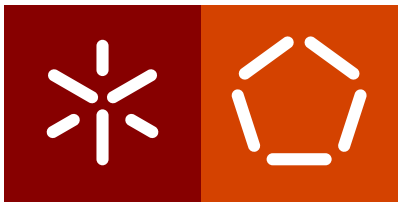
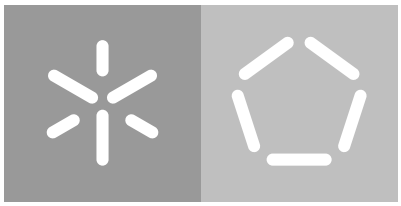


Universidade do Minho
Escola de Engenharia
Departamento de Informática

José Pedro Ribeiro Nunes Simão

Historical Data Management in Big Databases

October 2017



Universidade do Minho
Escola de Engenharia
Departamento de Informática

José Pedro Ribeiro Nunes Simão

Historical Data Management in Big Databases

Master dissertation
Master Degree in Informatics Engineering

Supervisor:
Orlando Belo, Department of Informatics,
School of Engineering, University of Minho
Company Supervisor:
Jorge Soares, Primavera B.S.S.

October 2017

ACKNOWLEDGEMENTS

Ao meu Professor e orientador, Orlando Belo,
por tudo o que me ensinou, pela dedicação, o auxílio
e toda a motivação e inspiração
que me passou ao longo destes anos.

Aos meus pais e irmão que ao longo da minha história me têm vindo
a acarinhar, ensinar, inspirar, a fazerem-me e deixarem-me sonhar.
No fim de contas, por me ajudarem a ser quem sou,
ao deixarem a vossa marca em cada um dos meus capítulos.
A par de tudo o que me oferecem, em particular neste capítulo da minha vida,
admiro e agradeço a maravilhosa inspiração que são para mim
e todo apoio que incansavelmente me dão.

A ti Sofia, que me tens acompanhado
e fascinado ao longo destes anos.
Por tudo de bom que temos podido compartilhar,
pois todas essas experiências têm-me feito crescer e ajudado a alcançar os meus objectivos.
Pelas nossas boas conversas e sonhos que partilhamos
que alimentam a nossa vontade de querer mais e voar alto.
Pelo teu espírito crítico, a paciência, pelos teus conselhos, e, claro,
por todas as horas que passaste com as minhas gafes!

A todos os meus amigos que me têm vindo a acompanhar ao longo da vida,
tem sido uma jornada fantástica tendo-vos a meu lado.
Sem individualismos, pois sabeis quem sois,
todos foram importantes na definição da minha pessoa
e contribuirão para os meus sucessos e bons momentos na vida para sempre.

À empresa Primavera B.S.S. pela oportunidade,
por todo o auxílio e a experiência transmitida.
Em especial, um forte agradecimento a toda a equipa de logística que me acolheu
e a ti Jorge Soares pela ajuda, o acompanhamento,
os ensinamentos e dedicação durante todo o estágio.

ABSTRACT

We are now living in a digital world where almost anything, or something is saved somewhere with very few considerations for determining if that was in fact relevant to be saved or not. Hence, it is predictable that most information systems are facing an information management problem. To overcome this issue, it is vital the creation of new and more specific data management techniques that will enforce the established governance policies and manage the information systems in order to maintain their ideal performance and quality. Currently, a solution that is able to cope with this problem efficiently is "pure digital gold", especially for the biggest players that have to handle an astonishing amount of data, which needs to be properly managed. Nevertheless, this is a problem of general interest for any database administration, because even if shrinking the dimension of the information is not a major concern in some cases, the data assessment efficiency and its quality assurance are certainly two subjects of great interest for any system administrator. This work tackles the data management problem with a proposal for a solution that uses machine learning techniques and other methods, trying to understand in an intelligent manner the data in a database, according to its relevance for their users. Thus, identifying what is really important to who uses the system and being able to distinguish it from the rest of the data, is a great way for creating new and efficient measures for managing data in an information system. Through this, it is possible to improve the quality of what is kept in the database as well as increase, or at least try to ensure, system performance. Basically, what its users expect from it throughout its lifetime.

Keywords: Databases; Data Quality; Data Management; Data Mining; Machine Learning.

RESUMO

Estamos a viver num mundo digital onde praticamente tudo que alguém ou algo faça é capturado e guardado em algum sítio, com muito pouca consideração que determine se esse evento é ou não relevante para ser guardado. Como tal, é previsível que grande parte dos sistemas de informação tenha, ou venha a ter, um problema de gestão de informação no futuro. Isto obriga a que sejam criados novos tipos de técnicas de gestão de dados mais eficientes e específicos para cada caso, que sejam capazes de governar os sistemas de forma a assegurar o desempenho e qualidade desejados. Atualmente, uma solução capaz de lidar com este problema eficientemente nos tempos que correm é "ouro digital", especialmente para os grandes intervenientes neste domínio que têm de lidar com uma quantidade exorbitante de dados e que, por sua vez, precisam de ser devidamente geridos. Apesar disso, este é um problema de interesse global para qualquer equipa de administração de bases de dados, porque mesmo que a diminuição da dimensão da base de dados não seja uma preocupação fulcral para certos casos, o eficiente acesso e a qualidade dos dados existentes numa base de dados serão sempre dois assuntos de grande preocupação para qualquer administrador de sistemas. Neste trabalho, é investigado o problema da gestão de dados através de uma proposta de solução, na qual através de técnicas de machine learning, tenta com inteligência perceber, aprender e classificar os dados em qualquer base de dados, de acordo com a sua relevância para os utilizadores. Identificar o que realmente é importante para quem usa o sistema e ser capaz de distinguir esta informação da restante, é uma excelente forma para se criarem novas e eficientes medidas de gestão de dados em qualquer sistema de informação. Assim, certamente, irá aumentar a qualidade de tudo o que é mantido no mesmo, bem como aumentar, ou pelo menos tentar assegurar, que o desempenho do sistema é o esperado pelos utilizadores.

Palavras-Chave: Bases de dados; Qualidade de Dados; Gestão de dados; Mineração de Dados; Aprendizagem Máquina.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF LISTINGS

LIST OF ABBREVIATIONS

DBMS - DataBase Management System

ERP - Enterprise Resource Planning

IDC - International Data Corporation

OLAP - Online Analytical Processing

SQL - Structured Query Language

XML - eXtensible Markup Language

1

INTRODUCTION

1.1 CONTEXTUALIZATION

In times like this, it is widely common for any business or other activity that needs continuous monitoring and storage of its valuable data to have an information system for solving this problem. However, what is really happening is a dramatic change in our world. Especially, in the way we work and have access to "things". Everything is digital now, and that is a tendency that no one can fight. From major companies and organizations to smaller ones around the world, databases are now the leading technology of choice for supporting most of organizational information assets. They are designed to store, organize and retrieve digital information, being a fundamental part of the information systems that most would not be able to function without them (Connolly and Begg, 2005). Therefore, it is possible to foresee a major problem, which is the lack of performance and management due to excessive amounts of data.

Nowadays, information management and systems performance are major concerns for large-scale organizations, since they potentially deal with massive amounts of data that have to be saved, monitored and its quality ensured. Other examples, which are following the global tendency, such as smaller organizations or even single entities, are now accommodating their businesses or working structures into digital platforms. This means that they might also be vulnerable to current data management problems. In these cases, data flood may not be a reality or even a concern, since there is not the same information traffic like in other examples. Nevertheless, digital space is expensive on a performance and money perspective and is certainly going to be more expensive in the future. Hence, if there is a way for increasing the performance of a system and to diminish the data's dimension into what really matters to users, it's obviously very interesting for any database consumer, increasing the value of having a management tool capable of that in any database scenario.

Focusing the attention on larger scale business organizations, it is clear that they rely on databases for storing every aspect of their businesses, making them the most "punished

victims" of the data management problem. These organizations have to operate and keep track business activity, in all the aspects related to sales, inventories, shipments, human resources, finances, just to name a few. To deal with this great variety of business processes, what has become a reality for supporting them is the existence of software capable of handling the information of a company.

An Enterprise Resource Planning (ERP) system is a powerful piece of software with the ability to support all kind of business activities, covering in general all of their digital necessities. An ERP system serves as a cross-functional enterprise backbone that integrates and automates many internal business processes and information systems (Marakas and O'Brien, 2010). Generally, every ERP system has a huge database system beneath it, depending on the organization's business scale, but typically they are pretty large and complex systems that are constantly transacting data and, consequently, they are continuously growing in its data dimension. Usually, the information that is kept in an ERP system is very sensible. However, usually there are some parts of the global data that are not so important. Possibly, due to the fact that those parts are not so frequently used, are becoming historical throughout the lifetime of the system, or simply because that data does not represent any worthy value for system users. Assuming that a software like this may support a business for years, it is predictable that in most cases there is information that will lose value along its lifetime. Consequently, an enormous quantity of disposable information will be generated on the long run, being possible to identify and separate it from the rest, as already concluded. Nonetheless, this may also be a problem in any database regardless of its size (Marr, 2016b).

PRIMAVERA BSS. S.A. is one of the biggest players in Portugal when it comes to develop business solutions software. One of its products is the PRIMAVERA'S Enterprise Relations Planning system and it is one of the bestselling software of its kind in Portugal. Nevertheless, even being a quality product, it is unavoidable that it will suffer from the lack of data management like any other system of its kind, if the global tendency keeps being verified. Besides, PRIMAVERA BSS. S.A. also offers simpler products that are designed for smaller businesses and enterprises. Despite being part of a different dimension category than their "bigger cousins", this kind of products will also encounter problems related to the exact same things. The issues will not be on a scale as big as the previous case, but they are manageable and it would be worth investing some effort on such kind of scenarios. Therefore, the size of a system does not quite matter. What is worth to pursue is the identification of what is important in each application scenario and suite the best strategy for enforcing the quality and performance of a database system.

This work was developed in cooperation with PRIMAVERA BSS. S.A. and the case study was designed over the main company's ERP system. The main objective of this project was to create and develop a solution that contributes on the mitigation of the general data management problem in a generic way. By being a generic solution, means that it is intended to formulate a methodical process that has to be applicable to any database regardless of its size or domain. The solution has its main focus in the identification of what is the most valuable information for a system user by differentiating data by relevance levels. By doing this, it is possible to create measures for improving the system. For example, implementing a parallel structure where the valuable data will be stored and can be easily accessed, like a cache memory, which deeply improves the performance upon data assessment. Identifying this kind of data is also very important, because aside from the performance factor, it is also possible to understand what is worth to keep stored in the database and then define new policies and actions for improving the quality of the system's data. Ultimately, based on the predicted classifications generated by the solution, we believe that what is relevant to the user is the most valuable data. Given that, implicitly, it is being defined what quality data is for that system in particular.

1.2 MOTIVATION

As it is known, every system sooner or later will suffer from data management problems and of course from the performance problems that are unavoidably linked to each other. The creation of solutions and new methods that will help to fight it are undoubtedly a global interest. With a clear conscience of this struggle in today's information systems, it is clear that a solution that mitigates this issue is extremely valuable, not only for big data scenarios but for any information system, as already referred. Often, we try to compare the problems from big data against the smaller scale database systems, tending to separate the two concepts and trying to divide the solutions and their applicability for each case separately, which is wrong. Big databases, such as the ones present in ERP systems, are not big data scenarios. However they have also huge difficulties to handle all the data efficiently and excel their performance. If we compare that to a big data case, it is clear that it will not be much different. Instead, the difference is going to be related to the complexity of the data dimension and its entropy (confusing mess of structured and unstructured information). Basically, the problems are pretty much the same but the data structure and dimension are different, which reveals the huge value of a solution capable of handling both, generically. So, in the future, if nothing is created until then, the scale of this problem will be even larger. It will not be solved with new hardware technology or by vanguard database systems. The problem is still going to be there no matter what. This is an unbounded and potentially highly scalable issue that needs an intelligent and generic

solution for improving the quality of any information system of today and in the future.

Studying data management without the relation with big data scenarios is a little bit utopian and that is probably going to be the favoured scenario for a solution of this kind. Analysing this scene, the global tendency is to generate more and more data and, obviously, store it somewhere. Some might not even think about this fact, but at the rate that this phenomenon is being spread, it will not be stopped or slowed, and surprisingly, it is increasing exponentially every day. This is the well-known data explosion phenomenon (Zhu et al., 2009). Another interesting fact to consider about this is that 90% of the data generated until today was created in the last two years. So, besides being a recent problem based on the explosion rate until now, it is daunting to think about the future dimensions of the problem in some cases if nothing is done until then. To have an even better insight over this, just picture that two years ago there were no such problems, or at least not as serious as today, and now they are real and are becoming even more frightening. Some say that in 2020 the amount of data is going to be ten times greater than today. So it is quite easy to guess that managing all that data is going to be a serious tough challenge (Marr, 2016a). In addition, besides all the facts stated above, the International Data Corporation (IDC) predicts that over the next three to five years, companies will have to commit to digital transformation on a massive scale, including fundamental cultural and operational transformations. They also predict that changing how companies interact with them will be mandatory for organizations that hope to grow revenue or increase market share. Thus, the big data scale of things is soon going to be a reality for any enterprise. Making any form of business hungry for data no matter its dimension, which is going to, consequently, transform most of simpler systems of today into big data alike scenarios - a kind of pseudo big data, but still considerably large (Marr, 2016b).

Most of the inexperienced new companies in the scene, believe that information is a valuable commodity, and many claim that: the more they have of it, the more they can learn from it, and make changes that will drive business success. This is a reflex of the rush for avoiding being left behind. By having such mentalities, many companies risk becoming data rich but insight poor and chaotically organized. They accumulate vast stores of data they have no idea what to do with and without any hope of learning anything useful from it. Another interesting and scary aspect of all this is that besides the enormous quantity of unnecessary data, a lot of it has a lifespan. At some point in time, it becomes no longer relevant, as well as inaccurate or out-dated and that affects general data quality. Despite that, it is often held onto the database anyway in the mistaken belief that someday it might become useful. This is a critical mistake on many perspectives, but mainly because it is not efficient to keep all that data if only the analytical purposes are attended, and also because

a huge amount of disposable data is going to be kept in the database, making this is an expensive action when it comes to money and performance of an information system (Marr, 2016b).

Observing and analysing the current state of things and the tendency for the future, two ideas come up to mind instantly: there is a need for reducing the dimension of useless data and improving the quality of the database by knowing what is really important to a user. Well, these two ideas are related to each other, because if there is a way for identifying what is important to a user, it is possible to properly act upon those conclusions in terms of governance. For example, less relevant data can be removed or transformed, while the most important pieces of information may be kept in another repository that will favour it assessment. Having this kind of dynamic governance mechanisms, it is possible to dramatically increase the performance of a system and assure its quality throughout the time. However, what is even more interesting is the fact that the system administrators may create and adequate governance measures to fit their criteria and necessities, which will have a deep impact in systems' performance and reliability.

