()
>
> #CART model
> vars <- data.frame(var=rownames(varImp(model.cart)[[1]]),varImp(model.cart)[[1]])
> varindx <- sapply((vars %>% arrange(desc(Overall)))[1:10,]$var,function(x) gsub("\\D*(\\d*)",
    ↪  "\\1",x) %>% as.numeric())
> sink('./TFM/contenido/codigo/output/varimp-cart.txt')
> varnames[varindx]; sink()
>
> #KNN model
> vars <- data.frame(var=rownames(varImp(model.knn)[[1]]),varImp(model.cart)[[1]])
> varindx <- sapply((vars %>% arrange(desc(Overall)))[1:10,]$var,function(x) gsub("\\D*(\\d*)",
    ↪  "\\1",x) %>% as.numeric())
> sink('./TFM/contenido/codigo/output/varimp-knn.txt')
> varnames[varindx];sink()
```

Source Code Extract 19: Top ten predictors for the different models.

```
> #Pot training, validation and oversampled data-sets.
> library(cowplot);library(readr);library(DMwR);library(caret);library(tidyr) ;library(dplyr)
> classdata <- read_csv("./TFM/contenido/datos/classdata.csv") %>% select(-Barrio) %>%
    ↪  as.data.frame()
> names(classdata) <- paste("Var",c(1:length(names(classdata))),sep='')
> classdata <- scale(classdata,center = TRUE, scale = TRUE) %>% as.data.frame()
> clusterdata <- readRDS('./TFM/contenido/datos/clustresults.rds')
> classdata$cluster <- sapply(clusterdata$cluster, function(x) paste("Cluster",x,sep="")) %>%
    ↪  as.factor()
> indices <- createDataPartition(classdata$cluster, p=0.7, list= FALSE, times = 1)
> training <- classdata[indices,]
> validation <-classdata[-indices,]
> trainingOversampled <- SMOTE(cluster ~ .,training,k=5,perc.over=300)
> data <- list('training'=training %>% select(Var20,Var9,cluster), 'validation'=validation %>%
    ↪  select(Var20,Var9,cluster), 'trainingOversampled'=trainingOversampled %>% select(Var20
    ↪  ,Var9,cluster))
> for(i in 1:length(data)){
>         ps <- list()
>         ps[[1]] <- ggplot(data[[i]], aes(x=Var20,Var9,fill=cluster,alpha=0.5,colour=cluster)) +
            ↪  geom_point(show.legend=FALSE,size=3, shape=21) + theme_minimal() +
            ↪  scale_fill_manual(values=c("#5b9bd5", "#ed7c31","#70ad47")) + scale_colour_manual
            ↪  (values=c("#2e75b6", "#c55911","#548235")) + xlab('Feature 1') + ylab('Feature 2')
>         ps[[2]] <- ggplot(data[[i]], aes(Var20, fill=cluster,colour=cluster)) + geom_density
            ↪  (alpha=.1,position="stack",show.legend=FALSE) + theme_minimal() + ylab('') +
            ↪  scale_fill_manual(values=c("#5b9bd5", "#ed7c31","#70ad47")) + scale_colour_manual
            ↪  (values=c("#2e75b6", "#c55911","#548235")) + xlab('Feature 1') +
            ↪  scale_y_continuous(limits=c(0.0,1.0),labels=function(x) sprintf("%.0f%%",x*100))
>         ps[[3]] <- ggplot(data[[i]], aes( Var9, fill=cluster,colour=cluster)) + geom_density
            ↪  (alpha=.1,position="stack",show.legend=FALSE) + theme_minimal() + ylab('') +
            ↪  scale_fill_manual(values=c("#5b9bd5", "#ed7c31","#70ad47")) + scale_colour_manual
            ↪  (values=c("#2e75b6", "#c55911","#548235")) +  xlab('Feature 2') +
            ↪  scale_y_continuous(limits=c(0.0,1.0),labels=function(x) sprintf("%.0f%%",x*100))
>         #gridplot <- plot_grid(plotlist=ps,ncol=3,rel_widths = c(1,1,1))
>         for(j in 1:length(ps))
>           ggsave(paste(names(data)[[i]],"_",j,".pdf",sep=''),plot=ps[[j]],device='pdf',path=
              ↪  './TFM/contenido/imagenes',dpi=400)
> }
```

Source Code Extract 20: Plotting training, validation and oversampled data-sets.