To tackle the problem, we have today a large variety of methods and approaches. The solutions currently available are not generic and most of the times are only applicable to the specific case for which they were designed, such as data governance measures or specific mechanisms to cope with certain database entries of a system. Other solutions, that might be generic in most cases, are often related to querying optimization, database redesign, denormalization or other well-known mechanisms like the intermediate aggregate values for huge historical records tables. The problem with those strategies is the fact that they are probably already being applied in most of the systems. Thus, it is necessary to design and create innovative and different data management solutions. A machine learning based technique is a possibility, which would be able to determine, through the power of predictions, what is valuable inside a database system in a generic way. Having the knowledge of what end users wants, is a great way for filtering the most precious information from all the rest. That may sound easy, but the true challenge is to create a method that will learn from the information systems and adapt itself to each one of them. A system able to do it is where the greatest value stands in this domain. A solution that is adaptable to any database is important, because it has to be a generic method that could be applied to solve the general problem of every system, and not just one in particular. This is a fundamental interest behind this project. If it would just be applicable to one particular case, the objectives and performance goals within that domain would likely be achieved, but would not be a meaningful contribution for the global problem solution.

1.3 OBJECTIVES

Once established the motivation for this project, it is now comprehensible that the right path to success is to formulate a generic solution capable of identifying relevant data in a database according to some kind of user criteria. By classifying the existent data, it is possible to know what is important and what is not, which is extremely valuable for an organization. For instance, for each and any case, it is possible to discover and create new governance policies for acting upon the classified data based on the fact of what is important. For example, if the data dimension is the main concern of a particular entity, then it is probably better to remove the useless data or store it somewhere else. Nevertheless, if the dimension is not so important, another good method to increase the performance of a system, could be caching the most important data into a secondary memory structure with faster access. This could be very interesting if there is a huge volume of data in a database that cannot be removed, but only parts of it are used frequently. With the acquired knowledge, it is possible to act accordingly to each system's struggle. However, the main concept to retain here, is that it is only possible to have this range of new possibilities if there is a way to retrieve those conclusions from any database table dataset.

With just some basic examples of practical cases, the endless possibilities of data governance rules are revealed simply by having access to the knowledge of what the end user considers meaningful. Knowledge about the users is going to be the most crucial factor in the solution, because no matter what the purpose is, the data knowledge acquisition is the fuel that ignites new possibilities to improve the quality of a system. Having this in mind, it is fundamental that the solution is able to capture every piece of information from a system that can be transmuted into knowledge and in its turn be used to determine the relevance of the data. The simplest approach for knowing what the user wants is by analysing the database transaction log. On those records, it is possible to retrieve information about the users' usage of the system and adequate the relevance classification over the data that is based on it. Besides the logs, there are plenty of other options for inducing knowledge discovery on certain datasets, which is the case of the application usage behaviour detection that may provide a deeper insight of where and what the user accesses on the application.

A user behaviour analysis could be very helpful to determine what are his main tasks and duties on each session and then, later, it is possible to translate and transform those conclusions into actions that are going to be applied to the data itself. Heuristic methods, like determining the most queried or frequent attribute of a table, or the most common value of an attribute, also provide information about what is important in a system. Knowing those insights is crucial for a classification process that requires constant refinement

due to the system's mutation over time. Therefore, summing everything together, it is of a major interest that the solution is able to perform this kind of knowledge scavenging, in order to improve its results during the database system's lifespan.

The classification process is what is going to determine if a certain tuple of information (which typically is a database table instance) is relevant or not. As stated before, the relevance factor of a particular thing will change over time. Hence, it is necessary to have a system that is capable of adapting itself towards the relevance changes. One of the best ways to do it is by inducing the perception of the relevance changes along time, being able to learn with those changes and adapt the classification model to be more precise and versatile. Knowing what matters is extremely important, being able to understand it beforehand and to incrementally learn from each system change, is what makes possible the adaptation and continuous refinement of the classification results. This is what a generic solution for any scenario must do to achieve the desired performance and the viability of the project for a longer period.

The automatic learning and self-improvement of the method are by far one of the crucial factors for the success of the final solution. Despite that, it is also very interesting to allow for users' definition of certain rules and preferences over data. Having a way to feed precise knowledge into the solution, such as user defined preferences and rules, is a great way to mitigate errors on the classification process and allow user customizations. At the end, it really does sound like: "this might be the solution for the problem...", but what should be kept in mind is that these kind of systems are not completely accurate or safe. The classification result is a mere prediction based on what is known about the data, thus, a fully trustworthy solution cannot be expected. Nonetheless, if the accuracy is high for each prediction, that error will not be so significant and may produce excellent results for a given system. To improve this method, the solution has to learn from its previous results, which can be achieved by using the knowledge that was extracted from the system throughout its usage and from what the users prefer as well. From that, it is possible to refine the predictions and improve the quality of the overall results.

Data management problems are a reality for many years to come. A solution like the one we propose here could be a helpful way to fight it. It may not be the perfect solution since it has an error percentage associated with the results. Nonetheless, it is definitely a robust and different solution that could bring serious improvements in several systems. May not be that linear for some other cases, but what is vital to retain is the contribution to this global digital problem, and that is a major objective for this project as well.

1.4 DOCUMENT STRUCTURE

Beyond this initial chapter, this document is composed by other five chapters, namely:

- Chapter 2 – *Related Work* - This chapter presents and describes fundamental database concepts and management techniques. A brief introduction on the available machine learning techniques will be presented along with the applicability of those methods in the solution. To finalize this chapter, an analysis on the current state of the data management solutions will be made so that the project's feasibility is clearer.

- Chapter 3 – *A Methodological Approach* - Here, it is presented the problem we faced and its challenges. The problem is going to be introduced at first followed by the proposed system architecture. Basically, in this chapter we will oppose the problem and the solution in order to explain the reasons that led the development to blossom into the final system architecture.

- Chapter 4 - *A Solution for Data Governance* - Having a solution capable of answering the problem is not enough and because of it, this chapter is related to the development phase of the project. Here it is discussed all the decisions we made in the project and their explanations. The implementation details are also very important and they will have a dedicated section as well as the outcomes, which are referred to the main results of solution. To strengthen the results and the proof of concept, it is going to be studied and presented the scientific evidence for them in a last section dedicated for this matter.

- Chapter 5 – *The Case Study* - In this chapter, it is presented the case study and the tests that were conducted to evaluate the created solution. Firstly, the case scenarios are introduced along with the tests' specifications, which are crucial sections to review in order to understand the approach followed. The section that follows is dedicated for the explanation of the validation method applied to evaluate the results, which are later analysed based on the requirements defined previously.

- Chapter 6 – *Conclusions and Future Work* - Finally, we present and discuss the conclusions of this work. Since the project has a quite large dimension, it was inevitable that some work had to be postponed. Therefore, in this final chapter the work that was not completed will be overviewed along with the explanation of the procedure that has to be done in the future for ensuring the success of the proposed process.

2

RELATED WORK

2.1 DATA MANAGEMENT

The technological outbreak of the decade has set new boundaries and requirements into our systems, especially in the information systems domain. In the last years, the digital world is facing a new problem - the data explosion -, which is demanding new and sophisticated technology and methods. It is absolutely imperative to create new sustainable ways for ensuring that this problem will not jeopardise the information systems data management area in the future. Without a proper solution, it will be impossible to manage the ridiculous amount of data that almost every system is going to have. Consequently, if there is no way for handling the excess of information, it will eventually become useless and stacked somewhere, resulting in a drastic loss of information and performance. In order to support events like this, database systems are suffering important developments. Probably that is quite arguable, but by mentally drawing the big picture, it is clear that a database is now the supporting structure of any information system and that it has changed the way most organizations operate. The development of this field has always been quite fascinating for the researching community in general. Contributions are impressive, but new blank spots of knowledge in this domain are always being discovered due to constant changes and advances that also occur in the area. With this, it is mandatory to continue researching for new techniques and solutions to improve and adapt these systems to its reality.(Connolly and Begg, 2005)

2.1.1 *Data management in database systems*

Database systems are facing a problem that demands new technology and methods to help in its mitigation. Today's businesses are generating a massive volume of data, which has grown beyond the limits of efficient management and analysis by the traditional data processing tools. Relational databases were designed and built in an era completely different from this one and, consequently, they are not very adequate to this new age of information (Grolinger et al., 2013). A database has to fit a certain purpose and to guarantee the

requirements that have to be taken into consideration. For instance, the way users access the information, the scaling strategy of the system, the data's structure of what it is going to be inserted in the database, or even the quality of what is kept in the database. To overcome this factor, database administrators had to redefine the way they manage their systems along the time. This is why this subject is so crucial. New measures and techniques were continuously researched and created in order to improve system's data management policies and ensure that their demands are achieved. The best way to accomplish those requirements and avoid data inconsistencies, lacks of performance or scalability issues, is by defining a management plan. Using it, it will be possible to have a complete control of what is kept in the database. A strict plan and measures to rule data is certainly going to be a major improvement in the way administrators control their systems and create solutions to oppose the eventual problems.

Data management is not a single task or something that may be singularly identified. It can be seen as a collection of procedures and definitions that are related to the governance of the information in any database system. Basically, it is related with every aspect of the systems' data, involving duties and procedures that have to be defined and executed for ensuring the quality of the systems. More specifically, some of the tasks that are included in data management are data reduction and inconsistency detection, the data strategy definition, which is a new and very interesting topic, data monitoring, business understanding and many others. In the end, with the well succeeded implementation of these duties, they will have to be able to ensure the system's scalability, performance, availability, security and data consistency, due to today's heavily demanding reality. To have a deeper insight on the most relevant techniques, in the next section we introduce and review some of the most relevant techniques that are currently being used in data management domain (Sakr et al., 2011)

2.1.2 *Data Management Techniques*

. The amount of data that has to be dealt today is enormous and the "one size fits all" method is no longer recommendable, because different systems have different needs. In response to that, entities have to define their own governance policies and data management tasks, in order to achieve their goals with their system. Thus, it is vital to understand each system's needs, and adequate the best management measures for keeping data consistent, available, correct, and, most importantly, relevant to the user. As stated above, data management is not a single task, it is the whole set of duties that are needed for ensuring the quality of a specific system. The actual tasks and the way they are implemented may differ from case to case, but in the end, the global objective is essentially the same. These

techniques started to be defined since the creation of the first database system. For instance, the creation of DBMS in the early 70'ies was probably one of the first pioneers and most influential inventions in the field. It made possible atomic management of data inside a database (operation management - insert, delete, select and update). It also brought the notion of a centralized database in which every user could access with concurrency controls and security measures (Connolly and Begg, 2005). Nevertheless new techniques and theories were formulated since then, such as the data strategy definition, which became known not long ago and that is a great example of a vanguard concept to cope with the current tendency.

Some of the most important tasks that compose any data management plan are related with data integration and consistency, dimension reduction, redundancy removal, data cleaning, documentation development about governance policies and data details, data profiling, and so forth. These were just some of the most common procedures that have to be done in any database system management with quality. They are pretty well known and are not going to be detailed in the sequence of this document, but it is very important to have them in mind at least. Other methods like data strategy, data monitoring or user behaviour analysis are different techniques that are also being used nowadays, but they are a bit different, especially, in the way they act. These techniques are not managed on an operational level. Instead, they are done on a higher level, which is related to crucial definitions of governance measures enforcing data organization and, consequently, the quality of a given system. These higher level techniques are incredibly interesting and are also becoming very trending due to the global tendency. The fact that leads to this fashion, it is because the best way for managing information systems is by defining strict rules and policies to control and treat data operations. Specifically, the data strategy started to appear in the radars, in part because researchers felt the need for defining a method to be a standard procedure in which is defined the data policies for a system. Essentially, those policies are rules that define what data quality is for a system and how to control every aspect of the company's data. The idea is to define strategies that are designed for defining how data should be handled in every situation and even for defining what are the quality standards and procedures to keep data at its best integrity inside the company's systems. The definition of how data should be managed a priori is a great method for predicting future problems and defining actions to avoid system malfunctions (Gertz et al., 2004).

By defining the data requirements for a system in up front, implicitly, it is being defined what quality data is for that particular system. It is essential to know and define this kind of procedures for an information system in order to keep improving it and ensure its desired performance over time. Therefore, to keep track of all the aspects related to sys-

tem's data, it is mandatory to have some kind of monitoring system to steward data. The data stewarding or monitoring concept, is also a primary management duty that should be considered for any database system due to the huge information traffic that these system experience. The definition of what is relevant for a user, is one of the most important definitions that could be imposed and the best way for discovering it could be by capturing and interpreting the user behaviour in the system. Having access to that kind of information, new governance policies and measures could be formulated. Being able to identify the most relevant information is crucial for assuring the quality of a system and consequently its performance. That is very important since the major goal is to manage data in order to serve users in the best possible way. In fact, it might be the most versatile information one could have about a database system in order to improve its quality in all the aspects considered pertinent. Anticipating the future, a consequence of the data driven world and the analytical requirements that are implied by it, the reality of management is soon going to change. The system's performance and quality in its data will be the main priorities in the following years and as difficult as it may sound, with far more impact than what they had until now.

2.1.3 *Data explosion and the exponential growth of the database systems*

With the development of the human being and technology, the size of data is increasing in an uncontrollable manner. According to the International Data Corporation (IDC), the data growing factor is ridiculously huge, meaning that from 2005 until 2020 the digital data is going to evolve from 130 Exabytes to 40,000 Exabyte (Gantz and Reinsel, 2012). Given these numbers, the size of data is going to be double in every two years from now on until 2020. This represents a massive growth of data that certainly will not get any smaller throughout time. In addition, if the tendency is confirmed, the data's quality will not be any better either. To understand this phenomenon, it is a good idea to drive into a retrospective on the very first years when we (humans) started to gather data. Those were the times when the human being was literally a pre-historic gatherer. The human began to carve stones and walls to record valuable information, which could be about the way they hunted animals or the looks of the herbs they gathered. In a certain sense, the human being of today did not change much. In fact, today's people also want to store valuable information and keep certain things that are important for them safe, which is not any different from what was done 40,000 years ago. People figured out that certain important things had to be recorded, and, since then, it became a common habit throughout the ages. As a result, from the time the first stone was carved until the modern times, there was more than one data explosion phenomena. The invention of the papermaking and printing was the turning point in the way people had access to information. In those times, everything that was important to be

recorded was thoroughly documented and represented with characters or figures and then printed into books or documents alike. This represented the first data explosion that our world went through and it was the first process of informatization so to speak, where information could be saved, replicated and accessed in a much easier way. With the invention of the personal computers, but especially with the World Wide Web and with the innovative devices (smartphones, digital storage devices, etc.) that came after it, a new era of data explosion started. With this happening and the Internet of things, the computer systems are explosively bursting with data. Just imagine that it is now possible to process and transform everything that was created in the previous data explosion into digital systems, apart from the massive amounts of new data that are now simultaneously being generated (Zhu et al., 2009).

An interesting fact, that is one of the actors behind the data explosion problem, is the current tendency for data driven businesses. With the emergence of data science, organizations are now able to trail their decisions more wisely with data analytics and with the power of prediction. This new era of knowledge gathering and decision support mechanisms are generating an unreasonable amount of data to feed the world's needs. This has been out for quite some time, but it was restricted to certain kinds of businesses or activities, because data gathering and its consequent analysis is an expensive task that not everyone could afford. Currently, this is not so bounded to the same entities and smaller businesses are now eager to collect and grasp control of their data and make use of it. This data generation is completely out scaling the data dimensions by each year and what is more concerning is that only an astonishing small percentage of data is being analysed. However, we recognize that there is an incapacity of treating and digesting all existing data, which demands new techniques to aid the data analysis and management tasks of every information system (Gantz and Reinsel, 2012). It is very concerning to know that people are generating data out of everything without even making sense of it, without the certainty that what is being kept is actually valuable or even true. The consistency of data systems is being jeopardized because there is so much information about the same things that it is hard to know what matters and what does not. Therefore, it is clear that if nothing is done in a near future the human incapacity of dealing with this problem is going to be even greater than what it is now.

Envisaging the future, for a third generation of the data explosion, there would have to be a new system for replacing computers. This kind of system would be fed with any kind information (structured or not) and it would save it and access it whenever it is prompted. It could also learn and capture information from what it sees and listens with appropriate devices. Having an idea of what this could enable and imply, it is quite difficult to antic-

ipate the future dimension of data and the problem itself. However, comparing it to the current state of things a slight idea comes right up to mind (Zhu et al., 2009). Anyhow, the problem still prevails and new alternatives to the computers and information systems are in a relatively distant future. Thanks to that, the adversity must be faced now. As a wise man once said, "the suggestion of what a revolutionary system could be is just a mere assumption, because predicting the future is something really hard or almost impossible to be accurate, even for the advanced future beings with all their powerful future technology" (Portela, 2013).

As far as this matter goes, it is crucial to keep improving the data management techniques in order to cope with this problem and guarantee user requirements. Database system performance is at stake and so is the rest of the information necessities demanded by users. Thus, a good way to start improving the quality of any system is by establishing a robust data management plan before even thinking about the operational techniques that could be used.

2.2 Database Optimization Techniques

Today, it is unavoidable not to relate database performance with the dimension of the data. Information grew both in size and diversity, demanding for new database optimization techniques. Since the creation of the first DBMS, researches struggled with the performance of the "heavy" queries in the system. From an initial starting point, this was an issue that ignited the will of developing new solutions (Jarke and Koch, 1984). When the subject is the database performance, it is imperative to mention query optimization as well. Queries are the orders that users give to a data base management system, which are interpreted and executed internally in order to perform user requests. The interpretation and execution of the query itself is what takes time, and influences the performance of a system. Knowing this, it is clear that the optimizations have to mainly act upon these requests and their execution.

2.2.1 *Relational database optimization techniques*

The development of Database Management Systems (DBMS) was probably the most influential improvement in database systems optimization. Still, these systems were unable to cope with the current state of the dimension of the data and business' demands. Systems needed other kind of performance upgrades. Nevertheless, it is important emphasizing the relevance of DBMS and their evolution in this subject, which played the most crucial role. Given the current state of the information systems and DBMS, database designers

struggled with the relational database tuning to be able to create sustainable ways for maintaining their systems. Techniques like indexes or auxiliary storage structures were critical optimizations in their era. This subject and its evolution is a critical concern in every business that depends upon a database for its operations and, therefore, it is imperative to develop new strategies. Techniques used to tune a database are, for example, the insertion of indexes to increase speed on primary keys, on sorted or group by fields, on fields that are frequently used for selection criteria, etc. Advanced techniques for data storage have emerged as well. These include specifying where to store data on hard disks for optimal retrieval. Disk striping and distributed techniques are also available for parallel processing and I/O access (LaBrie and Ye, 2002). Additionally, caching is also an awesome optimization for some cases. Caching frequently used data and creating in-memory structures to ease the assessment is a frequent and effective strategy worth to be used. Some other more exquisite techniques, like query planners, query cost estimation or other strategies to improve queries performance, were also key factors in the optimization subject (Ioannidis, 1996). Other good ideas came with data warehousing systems, which were concepts out of the box for the standards in any database system. Despite that, some of them really offered significant improvements in the performance, such as table denormalization or aggregation techniques, which are crucial for ensuring database performance. For example, a very interesting technique, which could benefit the solution presented, is the integrating usage analysis on cube view selection technique. This allows for the selection of the most important views, and with an adequate strategy it can be used to solve many problems (Rocha and Belo, 2015). In brief, the optimization revolution is still a subject of huge relevance for investigation - new techniques are constantly being developed. The crucial idea to retain is that every information system has its own requirements and flaws and one must be able to identify them for develop better and adequate optimizations.

2.3 Machine Learning

The machine learning revolution represented an enormous breakthrough in the way humans analyse data, especially in the data mining field. Machine learning techniques allow for us to learn from data and other sources of information, in order to build models that are used to predict, score and classify data properties in a much easier way than before. This capability is extremely valuable and is wide spread around the most various applications in computer science, and also in fields like medicine, banking or marketing. For example, classification methods analyse and learn from data in order to predict certain characteristics of a set. Clustering methods are another very interesting method that relates data through its similarities, and by doing so, it is possible to create groups of similar data and from the consequent study, the resultant insights are really valuable in an analytical perspective

(Ajinkya and Basil, 2017). Given the superior ability to classify data, these methods could be applied in a solution for data management. The purpose of this research work is to formulate a solution that through the use of machine learning methods will classify data according to its users' relevance. With this capability it is possible to discover problems in a system as well as what evaluate its data quality. This kind of insights is truly valuable for improving data management and having a proactive behaviour towards this matter. Therefore, next we will present a small overview of machine learning methods that are currently available and their applicability for solving problems in the data management area.

2.3.1 *Machine Learning Methods Overview*

Machine learning systems revolutionized the way data is handled and processed. This was extremely impactful in the data mining and analysis fields. The capability of extracting predictive and categorical insights or being able to score a precise value just from the data that is fed into a system is extremely valuable and spares most of the hard work that had to be done before building something alike. Regarding these techniques, the most fascinating thing is probably to understand how they are able to achieve such results, which is done through a refined process of learning from data itself in several distinct ways. This way, machine learning methods may be sub-divided in two general types: supervised and unsupervised methods. These methods differ in the way they learn to be able to build a model and in a practical way of putting things. Supervised methods need prior knowledge about data in order to build a model, while unsupervised methods are able to learn from the data itself. More specifically, the unsupervised learning methods try to create subsets of data that are independently and identically distributed. This way it is possible to relate data by its similarities and draw some conclusions from that. An interesting feature about these methods is that they learn for relating data from themselves without any supervision. The supervised learning methods have the goal to label certain attributes in a set by looking up correlations in a training set that has prior label mappings. Basically, the training set is going to teach the algorithm on how to predict those labels. This training set is built with precise knowledge (or at least is meant to be) that was induced beforehand and, therefore, it is a supervised method (Chapelle et al., 2006). Despite the different learning methods, the purpose of each is the same, which is to automatically learn from data and be able to create conclusions. The data-driven approach similarity that both follow is due to the fact that both required as much data as possible, either to train or to learn from it to efficiently build the correspondent models. Given that, they are clear the reasons that propelled the data explosion in this era. As machine learning evolved, so did the need for data for refining these methods and it is now one of the main reasons why there is so much data in today's systems.

Supervised methods offer interesting options when there is enough prior knowledge about data for training a model and it is being expected precise results based on that information. For instance, some supervised classification methods are fascinating to predict values and categorize data based on some known examples. These are useful for forecasting and other applications alike. Regression methods are also very interesting in the way they are able to predict numerical values of certain classes and therefore can be used in estimation processes. Another interesting feature is the way these methods may be evaluated, which can be performed by validating results against known examples and estimate their precision and accuracy. To better understand the way these methods work, some authors say that are certain steps that have to be followed when implementing a supervised learning system. For instance, firstly, it is required to gather data and knowledge about it for arranging a training and a validation set. After that, choose the algorithm that is going to build the model and then iteratively build a model by creating and validating the results with the validation set. To evaluate the model there are plenty of possible strategies available to choose, such as the cross validation method (Kohavi, 1995). This evaluation technique consists in iteratively evaluate the building model by comparing its predictions against a known set with correctly labelled instances to measure its accuracy, precision and other meaningful metrics. When the model is finalized, then it is possible to classify a new set of data. In figure 1, it is illustrated the workflow of the supervised learning methodology (Vapnik, 2000).

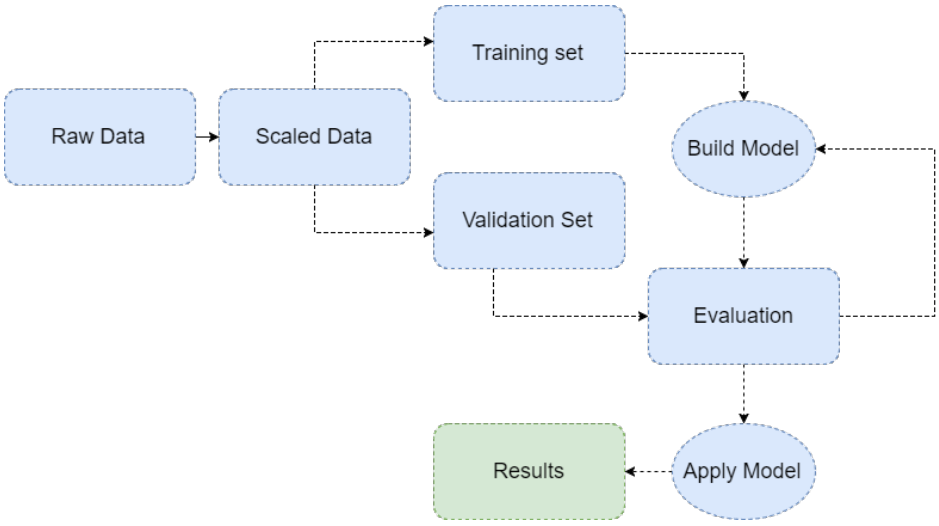


Figure 1.: Standard workflow of a supervised learning methodology.

Unsupervised methods, such as clustering techniques, are able to aggregate instances by sets, based on their data similarities. By creating groups of similar information, it is pos-

sible to establish patterns and other extremely valuable insights about data. Besides those techniques, there are methods that can be used to support the machine learning systems for identifying outliers in data by observing and selecting instances that show abnormal values. Dimensionality reduction methods are also used in pre-processing stages and are quite useful to eliminate useless attributes in the datasets and increase algorithm performance. Some other techniques are used to build neural nets, which are other interesting implementations of unsupervised methods, such as the self-organizing maps algorithm (Kohonen, 2001). Generally, to evaluate the performance of unsupervised methods it is a quite harder task, when we compare them to the supervised ones. For instance, clustering algorithms are difficult to evaluate. Most of the internal metrics used only evaluate if the clusters are compact and well separated. However, it is possible to define external metrics that perform statistical procedures in order to test the structure of data. Hence, they can also be used as evaluation methods as well (Halkidi et al., 2001). A subjective analysis of the results, with a strong knowledge of the data in question, may also be quite informative. In figure 2, we can observe an example of a standard workflow of an unsupervised learning procedure.

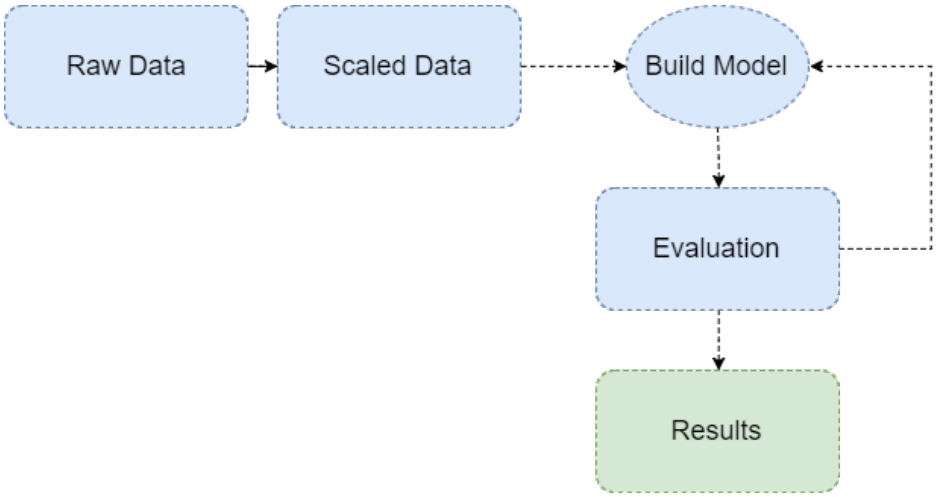


Figure 2.: Standard workflow of an unsupervised learning methodology.

These techniques offer efficient and easier ways for building better models and support various fields where they may be applied. The decision on what is the best type of learning is not a proper question. A matter dependant of the purpose that one has for the procedure as well as on the approach that is possible to take. Thus, the choice is only dependant on those facts. Figure 3 presents a simple representation of the difference between the two types of learning and their correspondent results (Valpola, 2000).

In short, machine learning methods serve many purposes with excellence and are frequently being applied in new scenarios. These techniques allow for achieving something that was never possible before with ease and efficacy. In the data management field, it

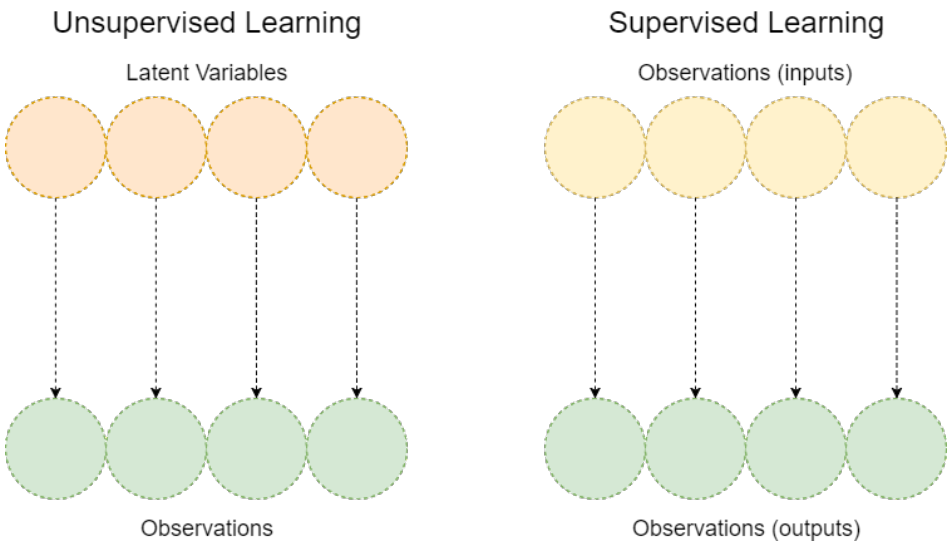


Figure 3.: Illustration of the difference between the two types of learning and their results.

is possible to imagine a large number of possible applications for these methods, such as dimensionality reduction solutions that are so well needed in times like these. Therefore, the following section will be dedicated for exploring the suitability of the machine learning in a data management solution.

2.3.2 *Suitability for a Data Management Solution*

Since it is a popular topic, data management problems are continuously having new solutions to overcome its issues. The dimension of data is greater than ever and the urge for new solutions for managing it is in the mind of every researcher in this field. In addition, the dimensionality is not the only problem related with the insufficient management. Organizations are now being struck with different issues related with their data quality. This is a very concerning topic. The lack of data quality is often related with the amount of data, because one cannot make sense of chaos. Nevertheless, some other times it has nothing to do with the amount of data, but instead it is related with the lack of meaning that data has for users.

Data quality is related with a set of conditions that each information system imposes, and it varies from each and every case. It is hard to define data quality and there are plenty of acceptable definitions, but data is considered to have high quality if it fits its purpose (Redman, 2008). For organizations, it is vital having their data with quality since they may have a lot to lose from the lack of it. Wrong decisions are definitely quite rough in terms of costs for most organizations. Some studies state that more than 60% of medium-sized

companies in the United States (with annual sales of more than 20 million dollars) have problems in their data. This is an alarming number that demands organizations to enforce their data management policies for improving their data quality. However, to improve data quality it is necessary to understand what data quality is for a system in particular. The best way for achieving this is by analysing what is the most important data for who uses the system – the consumers (Wang and Strong, 1996). By identifying the most important pieces of information in a system for its users, it is being implicitly defined what data quality is for that system and its administrators may suite the best governance measures for ensuring system performance.

An interesting idea is to study the application of machine learning methods, which may learn from data and other information, to identify the most precious data of a system. The application of these methods in a solution for managing data is quite conceivable. For instance, by using classification techniques it is possible to categorise data according to a certain criteria, which in the data management scene could be the relevance grade that a user has for that piece of data. Following a reasoning alike, with the application of clustering techniques, it could be possible to identify various groups in data and study their characteristics trying to relate them with their quality and relevance for the system. Essentially, with the use of machine learning methods it is possible to classify data in a database according to the system users' relevance. Having a classification grade for each table instance, it is possible to analyse those results trying to discover the most and least important information in that system. With this information, new data management measures could be created for improving system performance and data quality. For instance, if it is possible to identify the most important instances from a frequently used table, then it could be a good idea to create a cache memory for that information for favouring access to its users. When a system is struggling with the dimension of the data and the removal of information is a viable option, then removing or decentralizing the least important data from the mainframe, could be a feasible solution for the problem. All these new options for managing data could be created from the analysis of the relevance factor. Therefore, the important idea to retain is that is valuable to know what is relevant for a user to suite the best data management measures for improving the system. Through machine learning techniques, along with the adequate auxiliary systems, it may be possible to conceive a solution to unveil efficiently the relevance factor for system's data. As for the auxiliary systems, these would have to have tools to gather knowledge about the users that could be used to improve the machine learning methods and for other utility tasks like data preparation procedures. This approach could also be useful in further data mining applications, because by creating a new measure that evaluates data, it is possible to conduct more studies for discovering new insights that could really be useful in a business perspective, not just for

data management purposes. Some of the conclusions could also be obtained from the general data mining techniques over the existent data, but having this new method to classify information may offer different perspectives that could be worth to explore. Saying this, it is important to refer again that the main purpose of this research work was to investigate a way for improving data management in big databases. The approach taken for solving this problem was to explore the machine learning methods to classify data according to its users' relevance. This way, we believe that new governance measures could be created as well as new means for reinforcing the existent policies from knowing what is important in a system. Besides, it may also offer valuable insights that could be applied for improving business aspects.

3

A METHODOLOGICAL APPROACH

3.1 OVERVIEW

The problem and its challenges. Being aware of the problems that could affect the systems and where the value stands, the basic idea for a data governance solution is to identify and classify data according to its relevance to system users. The approach taken was to develop a solution that through machine learning techniques will be able to classify data extracted from any database according to its relevance to users. Having every table instance classified, users or system administrators will have the power to decide accurately about how to proceed and manage the information based on those results and conclusions. Basically, one can adapt data governance measures by having access to the relevance factor for each piece of information. The process acts as a low-level procedure, which analysis data on the operational level that unlocks new high-level measures for improving system performance and quality. In Figure 4, we can see a conceptual view of the approach we will follow in the methodological approach.

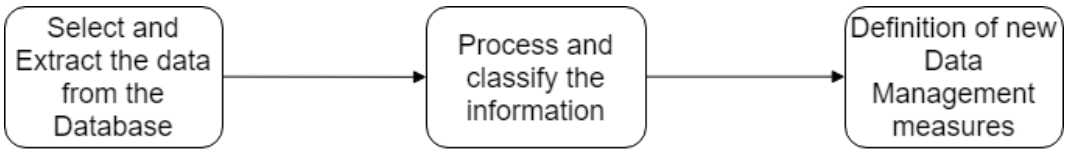


Figure 4.: Main tasks of the methodological approach.

There are plenty of techniques available for determining the relevance of a single piece of data. One of the most well-known machine learning techniques to achieve these results are the classification methods, as already mentioned. Although these methods are quite powerful when the objective is to classify information according to a certain criteria. They can be very demanding in terms of performance as well as prior knowledge that one has to possess for inducing the machine learning process and ensure its success. Besides those techniques, clustering methods are also a very robust choice, when it comes to distinguish information in a dataset that is achieved by comparing data and creating groups of similar

instances called clusters. Although the clustering methods are not to an option to exclude, they may not be the most suited for solving the problem. Since the objective is to classify data and imagining that there is sufficient knowledge for feeding supervised classification methods, then it is probably a better option to pursue first. Having these in mind, supervised classifiers are the approach we will follow and though the methods to achieve the relevance grade may be different, the main idea remains, which is to use the information from that analysis for improving system data management.

An interesting perspective on the machine learning techniques is their capability to continuously learn throughout time. With the implementation of these techniques, there is going to be created plenty of new data to support them. This new data allows for bringing up to date knowledge about the system usage and other valuable insights that may be used for rebuilding previous models. Data changes along the time as well as the users' interests. Thus, it is at the best interest to reinforce the models with new data. Nevertheless, old data that was used for training the initial models is not to be disregard, since it may also provide valuable knowledge. Therefore, an efficient way for selecting the best data to train the models must be implemented, such as semi-supervised learning techniques to classify new data or clustering analysis to create groups of similar data and reduce the dimension of what is going to prepare the models (Chapelle et al., 2006; Halkidi et al., 2001). All of these techniques may be explored when implementing a strategy to retrain the models following this process.

The atomic pieces of data that are going to be analysed by the algorithm are database tables instances with its attributes and values. Data extraction is the easiest part, but building the models requires a training and testing set for each single table. Starting with the simplest, the testing set corresponds to each table's rows that are transformed into a dataset, which later is going to be classified by the algorithm. On the other hand, to create a quality set for training a prediction model is not so easy. This set has to be composed of previously labelled instances and these classifications have to be accurate and up to date, which is hard to achieve without the proper approach and tools. To overcome the lack of knowledge problem, the exploration of the database logs information is by far one of the best options for extracting knowledge about the most frequent operations, and therefore the most important tables, their attributes and values in a system. By having access to the most important operations and other insights that the operational logs may offer, it is possible to extract valuable knowledge that can be used to influence the scoring and labelling phases of the initial training set instances. For example, each operation has at least a table instance related with it along with its values. If the operation is a delete, it is not completely wrong to conclude that the given instance and its values may not be relevant, or at least will be less

relevant than the instances associated with insert or update operations. Despite of being naive with this kind of reasoning, it is conceivable the construction of an initial training dataset classified only by analysing the type of operation. However, this procedure is not sufficient, because there are certain details that can mislead the relevance of the classification. For example, there are old records that are not often used, but might be crucial to the database system, and they should not be marked as non-relevant. In some cases, a delete operation may have important attribute values associated that will be marked as non-relevant, which may influence the initial label scoring in a wrong way. Therefore, it is required to explore new ways for refining the available knowledge.

Analysing those simple examples, it is concluded that will be needed auxiliary scoring systems for supporting the initial training set classification process and refining the initial classifications. Nevertheless, the operational log is one of the most important artefacts considered in this process, and should be seen as the main source of knowledge about a system. In a real scenario, it is advisable to create a custom logging system that will serve the knowledge induction phase in the best possible way. For example, important features to be logged are different in every system and, besides that, each system user knows what are the best features to be logged that will provide the most valuable knowledge about what is relevant in a database. For example, adding specific metadata to each log operation, such as a timestamp or the system user that executed the operation, are very valuable features for determining specific insights that may be related with the relevance factor and used in its initial calculation. As for the auxiliary scoring systems, they represent a very important role in the whole process, because, as stated before, it would not be sustainable to label the training sets through the operational logs insights alone. These systems are based in the simplest and most effective form of gathering knowledge about what is important in a database, which is by asking users about what they prefer. By simply asking the users about their data preferences, it is possible to create systems that will process and refine the relevance factor in order to meet some system users' criteria. In this stage is where the data strategies might come in handy and could be enforced on an operational level. As for the implementation of these systems, it is necessary to have a format for representing the defined preferences and rules over data and a simple way to process them, in order to adjust the relevance factor of a given training set. Other conceivable auxiliary scoring techniques, like user behaviour detection systems, are another plausible and very interesting way for capturing valuable insights about what are the most important pieces of information inside a database, which can be used to refine more the dataset's scoring procedure. Cooperation among scoring systems is the key, because, after all, what is going to ensure that the classifications are well inferred, according to the users' relevance. Figure 5 illustrates the process

of building a predictive model by having access to database operational logs having scoring systems implemented.

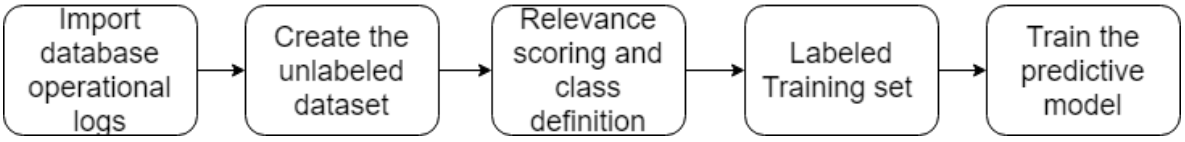


Figure 5.: Process of Training a Predictive Model.

Lastly, the ultimate and fundamental decision is related with the choice of the machine learning algorithm. There are certain algorithms that perform better with specific types of data and for specific purposes than others. Thus, it was very important to test and evaluate different algorithms and data preparation techniques by conducting several practical tests. Then, trough the analysis of the results, one should be able to achieve a solid choice that fits the solution goals. Similarly to the type of machine learning techniques, the classification algorithms are immense. Each one of them has its own characteristics that are suited for certain specific datasets. Due to this reason, it is imperative to investigate the best possible option for processing data and building a robust predictive model. The data preparation techniques are also within a very important domain to be considered and explored. As we know, each algorithm has its own data requirements. Thus, each dataset must be effectively prepared for achieving an optimal performance by the machine learning methods we chose. This is a task that has to be well designed, since the quality of the predictions is going to be critically affected by this pre-processing phase.

Imagining a perfect scenario, where the classifications of the relevance of the algorithm are ideal. The next step is the definition of new measures to support the management of system's data, based on the insights obtained from the analysis of the classified data. The variety of possibilities is the fascinating part, because one can adapt its measures to fit their own purposes based on the insights achieved with this procedure. For example, if the data assessment is an issue due to the fact that data is disposable, then the data marked as irrelevant should be deleted or sent to a different structure other than the original platform. On the opposite, if it is relevant to the user, its access should be favoured instead, as it is in a caching structure. Through the process proposed here, it is defended that it is possible to improve the performance and the data quality of a system by identifying what is important to a user. Consequently, after the analysis of the results, it is possible to infer new management measures in a system that otherwise would not be possible. Besides the management factor, the discovered insights may also be useful in a business perspective, which could be extremely valuable for a company. For example, by having the sales' transactions classified, it is possible to infer a good number of conclusions about the relevance of the products

and services that could be sold by the company. Next it is going to be presented some eligible scenarios, and their requirements, for implementing a solution such as the one here proposed. After that, it will be presented an analysis on the expected results and also on the possible validations of the produced results to evaluate the usefulness and correctness of the process.

3.2 POSSIBLE SCENARIOS AND REQUIREMENTS

To develop a solution based on the process proposed, it is imperative to analyse the domain of the problem. Therefore, it is important to conduct a study on the data management problems of an information system in order to evaluate the necessity of implementing a solution, such as the one we proposed here. There are certain application scenarios where this proposal excels while in others might not be necessary. The importance of this study is to avoid the implementation of a complex and demanding solution that has to meet certain requirements to be well succeeded. An eligible scenario could be any that has data management problems associated with it. An information system that is experiencing a data explosion related problem might need to explore a way to diminish its database dimension and it would benefit from having classified the relevance of the data. With a data relevance analysis, it is possible to evaluate what are the best pieces of information that might not be important to keep in the database. This way, in case of having to remove some data from the database, there is more knowledge available to support the decision of which data should be removed.

Since performance and availability are always at stake in most systems, some may have to find solutions for improving their performance on accessing the most valuable information. Most of the information of these systems may not be removed and they might not be able to afford losing data in order to improve the performance. One alternative could be the implementation of auxiliary data structures for favouring the access to the most frequently requested data. This way, some of the data could be moved to those structures and avoid the system over occupation, favouring the performance in the main platform as well as of the most valuable information located in secondary systems. Once again, the analysis of the classified datasets could provide plenty of worthy insights about the most adequate instances of each table that should be held on those auxiliary structures. Those examples were meant to illustrate a picture of the most common problems and a possible solution with the methodological approach. However, these are just a tiny portion of the possible scenarios, which could benefit from having their data classified. Some businesses could make use of the relevance classifications for discovering new knowledge about their data. If it is possible to rate each instance of a database according to their user usage,

then it is also possible to discover what are the most important tables, attributes and correspondent values. The use of data mining techniques, could help to discover unknown conclusions that otherwise would not be possible. For example, those conclusions may be related with the most important products that a company might sell, the most important clients, or even about data patterns that could be used to support business decisions. Most importantly is that some of those conclusions could not be achieved without having the relevance classified, because it provides a different perspective of the information. Following the same reasoning, if a database administrator analyses the classified data, he may be able to discover data patterns that could serve various other purposes than the ones related with business activities. Since the data management is the major concern, the conclusions acquired from the analysis could be used to define new data management measures and enforce system governance policy. Having this kind of approach to solve data management problems is certainly going to improve the management itself and it is much more effective to have a pro-active attitude towards these tasks rather than reactionary behaviour. If problems can be predicted beforehand, then it will be possible to adequate the best solutions to solve, having plenty of time to react. In the future, having a solution that is able to have this predictive functionality it will be possible to anticipate problems and their solutions for a longer term.

Analysing some of the possible scenarios, it is possible to anticipate certain requirements that are common in every system for implementing a solution based in the proposed approach. However, there are some other requirements that are not so evident and should be discussed before the implementation. Firstly, and as stated before, it is very important to evaluate the usefulness of having such solution, and, of course, the associated implementation costs. If there are problems with data management, then the costs must be evaluated, in order to conclude if there is a worthy necessity of implementing this solution. Basically, each system administrator responsible has to evaluate if they are facing any of the typical scenarios and study the usefulness of having their data classified and what they could do with it. More technically, there are some requirements and remarks to have in mind related with the size of the database and the type of the engine. As for the size, although bigger databases theoretically represent a better candidate to have any data management problems or at least one of the scenarios presented above, the smaller databases may also represent a strong candidate to make use of the solution to solve their problems. Again, if a system has any data management problems and could benefit from the use of its data classified to solve them or to extract valuable insights about its data quality, then it is worth to pursue an implementation based on this process. As for the database engine, it is important to know that there must be an intermediate and generic representation of the database structures, so they can be processed by the solution and its algorithms. For example, database engines such

as NOSQL (commonly referred as not only SQL databases) or relational database engines are some examples where it is possible to convert a table or document into an interpretable file format that can be processed by the application. Essentially, if it is possible to create a dataset for each database entity with its attributes and values, then it is possible to adequate this process to that database engine. A different structure of dataset can also be explored for solving the problem where the procedure cannot be entirely applied. For example, if there is not any way to represent the attributes and values of the database tables' instances, the dataset's format must be adapted to represent an interpretable form of knowledge that may be used to classify the data's relevance.

Besides all these requirements, another is mandatory to consider due to its criticalness for the success of the application of the process. Since the most adequate algorithms that are going to be used to classify information are supervised methods, there must be a way to gather knowledge about system usage, and a translation into insights that may help the classification process of the information, according to the system's user relevance. One way this could be achieved is through the database operational logs analysis, with the help of user behaviour detection systems, or simply by asking users what is preferable, as suggested previously. The operation registry is full of worthy information about the relevance of the data that may be explored. The behaviour detection may also provide more knowledge that could be induced on the training sets. As for the auxiliary systems, these represent further alternatives that must be explored in order to capture more information and tune the one that is already available. The coordination of all these, will hopefully ensure that the implementation of the process is consistent and will be able to deliver the expected results.

Finally, the last requirement that must be considered is the fact that it must be conducted a supervision through some follow-up tasks for ensuring the correctness of the solution and results. Additionally, we need someone capable of evaluating if the results delivered are correct and useful for solving problems. Besides, since the solution is based in machine learning techniques, it is advisable to have prior knowledge on how to evaluate these procedures. Furthermore, some of the machine learning tasks need to be customized with user defined data pre-processing procedures and one has to be able to identify these sort of problems as well as being able to create solutions to solve them. Hence, if an eligible scenario meets most of these requirements, then it is worth to conceive a solution based on the proposed methodical procedure.

3.3 EXPECTED RESULTS AND VALIDATION

Knowing what the eligible scenarios are, it is clear that results will have to be related with the data management subject applied to each specific case. The goal of processing a given solution is to deliver a classification grade to each instance of a table in a database. This way, we believed that it is possible to formulate insights about data quality and based on this adequating the best possible governance measures for solving the problem. Therefore, the most notorious expected result is the relevance classification of each database instance that is processed. The consequent results will have to be related with the specific information system's problems and the way their administrators want to proceed to mitigate them. For instance, if the objective is to diminish the dimension of the database, then an expectable result would be the identification of what is irrelevant and could be deleted.

Besides that major objective, from the analysis of what is irrelevant, the administrators may expect to find relations between irrelevant data for predicting future problems and identifying what data quality is in their systems. In a different perspective, when the goal is to improve the system performance in terms of data assessment, a possible result could be the identification of the best instances to be moved into an auxiliary structure that offers a better reading performance. Following the previous example, along the path for achieving the main goal, it is possible to discover new and useful insights from the analysis of the classification results that will ignite the creation of new data management measures and provide new conclusions about the data quality of that system. A predictable result is to improve the data management in an information system. But, throughout the process, it is possible to discover new blank spots of knowledge about the data that may be extremely valuable for the administrators. Hence, the desired results will have to be dependent from system needs and the way their administrators decide to act. Finally, the main objectives for having a solution based on the approach we proposed, must be defined beforehand, in order to guide and define the purpose of this implementation for each particular case.

Since the actual data management improvement is going to be dependent of the measures applied after data relevance analysis, it will be important to certify that every inference is correct. In order to do so, the machine learning methods have specific evaluation methods that provide enough information for deciding if the created model is able to perform the task or not. In addition, a person responsible for the system should also validate what was classified, so that it is possible to evaluate the correctness factor of the results. This person would be someone that knows the database and should be in touch with implementation of the solution. Therefore, this person knows what the data preferences and issues are and may be able to determine if the results reflect them or not.

Another crucial factor is the usefulness of the results, because, sometimes, results may not be conclusive and they could not be able to provide enough information for supporting data management decisions and providing quality insights about system's data. Most of the times, this problem is related with the lack of prior knowledge about the usage of the system and with the defined data preferences. Thus, it should be possible to overcome this situation with an extra knowledge-gathering phase. However, it is extremely important to be able to detect the usefulness of the results for determining if they are really worth to look at or not. An interesting technique to support the usefulness evaluation of the results is performing a data mining analysis over the classified data, in order to see if the classifications do make sense and could be used to support decisions related with the data management of that system. These analysis are not easy to conduct, but are indispensable for ensuring the correct functioning of the solution. They also have to be conducted in a long-term analysis, since most of the results will only be available after the application of the first data management measures. After analysing the impact of the new created measures, it is possible to have the complete validation of the procedure. Some results from the applied measures may be instant, but others will only show its effects in a longer run. This way and as discussed before, the requirement of having some follow-up tasks for ensuring the correct behaviour of the solution is extremely important. It is vital monitoring results and being able to infer if they are correct and useful to solve the data management problem.

4

A SOLUTION FOR DATA GOVERNANCE

4.1 AN OVERVIEW

In order to explore the validity of the process presented before, it was decided to implement a prototype for proving the ideas that are being defended in this work. The idea was also to make the prototype a usable tool that is able to execute a procedure for processing an entire database by applying the conceptualized process to achieve its goals. Having a methodical process conceptualized, the following step was to develop an architecture for the prototype that is intended to be develop. Since the idea is to classify data according to user criteria, in order to acquire more knowledge for unlocking new governance measures, the efforts must be channelled into a machine learning engine. This means that the whole solution is going to revolve around that core, and every system created will have to support and ensure its success. In other words, all the procedures created will be related with how data has to be extracted from a database, pre-processed and scaled before applying a machine learning process. Besides that, since the type of learning of the algorithms is supervised, knowledge has to be gathered, translated and induced into a machine learning module in order to classify each piece of information. Therefore, it is safe to say that the prototype is composed by a main core and set of secondary systems for ensuring its efficiency and performance.

The program and most of its auxiliary systems were built in Java, often because there are plenty of tools and libraries conceived for this language, which are helpful during the development period. Another benefit is the multi-platform compatibility that it offers (Arnold et al., 2006). Moreover, to ease the processing and compatibility issues, the machine learning engine we selected for use was the Weka platform, which is also built in Java (Witten and Frank, 2005). This platform offers a complete data mining module composed by machine learning algorithms for different mining processes, a package of functions and features for helping data pre-processing and model evaluation, as well as a wide domain of other features that are fundamental in any machine learning project (Hall et al., 2009). Finally, for increasing the accuracy and explore further available algorithms, it was also implemented

a deep learning algorithm offered by the H2O platform (Candel et al., 2017). The idea was to use these tools since they offered a wide scope of very well implemented resources to be explored and used, and this way have enough means for building a prototype according to the process we designed.

In order to better understand the procedure and its application to the presented solution, it will be better to divide the architecture procedures in four major phases. The first one is the table selection or scope definition phase, which happens when the tables from the database system are selected before the data extraction, and therefore it is being defined the scope of analysis. Right after that, it is imperative to build the training and testing datasets. This is the second phase. Having access to the datasets, the next phase is to submit them into the machine learning module so that they can be classified. Finally, we reach the fourth and final stage, where we adequate the management measures based on the analysis over the labelled data. One example of this could be to cache the most relevant data or decentralize it from the main computational platform. In Figure 6 we can see the process schema, in BPMN (Business Process Model and Notation) of the prototype and its auxiliary systems.

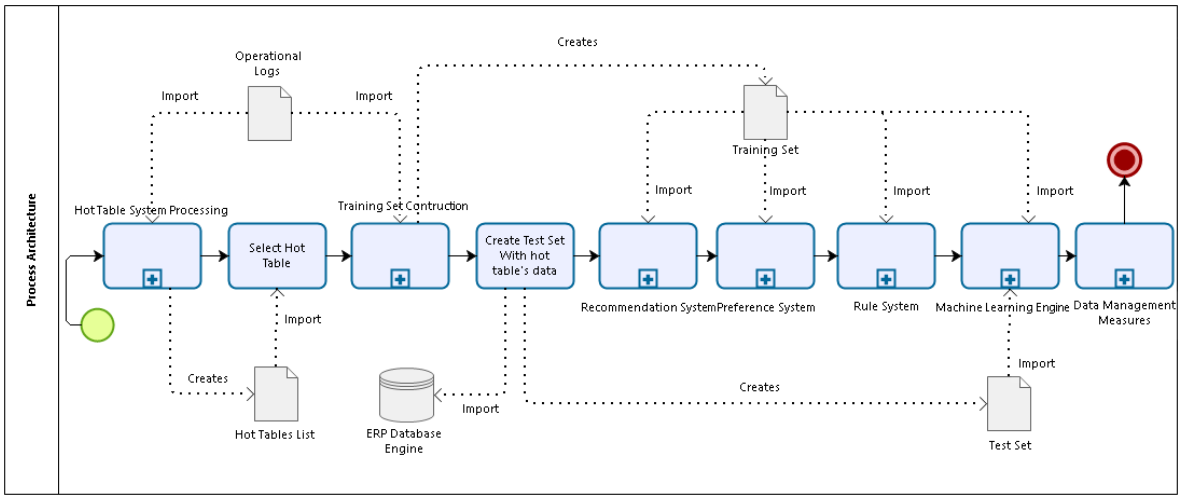


Figure 6.: The processing schema of the prototype.

The idea was to transpose the conceived procedure into the presented architecture. In the process presented in Figure 6, it is possible to observe all the four distinct phases. Although, some of the tasks presented are related with the auxiliary systems that had to be implemented, such as the “Hot Table System” or the “Preferences System”. The “Hot Table System” represents the selection phase. This system consists in an implementation that selects the most adequate tables to be taken into further analysis through the observation of the database operational logs. The training set construction is the next task to be performed. It uses the operational logs as well for performing the first scoring phase. Basically,

it consists in defining an initial relevance grade for each logged instance from the table in analysis, according to its operation for building the first training dataset. As suggested before, the database operational logs were used to create the training set and to add general knowledge about the system. But, essentially, they constitute one of the main resources of the prototype. Building the testing set is a simple task, as it is only required to extract the database raw data and create an intermediate readable file by the machine learning module. The procedures after the dataset construction tasks compose the training set second scoring phase, which is executed by the auxiliary systems. Each one of these systems adjusts and refines the training set initial relevance scoring differently, but are all based in knowledge about user preferences over data. The preference and rule systems act as a kind of filters based on user criteria about certain data aspects. However, rules perform a stronger influence than preferences.

The recommendation system was created to generate automatically suited user preferences over data by mining the initial association rules of the training set. This way it spares the hard work of having to define numerous preferences and rules for each table. In any case, it may be used for further refinement even if there are enough preferences or rules. This way, this system may be optional as well as mobile in the architecture, which means that it may be executed before or after the second scoring phase for improving its results and besides that application, it may also be used to replace the rules and preferences systems. In a real case scenario of the application of this solution, this kind of tweaks is crucial to explore in order to find the best processing setup for each case. After having the training set scored, the machine learning engine processing composes the third phase. One of the best approaches in machine learning is to train, evaluate and adequate data to fit an algorithm, doing this process iteratively until the best model is achieved. Therefore, this module is composed by two supervised classifiers, which are the Naïve Bayes and a Deep Learner classifiers, along with auxiliary tools for supporting the machine learning process. The decision of those specific alternatives was not simple to take, since there is fair amount of options and some perform better than others do, depending on each particular dataset and purpose. Another development issue was related to the data preparation of each different sets, which demands specific data preparation procedures. The idea was to come up with a generic way for preparing data, which would fit the necessities of the set and still deliver acceptable results. Hence, from the analysis performed, the Naïve Bayes algorithm seemed to offer the best performance with the minimum data preparation, which is ideal for the majority of the cases found (Russell and Norvig, 2003). The deep learner is incredibly precise and accurate, but it is also quite demanding in terms of performance for a system (LeCun et al., 2015). Therefore, it works as a "backup" solution when the first classifier fails to achieve acceptable results. With the data classified according to user relevance, it

is when the fourth and final phase takes place, which is composed by the analysis of the results and the creation of new data management measures. As mentioned before, this is the most fascinating part of the process due to the variety of possible measures that may emerge from the insights achieved with this procedure.

4.2 DEVELOPING AND PROCESSING DETAILS

Initially, the idea was to process an entire database. However, it was not suitable for the purpose due to the wide universe of tables that a system may have, and, in addition, some of them may not even be important to be processed. Thus, it was required to explore some alternative ways for diminishing the dimension of the analysis by selecting the most adequate tables to be processed. This way, the first auxiliary system implemented was the Hot Table System, which has the objective of analysing the database operational logs, in order to select the most important tables for system users. This is used to complete the first stage of the procedure as referred before. To determine what the most important tables are, is required to capture user operations in the system. To achieve this information, an utility was developed to analyse the session logs and the queries that were made on each. Through that, it was possible to identify which were the tables present in each session and with that information, build a Markov chain for reflecting the sequence of queried tables and their probability of being used in each session. The Markov chain is like a graph where each node, in our case, is a table. The links that connect each node are based on the sequence of queried tables in each user session, and the cost of each link is the probability of the edge node to be selected after the current node. With a chain built, it is possible to achieve a sort of map that reflects system usage. The selection process is quite complex and it is explained with more detail in the auxiliary systems section, but the main idea to retain is that this procedure builds a graph for reflecting system user usage. Through the definition of specific selection criteria, and by analysing the generated network, it is possible to choose the best tables to be processed. This process consists in an adaptation of a proposal for selecting OLAP cube views to be processed in a data warehouse system based on user usage (Rocha and Belo, 2015). Having the tables selected, the next stage is the extraction process for building the datasets to be processed, which is quite simple, since the program only has to fetch the required data and build the testing set out of it. This set is a spreadsheet, where the columns are table attributes with an extra column reserved for expressing the relevance class, and each instance of the set is a copy of the correspondent table row in the database. In Figure 7 we can see a brief sketch of how this process works.

Usually, the training set construction is more complex than the others, essentially due to the relevance factor. The training set is also sustained by a spreadsheet, having the same structure as the testing set. However, it has already the relevance attribute labelled, which is

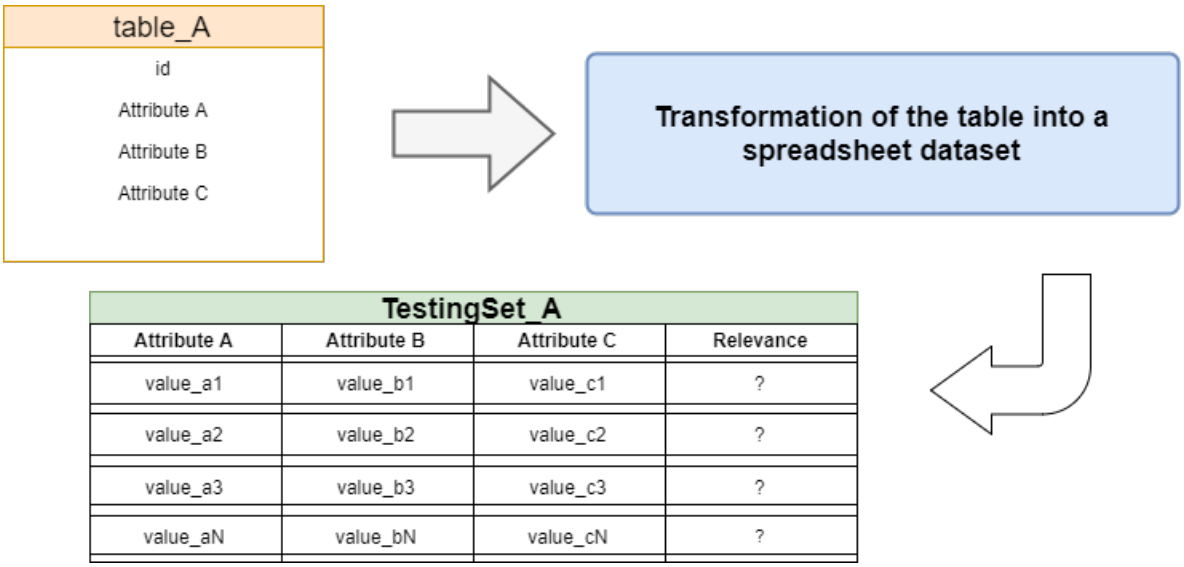


Figure 7.: The schema of the data extraction process.

the hardest part in the process. The way the solution handles cases like this, is by analysing and developing a training set through the information containing in database logs. The log records can be translated into some insights about the most queried operations, tables, and attributes. With all this information it is possible to relate it to the user relevance over data. For instance, if a table is constantly being selected, then probably is one of the most influential tables in the system as well as its data. If an attribute is always being updated, like the stock of a product, then it is one of the most important attributes in the table. The other interpretations are pretty much analogues to the previous ones and follow the same type of reasoning suggested by the procedure presented in the previous chapter. Through that the solution is going to extract the primary knowledge about the system users for performing the first scoring phase. The last procedure for building the training set is the actual ensemble of the knowledge captured from the logs into one labelled set. The way the training set is built in the first scoring phase is quite naive. Although, it works as a proof of the concept defended, as it will suffer further refinements along the process. Therefore, a log processor was developed for this purpose in which were defined, the user, the table, the instance's attributes and the type of operation for each log entry. Each record is later saved in an intermediate file format, which in this case is XML.

To process the logs and create the training set, the definition of a specific criteria was required. For instance, if we have a delete operation, then it is not wrong to say that the data from that record might not important, or even better, it is possible to teach the solution that those attribute and values for that specific table might not matter. The opposite, which

is the case of an insert operation, is also acceptable for performing this sort of inference about the relevance of the data. Queries including update and select operations follow a treatment quite similar to the insert queries, since these operations also indicate that the instance associated with the statement is being used. Thus, it is important to known who uses the system and, consequently, its relevance should be increased.

To ensemble the actual set, an operational log record has the primary keys of the instances referred in each operation. This way, it is possible to obtain the main table's instance mentioned in the operation. Its attributes and values are then added into the training set along with the class attribute. Essentially, the training set will have the same structure as the testing set (or the table itself), but it is composed by the instance of the tables referred in each entry of the operational log, with the calculated relevance grade labelled. In Figure 8 it is possible to observe the way the transformation process is executed, where we have a simple example containing a table and an operational log table having some related entries.

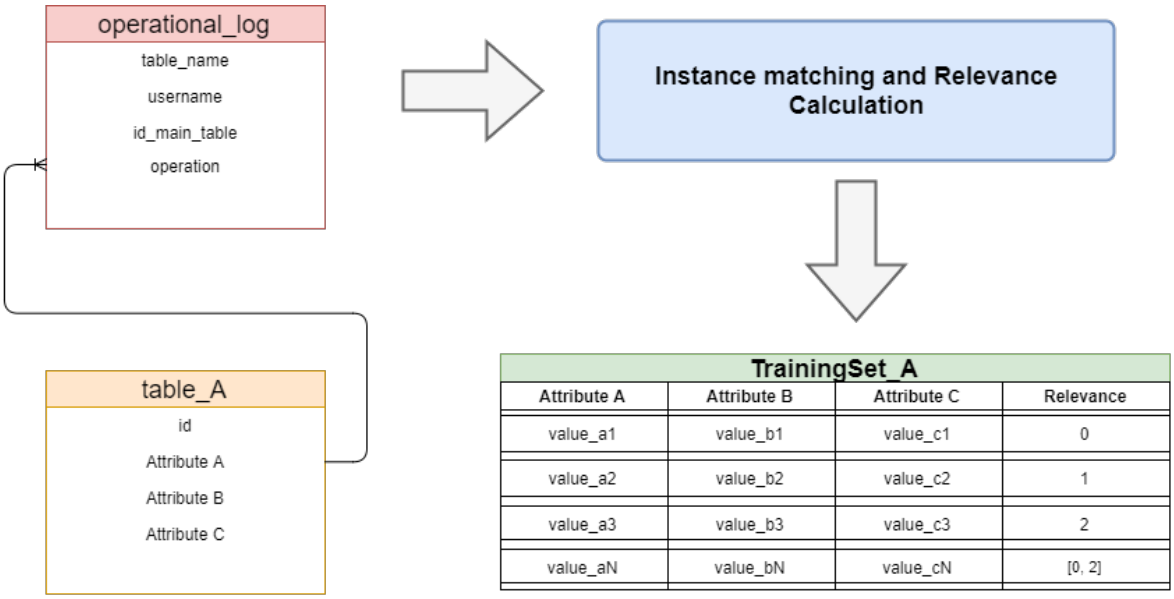


Figure 8.: Operational Logs' Transformation into a Dataset.

Having this in mind, the way the relevance attribute is calculated is quite simple. The relevance is a multi-nominal class attribute that by default ranges from 0 to 2. However, if desired, it may be changed to a new maximum defined by the user. Essentially, relevance levels are nominal values, which are represented in this case by natural numbers that are within that range. The suggested range was a mere simplification for the relevance levels results, as it is simpler to associate the three levels with the meaning of low (0), medium (1) or high (2) grade relevance classification. If one desires to create a higher level of

specificity, he may simply increase the range of the interval. To define the relevance of each row, the increment (or decrement) value is weighted based on the type of operation. The way to obtain the operation weight is by simply counting all occurrences for each type of operation found in the operational logs for that table, in order to determine the percentage of times that each operation type was queried. The percentage is then used as a weight. The relevance increment is calculated as follows:

$$Increment = MaxRelevance \times OperationWeight \tag{1}$$

After having the correspondent increment (or decrement) factor, the operation type is evaluated. If it is not a delete, the relevance grade for that instance is going to be increased. However, if it is indeed a delete operation, then it must be decreased. This is a sort of naive way for defining the relevance of each piece of data, but for proofing the concept behind this approach, it is believed to be valid. Despite that, it is still required to submit the created training set into further refinements that will adjust the initial relevance attribution based on knowledge gathered about the preferences of system user.

The reason why we used some auxiliary systems is due to the fact of the initial tests, with the only source of knowledge being the operational logs, did not provide results that were conclusive or accurate enough. Those problems were jeopardizing the reliability of the training set, which is the crucial component for determining the success of the approach. Having a consistent training set for each table is what going to determine if a particular database instance matters or not to a user. Therefore, by refining the method that constructs the training sets, it is possible to improve the accuracy of the machine learning algorithms and the overall results. Excluding the data preparation methods and relevance calculation refinements, there is only one way to improve the quality of the training set, which is by capturing trustworthy knowledge about the users and refine the dataset with that information. To do so, the idea was to develop a supporting system where users could define their own preferences over data. For example, if the user knows that the sales from a certain year are essential then it is possible to favour the relevance factor for the sales from that year. Therefore, initially, it was developed the rules and the preference systems, which, technically speaking, are different from each other in the way they influence the relevance grade.

Rules work as "hard" filters for the information whereas preferences act as "soft" filters. For instance, if it is defined a rule to favour sales from the year 2017, then the relevance factor of the instances in the sales training or testing set that are according to the rule, are set to the maximum value or (and) vice-versa. Basically, it is adjusted to a significant higher or lower level. On the other hand, a preference will benefit the relevance calculation for each

instance in the dataset that matches it, based on the attribute or the preference's weight. This way, it does not restrain the domain of the results as much as the rule system. This idea was based on the proposal made by Matteo Golfarelli and Stefano Rizzi for expressing OLAP preferences in a data warehouse system (Golfarelli and Rizzi, 2009).

The recommendation system is rather different from the other two, but as overviewed before, the purpose of this system was to create a new and easier way for defining user preferences based on the association rules that are mined from the training set. Generally, this system is optional because it is only used when there are not enough preferences or rules defined. Nonetheless, it may also be used to improve the training set after the previous scoring systems. In this case, the machine learning engine results must be evaluated with the traditional techniques to study the effect of the recommendation system in the procedure. After the auxiliary processing system is completed, the second scoring phase is finalized and the data is scaled and ready to be submitted to the machine learning engine. Analysing the process workflow presented in Figure 9, we can see the process that is executed in the prototype for creating the predictive model for classifying data.

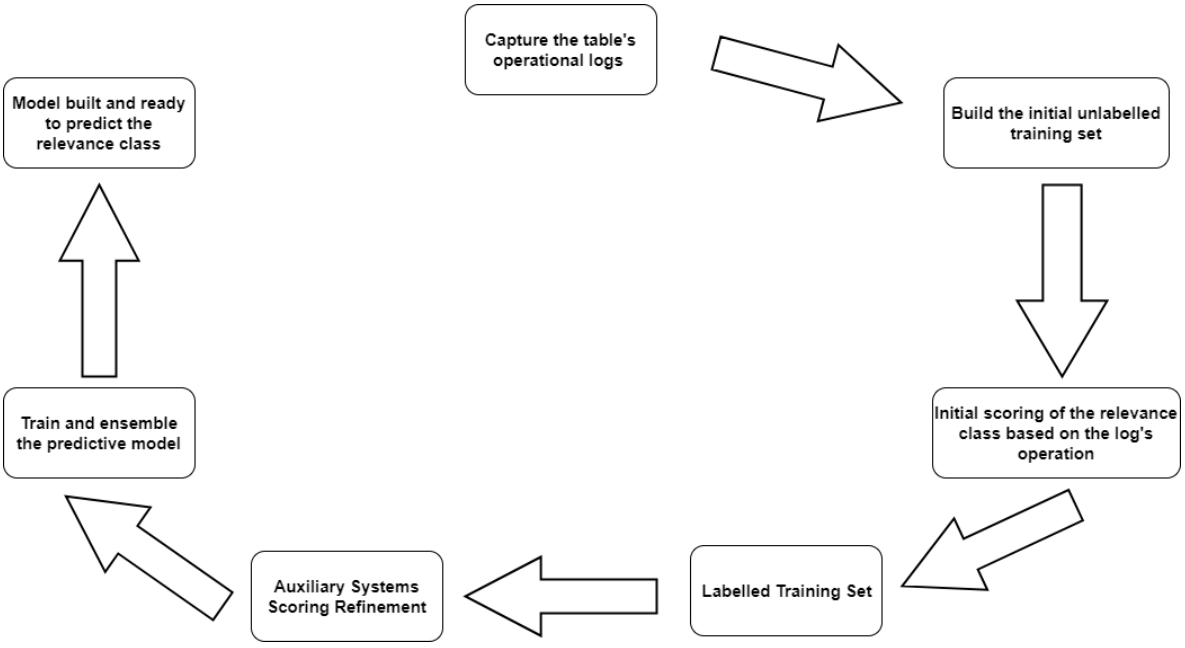


Figure 9.: Constructing process of the predictive model.

The major concept in supervised learning is to prepare the appropriate data iteratively, in order to train and evaluate a model created by an algorithm, which is then used to generate the desired model to perform the final predictions. A problem found with this solution is the variety of training sets, having different data structures and types that will demand specific data preparation procedures. The data preparation is a crucial factor for

assuring that a machine learning algorithm excels. Each algorithm and dataset has its own ideal preparation and it has to be performed generically for improving the processing of an entire database. The way to cope with this issue was to identify typical data preparation procedures for certain types of data. For example, a continuous numerical attribute can be discretized in some cases and in others it can be normalized to provide better results. The same kind of reasoning was applied to the textual attributes, for example, which can be converted to a nominal type if they are small and repetitive strings, applying if necessary data cleansing and conforming techniques, amongst others.

Another standard data preparation procedure is to identify the blank and null values, which can be identified as such or even removed for avoiding processing of incomplete data. Essentially, the idea was to identify automatically the data type of the attribute and adequate the best preparation method to treat it. This task was difficult to achieve, because it demands various complex tests in terms of data preparation techniques as well as the tuning of the algorithm to fit these generic preparations. This may seem like a utopic subject, but it is crucial to find a way for minimizing the damages of not having a generic approach to prepare the data that is submitted to an algorithm (Zhang et al., 2003).

In the prototype, a few processes that identify the type of data of each attribute and try to adequate the best generic preparation possible were implemented, which were based on the conclusions presented above. This is a measure to enable some automatism to the procedure since processing an entire database and preparing each final dataset is a very demanding task in some scenarios. Nevertheless, one can prepare each dataset individually for assuring that data preparation is the best possible, which is strongly advisable. As for the mechanisms implemented, it is important to highlight some of the most important methods. The outlier detection, data cleansing and conforming methods, discretization, normalization and standardization of numeric values, feature selection procedures and even string indexation. Some of these were implemented with the support of the data mining tool library used (Witten and Frank, 2005).

The choice of the machine learning algorithm was a hard one, because there is no consensus of what is the best to perform this kind of job. For the architecture purpose and after various tests, the Naïve Bayes algorithm implementation seemed to offer the best performance with the minimum data preparation and that was ideal for the purpose. Nevertheless, sometimes this classifier may fail to achieve acceptable results. Therefore, a solution to overcome this problem was to implement a powerful classifier such as a deep learning algorithm. This algorithm was provided by the H2O platform (Candel et al., 2017). Basically, the idea was to evaluate each model created by the first classifier through a cross-validation proce-

dure. The ones that failed to achieve an acceptable accuracy were submitted to the deep learning algorithm to build a better model. This way, it is possible to improve the efficiency of the model of the more complex training sets without jeopardizing the correctness of the procedure at expense of performance.

Deep learning algorithms are based in deep convolutional networks, which work in a similar way as the conventional neural networks, but they are able to excel in comparison with the rest. Computational models that are composed of multiple processing layers are taught with data abstractions with multiple levels, which greatly improved the state-of-the-art classification methods for large and complex problems, such as image or speech recognition. The deep learning uses backpropagation algorithm (alike the neural networks) to guide the learning of the machine in the way it should adapt its parameters to compute the various representations of each data layer (LeCun et al., 2015). The features available in the implemented algorithm also offer interesting ways to tune the learning procedure, such as adaptive learning, automatic methods to train models. Generally, it offers a large variety options to configure every aspect of the model's training. The automatic method to train and evaluate multiple models is a very interesting one. It is able to train various models with different configuration parameters and choose the best, based on the evaluation performed for each. This is a very demanding process in terms of performance. However, it is the most efficient and adequate for the purpose of the prototype (Candel et al., 2017).

Other classification methods were evaluated, such as the Neural Networks Multi Perceptron algorithm, which has an amazing capability to learn and a high predictive power. Despite that, it has a major downside, which is the performance. Thus, it did not offer better results than the deep learning implementation (Witten and Frank, 2005). Another example tested was the incremental classifier algorithm IBK, which is a lazy algorithm that can learn and train the model incrementally. This algorithm was an idea for reducing the retraining time of the model compared to other algorithms when data changed and more logs were gathered. It delivered acceptable predictive results. However, it is also very demanding in terms of performance and the only purpose was to accelerate the retraining process. (Zhihai Wang and Webb, 2002). Moreover, for some cases, where new data and logs are constantly being generated, it is best to pursue better and more efficient alternatives to select the training set's data. Some applicable options are the semi-supervised learning methods, data selection through clustering techniques and analysis of the available data (Madasamy and Tamilselvi). Other algorithms, such as the tree based J48 were also tested and were able to deliver results as good as the Naïve Bayes's for some cases, but lacked in the performance in other cases (Ahmed and Jesmin, 2014).

Some other interesting techniques, like the clustering methods, where not explored in the presented prototype, though they may also be used to achieve the desired classifications. These techniques, are for some the most obvious to use here, because it is an unsupervised method that does not need user knowledge for inferring data. That may not be the most suited approach, but if results are relevant they could be tested and used. Still related with clustering, it could also be used as a way to refine the classification methods, for example, by only selecting data from certain clusters for the classification process. This way, the analysis's dimension is reduced and some outliers could be removed (Madasamy and Tamilselvi).

Having the algorithm strategy defined, the procedure is able to classify information with a good balance between performance and quality of the predictive models. With the testing sets classified it is possible to adequate the best management measures for handling data. In this architecture proposal the idea is adequating the best measures to each particular system based on their needs. These measures range from a wide universe of options that are originated from the analysis of the relevance classifications. Each system administrator must be able to decide and evaluate which are the best to solve their problems. The important fact is that it is possible to define measures to improve the quality of the system.

4.3 THE AUXILIARY SYSTEMS

The auxiliary systems represent a huge part in the solution developed. Without them the success of the process could be jeopardized. The reason for it is that some of the problems require alternative methods and other supporting procedures for ensuring a correct solution. For example, in cases where the operational logs are not able to provide enough information about the usage of the system, it is required to develop new alternatives for capturing and refining the knowledge gathered. Other auxiliary systems may be related to some performance issues or any other problem that may occur when applying the procedure in a system. Therefore, one must be able to identify these issues and formulate a solution for solving them.

4.3.1 *A Hot Table System*

The process of classifying data through machine learning methods is quite expensive for a system. Given the fact that an entire database is going to be processed, the performance is going to be a critical factor. Thus, reducing the dimension of the analysis is a great idea to improve the execution of the process. To select the most adequate tables to be processed, it was implemented a "Hot Table System". Basically, it consists in an adaptation of the pro-

posal made by Rocha and Belo (2015), in which is presented a solution for selecting OLAP cube views to be processed in a data warehouse system based on its usage.

In this case, the idea is to identify the most important tables for a user in a given system. Thus, we need to capture their behaviours when they are using the application. To extract that information, it was created a procedure that analyses the database logs and identifies the tables that were queried in each user session. To illustrate a possible sequence of sessions in a database, Figure 10 presents a small example of what are the results when executing the referred procedure. The queried tables are represented by the nodes located between the "INIT" and the "END" node, which represent the start and the end of a session, respectively.

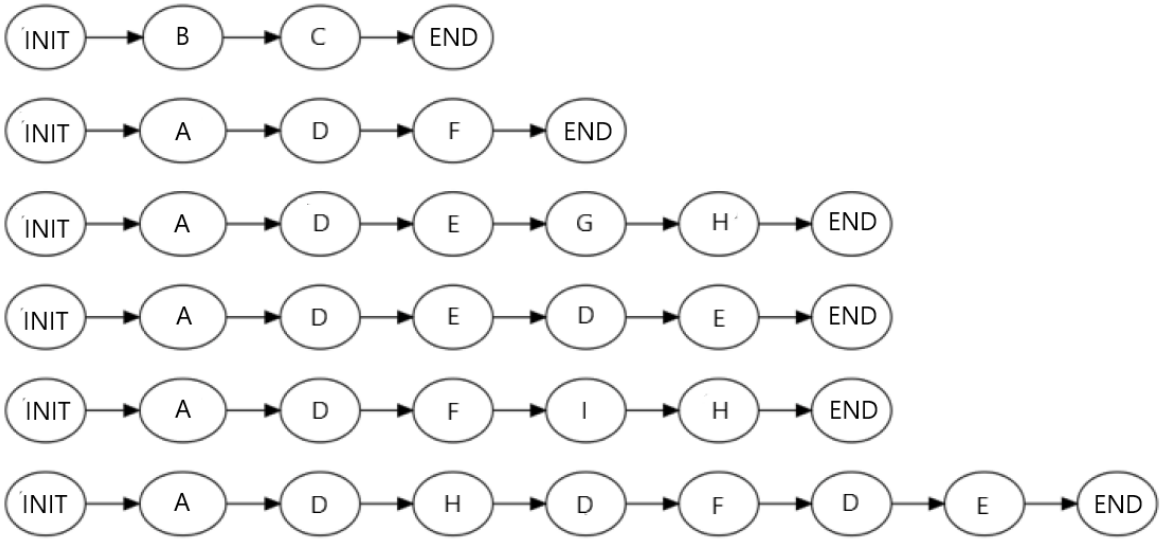


Figure 10.: The Sequence of Queried Tables of a querying session.

After this, it was possible to know which tables were queried in each session, and build a Markov chain for reflecting the sequence of tables and their probability of being queried in each session. A Markov chain is like a graph, in which in this case, each node represents a table. (Gagniuc, 2017) The links are based on the sequence of queried tables in each user session, and the cost of each link is the probability of the edge node to be selected after the current node. In Figure 11, we can see the resultant Markov chain based on the example presented in Figure 10.

With the chain built, it is possible to achieve a sort of map reflecting the usage of a system and then prune less frequently queried nodes, if necessary. To do so, it is defined a minimum probability for each link. Those that do not meet this requirement are removed

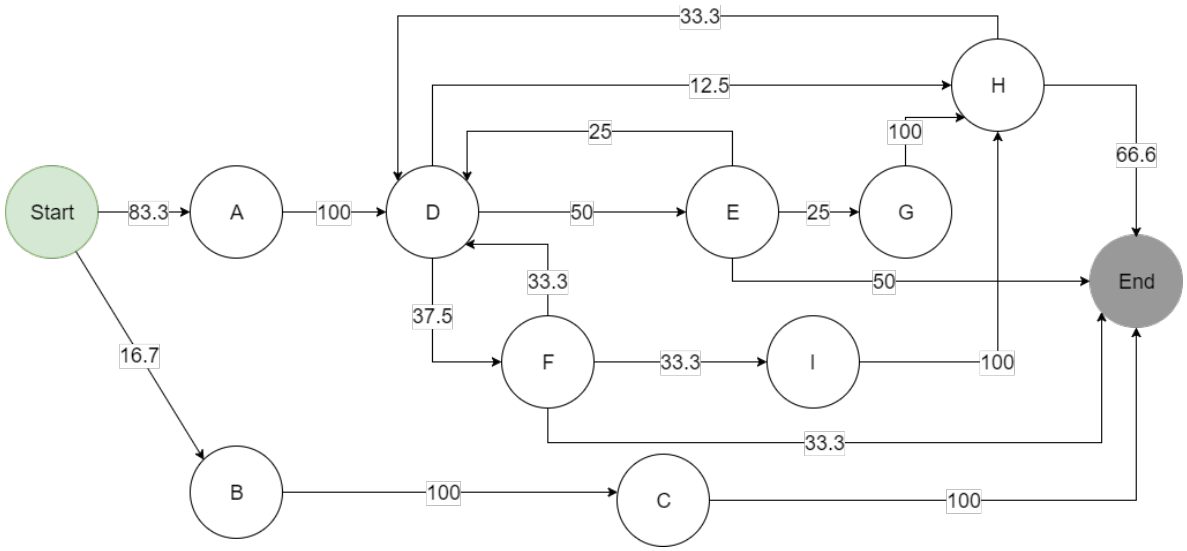


Figure 11.: The Sequence of Queried Tables of a querying session.

from the chain. Finally, the nodes that are not isolated, which means the ones that can be reached from the initial node (session's starting point) and also the final node (session's ending point) must be reachable from themselves, represent the most frequently used tables of the system and the ones that should be taken into further analysis.

The way the system infers about the relevance of a table is by colouring them, distinguishing stronger colours ("hotter") from lighter ones ("colder") based in three major criteria. These are weighted before the process and each table has to verify them in order to sum the weighted colour value designated for each criteria. Therefore, the minimum usage criteria is related with the minimum usage rate that a table has to have, ensuring that less frequently queried tables are left out of the selection. The minimum space criteria is related with the space rate that a table must have to be coloured according to this criteria. Thus, through this, it is possible to define a minimum dimension for a table allowing for larger tables to become more relevant even if they are not so queried. Lastly, the maximum space criteria is used for evaluating the size of a table, so that it meets the desired dimension reduction limit, which is also defined before the procedure. After the validations are completed, each table will have a colour designated and one can select the most adequate tables by analysing their colours in the Markov chain. This selection can be achieved with a simple filter that extracts the tables that have a colour above a certain value. The next example, presented in Figure 12, represents a coloured graph without the weaker links and isolated nodes, which are represented with dashed lines. The result will be a Markov chain where the final nodes are coloured according to the defined criteria. Shades coloured red are assigned to the hottest tables, while the ones coloured blue are meant to represent the

least relevant structures.

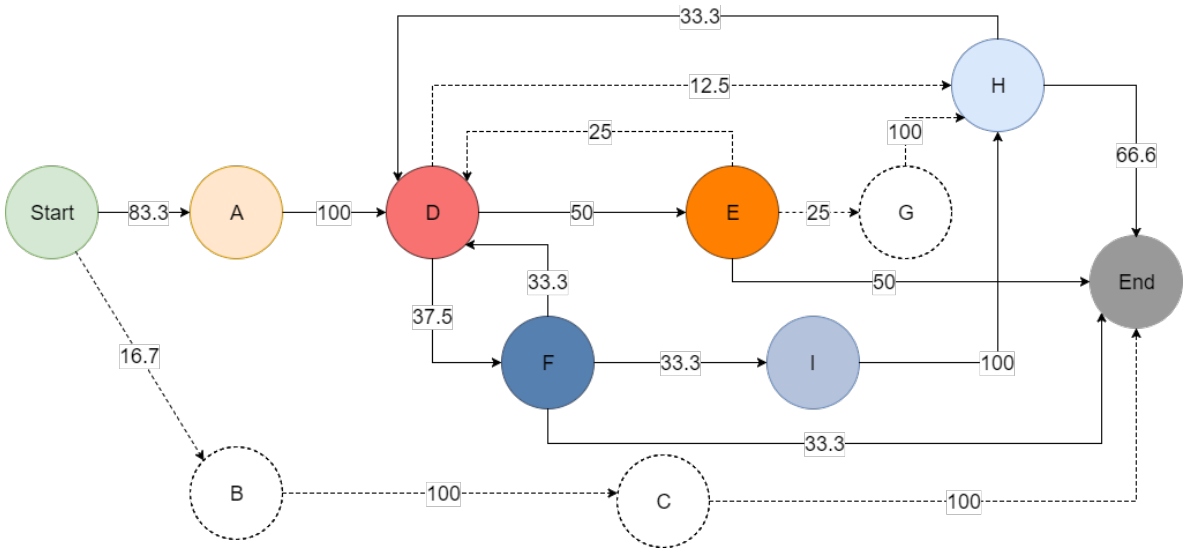


Figure 12.: The final coloured graph without the nodes removed.

At the end of the process, it is possible to get the most influential tables to be taken into the next phase, avoiding the cost of processing irrelevant tables, which may drastically jeopardize system performance. Another interesting fact is that through this procedure, it is possible to identify where the master data of the system is and ensure that the next phase will process the most important data of the system, which is where most of the management problems are. The data not selected, may not be important and can be identified as such, so that it can be treated accordingly to some administrator's decisions.

4.3.2 *Preferences and Rules Systems*

In order to establish more interactions with the user, it was decided to create a system where it is possible to define his preferences over data. For example, if a user knows that the sales from a certain year are essential, then it is possible to favour the relevance factor of the sales from that year, and so forth. With a system like this implemented, the initial scoring can be significantly improved for reflecting the current relevance over data given by a specific user. This kind of refinements of the initial classifications of the training set are crucial for ensuring that the solution will be able to provide correct and useful conclusions. The training set is probably the most important piece of the whole solution, due to the fact that every result is going to be dependent on it given the supervised approach implemented. Thus, it is very important to certify that the training data of the model is

robust and able to infer correctly. For implementing a system like this, it was required to create an intermediate format, in which users can express their preferences. After that, it is needed to implement an interpretation method so that it is able to understand the defined preferences and adjust the relevance grade of each instance. Since users may have some preferences stronger than others, which mean that some clauses are more important than others, it was also developed a way to distinguish these cases of preferences. Therefore, it is possible to define preferences and rules, which, technically speaking, are different from each other.

Rules act as "hard" filters for the information whereas preferences perform as "soft" filters. For instance, if it is defined a rule to favour the sales from the year 2017, then the relevance factor of the instances in the sales' training set that are according to the rule are set to the maximum value or (and) vice-versa. On the other hand, a preference will adjust the relevance calculation for each instance in the dataset that matches a preference, based on the preferred attribute weight or preference weight. This way, it does not restrain results as much as the rule system (Golfarelli and Rizzi, 2009).

As for the calculation methods, rules have two processing styles, in which one is smoother than the others. The purpose is to influence the classifications in more than just one way, so that the user is able to evaluate and choose the calculation method that suites his needs. In the restrictive mode, the rules are able to maximize or minimize the relevance of an instance that complies (or does not) a rule. The smoother method will only increase the relevance factor by one level or decrease it, if that is the case. The decision of whether the relevance grade is increased or decreased is related with the instance match with the rule and its conditional operator.

Preferences are quite similar to rules in the way that it is possible to select distinct calculation methods for adjusting the relevance levels of the instances in the datasets. In the first case, the calculation method is based on the preference weight in the set, which is calculated as:

$$PreferenceWeight = \frac{1}{NPreferences} \tag{2}$$

This weight is then related with the maximum relevance value to achieve the final increment value for each instance that verifies a preference, which may be obtained with:

$$Increment = \pm MaxRelevance \times PreferenceWeight \tag{3}$$

Through the previous method, it is possible to define just a set of crucial preferences and influence the relevance levels with a greater impact. However, when the specificity and the

level of adjustment have to be acutely precise, then it is probably a better choice to define the alternate calculation method, which is based on the attribute's weight in the dataset. This way, the weight of each preference is calculated according to the following formula:

$$PreferenceWeight = \frac{1}{NAttributes} \tag{4}$$

The increment value designated for each preference will be calculated through the formula (3). The way the increment will affect the current relevance grade is by analysing the instance match with the preference and its conditional operator. Therefore, if there is a positive match, the increment will be positive. On the other hand, if the instance does not match the preference according to its conditional operator, the increment should be negative. With these two methods for calculating the influence of each preference, it is advisable to evaluate the options and choose the best for each dataset. Having these options and methods of calculation for both systems is a matter to be discussed when implementing the procedure and these systems in particular, in order to ensure the best configuration for each database.

Other kinds of influence that are different from the ones proposed in this document are also viable. Although, the idea should be the same, which is to refine the initial knowledge gathered from the operational log's translation. In the presented implementation, the program can import the rules and preferences through a XML file. The structure is very simple and similar for both cases. Basically, each rule or preference has a condition attribute that defines the conditional operator, which can range from the simple equality ($==$) to the greater-equal or less-equal ($\geq$) operator that can be used to compare numeric values. The remaining attributes that compose the XML entry are entitled to represent the attribute and the correspondent value. In the following examples (Listing 4.1 and Listing 4.2), it is possible to observe a rule set (Listing 4.1) and a preference set (Listing 4.2) in the XML format. Note that the "nAttrs" attribute is strictly auxiliary in the context of the given example and it was meant to indicate the number of attributes of the table.

```
1. <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2. <Rules>
3.     <Rule attribute="ContaOrigem" condition="==" nAttrs="48" table="[dbo
   ].[Movimentos]" val="UNKNOWN"/>
4.     <Rule attribute="Mes" condition="!=" nAttrs="48" table="[dbo].[
   Movimentos]" val="9"/>
5. </Rules>
```

Listing 4.1: Example of a Rule Set Interpretable by the Prototype.

```
1. <?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<Preferences>
    <Preference attribute="Utilizador" condition="==" nAttrs="48" table="[dbo].[Movimentos]" val="marta"/>
    <Preference attribute="Iva" condition="!=" nAttrs="48" table="[dbo].[Movimentos]" val="21%"/>
    <Preference attribute="TipoTerceiro" condition="==" nAttrs="48" table="[dbo].[Movimentos]" val="0"/>
</Preferences>
```

Listing 4.2: Example of a Preference Set Interpretable by the Prototype.

4.3.3 *Recommendation Systems*

In order to simplify the process of defining preferences and rules for the table, we designed and implemented a recommendation system. The problem that motivated the creation of this system was the fact that having to define a set of preferences and rules for every table that was going to be analysed, the process to do that is very demanding and requires a complete knowledge about the system's data. Sometimes, this process is not possible to complete, and the recommendation system enables the possibility of automatically generate a new sort of that are suited for each dataset. The recommendations are a result of the knowledge gathered from the analysis of each training set. The way this is achieved is through the exploration of a data mining technique that processes and combines data in order to discover patterns (and other kinds of insights) about the way data is related. This technique is called as Association Rules Mining. These rules are basically combinations between all items in a dataset with their attributes and values, which are related in a form of an implication $X \rightarrow Y$, where X and Y are disjoint itemsets. The X set is the antecedent of the rule, while the Y set is composed by the consequent of each rule, representing the consequence set (Agrawal et al., 1993). For a better understanding, Let us consider I, a set of attributes called items, and T, a set of transitions. Each transaction t has a subset of items from I. For instance, imagine that $I = \{action, horror, thriller, comedy, sci - fi\}$, which are binary attributes and a transaction set represented in Table 1.

transaction ID	action	horror	thriller	comedy	sci-fi
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	1	0	0
5	0	1	0	0	0

Table 1.: Transaction Table.

Analysing the set of transitions it is possible to obtain a rule, such as $\{thriller, horror\} = \{action\}$, because every time there is "thriller" and "horror" (antecedent) equal to 1 in a transaction, from the observation of the table it is implied that the "action" (consequence) attribute is also equal to 1. Analysing this rule in a marketing perspective and imagining that I is a set of movie genres and t is a person's response to an enquiry of movie desires, it is possible to conclude that if someone likes thriller and horror movies, most certainly he also like action movies as well. Basically, through the analysis of association rules, it is possible to find some patterns about data and correlations between the distinct attributes in the datasets. To evaluate the correctness of these rules there are certain important metrics to achieve that. For example, the confidence and support of the rule are probably the most important ones. However, there are other metrics suited for different purposes of evaluation that may also be useful, such as the lift and the conviction. Focusing on the support, this metric indicates how frequently an itemset appears in the dataset and it can be calculated as:

$$Support(X) = \frac{(\mid \{t \in T; X \subseteq t \mid\} \mid)}{(\mid T \mid)} \tag{5}$$

The confidence is a metric that defines the percentage of times that a rule was found to be true in the dataset. This metric is can be calculated through the following formula:

$$Confidence(X \rightarrow Y) = \frac{(Support(X \bigcup Y))}{(Support(X))} \tag{6}$$

Analysing these two metrics applied to the previous rule, where $X = \{thriller, horror\}$ and $\mid T \mid = 5$, then the support of the rule is 20%, since the antecedent only appears in one transaction out of the five total transactions. The confidence in this case is 100%, because in the only time the antecedent is verified in the transaction set, so is the consequence and therefore, the confidence is the maximum possible (Tan et al., 2005). These two metrics in particular are very important to evaluate the importance of a rule. In the example, the confidence guarantees that the rule will always confirmed, but since only 20% of the transactions verify the antecedent of the rule, it may not be such a relevant rule after all or it may be a surprising pattern that was not yet discovered. Analysing these two metrics is very

important to decide if the information given by the rule is precise or relevant to consider.

In the system we implemented, the way rules are mined is through an implementation of the Apriori algorithm provided by Weka (Sumithra and Paul, 2010). Through this implementation, it is possible to set a special option that allows for the consequence of each rule to be related with the class attribute, which in this case is going to be the relevance factor. Therefore, the implication's consequence will be directly related with the label attribute. The rules are mined over the training set after it is labelled by the first scoring phase, achieved through the analysis of the operational log. This way it will have some knowledge about the relevance of each instance in the set and it is possible to mine new association rules from it. It is also a good idea to mine the rules after the second scoring phase, which is after the rules and preferences systems, in order to have refined knowledge from the first scoring phase before mining. The idea of this system was to avoid the rules and preferences definition. But since it may also help to refine the training set, exploring this idea for that purpose is also viable and advisable.

The algorithm generates the best rules based on the information that was submitted for analysis and the way the selection is made is by evaluating each metric of the rules. The confidence is the percentage of times that a rule is verified amongst the possible instances that verify the antecedent of the rule, while the support is the percentage of times that the rule can be found in the dataset. With these metrics it is possible to restrain the scope of analysis to capture only rules with a minimum support and confidence, and that is the way they are selected from the generated set by the algorithm. The resultant rules are exported into a XML file readable by the prototype, and the structure of each recommendation is pretty simple and similar to the rules or preferences. In this case, each recommendation will be composed by an attribute for each premise and consequence, together with the evaluation metrics of the rule. The calculation method to adjust the relevance of each instance is quite different from the rules or preferences. Thus, since there are metrics to evaluate the recommendation, the idea was to make use of them once again. This time to calculate the increment or decrement of the relevance grade for each instance. Hence, when an instance verifies every premise of a given recommendation, its relevance level is adjusted based on the consequence relevance of the recommendation value and the current value of the instance. If the current instance's relevance value is greater than the recommendation's, then the resultant adjustment should be a decrement of the current relevance grade and vice-versa. When they are both the same, nothing is done. The increment or decrement value is calculated using the support metric of each recommendation. Since the support is related with the number of times that the rule is verified in the dataset, it could be used to

weight the consequent relevance increment value. Therefore, the increment (or decrement) of the recommendation value can be calculated as follows:

$$Increment = MaxRelevance \times RecommendationSupport \tag{7}$$

This way, the training set is further refined and provide better knowledge for the machine learning techniques. In Listing 4.3 it is presented a small set of generated recommendations in XML.

```
1. <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2. <Recommendations>
3.   <Recommendation Relevance="2" availability="(80-inf)" confidence="
     0.9020979020979021" costrate="(16.666667-inf)" support="28" table="
     Production.Location"/>
4.   <Recommendation Relevance="1" availability="(80-inf)" confidence="
     0.9020979020979021" costrate="(16.666667-inf)" modifieddate="
     2008-04-3000:00:00.0" support="28" table="Production.Location"/>
5.   <Recommendation Relevance="0" confidence="0.8928571428571429"
     locationid="(40.333333-inf)" support="22" table="Production.Location"/>
6.   <Recommendation Relevance="1" confidence="0.8928571428571429"
     locationid="(40.333333-inf)" modifieddate="2008-04-3000:00:00.0"
     support="22" table="Production.Location"/>
7.   <Recommendation Relevance="1" confidence="0.8915094339622641"
     costrate="(16.666667-inf)" support="42" table="Production.Location"/>
8.   <Recommendation Relevance="1" confidence="0.8915094339622641"
     costrate="(16.666667-inf)" modifieddate="2008-04-3000:00:00.0" support=
     "42" table="Production.Location"/>
9.   <Recommendation Relevance="2" confidence="0.881578947368421"
     locationid="(20.666667-40.333333]" support="15" table="Production.
     Location"/>
10.     <Recommendation Relevance="0" confidence="0.881578947368421"
     locationid="(20.666667-40.333333]" modifieddate="2008-04-3000:00:00.0"
     support="15" table="Production.Location"/>
11.     <Recommendation Relevance="1" availability="(80-inf)" confidence="
     0.8783783783783784" locationid="(40.333333-inf)" support="14" table="
     Production.Location"/>
12.     <Recommendation Relevance="1" availability="(80-inf)" confidence="
     0.8783783783783784" locationid="(40.333333-inf)" modifieddate="
     2008-04-3000:00:00.0" support="14" table="Production.Location"/>
13.   </Recommendations>
```

Listing 4.3: Example of a Recommendation Set Interpretable by the Prototype.

4.3.4 *User Behaviour Detection System*

The knowledge about system usage is the key for the success of what we are proposing. It has been reviewed before that, in order to have useful classifications, training sets must contain accurate knowledge for providing meaningful results. This way it is possible to train the predictive models in an efficient manner. The scoring systems are based on user preferences and operations that are made in the system. Therefore, the idea behind the creation of this system was to gather more knowledge, but this time from the user behaviours on an application level. By being on an application level, it is meant that the user behaviour is gathered from actual operations in the software that is supported by the database. This way, it is believed that this approach may bring different and improved meanings to some of the previous classification scores, as well as work as an alternative way for processing operational logs, depending on the implementation and purpose of the system one has. In our case, it was built with the purpose of being an incremental scoring system.

The behaviour detection system was implemented based on the case study we referred previously. Therefore, it was developed to simulate the use of certain features in the application that were found to be useful for refining datasets, being applicable only for this piece of software in specific. Other approaches for developing a behaviour system are also viable options, but should be conceptualized and developed around the information system in question with the purpose to extract the most valuable knowledge for the classification process. The presented proposal supports that a system of this kind can be used as an incremental scoring system, and so it is not specified in the diagram of the process. Nevertheless, this sort of system can also be used as an alternative to the database logs system, if it is built with the demanded specificity that will induce enough knowledge for constructing powerful predictive models. In this case, the software has certain key features, such as quick-searches, help suggestions, menu clicks and inputs or even the windows that are opened. These compose a set of important behaviours that may be captured and used in a new scoring system. Hence, the strategy for capturing them was simple, since it was only required to capture the clicks (and some user inputs), saving them into an intermediate format in the end of each session. This format can be anything from files to database tables. However, the approach followed was to simply save the behaviours in a XML file with a structure similar to the previous ones that were used in the auxiliary systems presented along this chapter. After all behaviours gathered, the following step is to calculate and refine the relevance grade of the datasets, according to the acquired knowledge. This procedure is quite similar to the other scoring systems implemented. The diagram presented in Figure 13 illustrates this process in a very simple way.

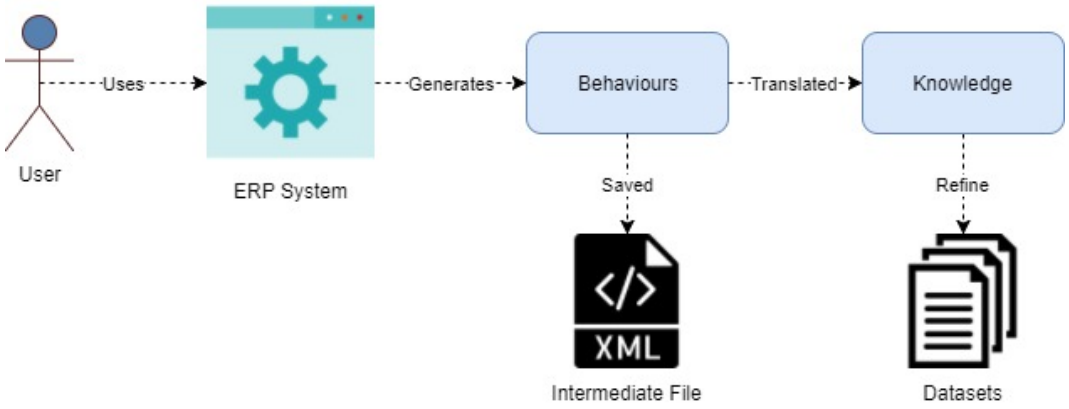


Figure 13.: User behaviour detection system workflow diagram.

The way the system scores the instances is quite different from the rest, because there are different types of behaviours providing distinct perspectives over data. Firstly, it is required to understand the types of different behaviours that exist. As introduced before, the features that may be used are specific button clicks, inputs and suggestions used in the application. More specifically, one of the behaviour types is the "F4_Intel" and "F4_PopUp", which are related to suggestions that may be used to fill a form field. The first one is word auto-complete, which is based on previous typed values and the other is a value that was selected in a window with various suggestions of inputs. Another type of behaviour is the "Help", which is triggered when a user uses a suggested help notification. These alerts are used to advise or inform the user about certain sections that require supervision or some sort of action in the ERP system. The use of these suggestions trigger the opening of the ERP section windows that have tables related, either to display listings with information from the database or associated with, for example, operations like updates or inserts in case there are forms inside the page. The tables related to each window may be captured and used as knowledge about important tables that are used in the software. The "Window" behaviour is pretty much the same as the previous one, but it is triggered when a user clicks in a section of the ERP system. The last behaviours are the "Menu_Click" and "Menu_Input", which are captured when a user clicks in an item from a menu that is associated with a table (for example, a product from a list of products), and the last is captured when a user inserts a certain input in the software, such as form fields. These two provide information related to the table, attributes and their values, which compose a rich source of knowledge that can be used. the Table 2 we can see an example of each type of feature in the software that triggers a behaviour related with the information they provide about certain data entities, such as tables, attributes or values.

	F4_Intel	F4_PopUp	Help	Window	Menu_Click	Menu_Input
Table	Yes	Yes	Yes	Yes	Yes	Yes
Attribute	Yes	Yes	No	No	Yes	Yes
Value	Yes	Yes	No	No	Yes	Yes

Table 2.: Feature and Type of Information Correspondence Table.

Having the different types of behaviours presented and how they are triggered, it is also important to understand the way the auxiliary system is going to influence the relevance levels of the training sets. Hence, the objective was to gather the most frequent tables, attributes and values from the captured behaviours. For each behaviour, the associated tables can be used for improving the table selection process, for example. In addition, the attributes and values are exactly the kind of knowledge that can be used to refine the datasets. This way, the scoring system first analyses the behaviour set and counts every occurrence of each attribute and their values, in order to create a map with the totals of each distinct attribute and value. With each map it is possible to have a sort of ranking of the most frequent attributes and their values. In this approach, these are the most attributes and values of each respective database table. Therefore, the table instances having such values in their key attributes represent the most relevant information for the user, and so its relevance should be adjusted accordingly. After the rankings are completed, the user may define the set of top attributes and values that he wishes to evaluate in the scoring process of the instances. For example, if there are 5 top attributes and 3 top values for a given table, it means that each instance is going to be evaluated on each of the 5 key attributes, if the values are any of the top ones. The way the relevance is calculated is going to be based on the top N attributes' weight in the total of attributes identified in the table's user behaviours set. More specifically, the system first identifies the main attributes and weights each one based on the percentage of the total occurrences of each one in the behaviour set. The weights are calculated as follows:

$$AttributeWeight = \frac{TopAttributeOccurrences}{TotalTopNAttributeOccurrences} \tag{8}$$

The weight will be used to calculate each relevance increment based on the matched value of each attribute. Since not every attribute was referenced equally in the behaviour set, their weights have to balance the resultant relevance increment for reflecting the importance of that value based on its attribute. After the top attributes are weighted, the training set instances are analysed, and if they match any of the top values of each attribute, a

relevance increment is calculated according to the weight of that specific attribute. This is done according to the following formula:

$$Increment = MaxRelevance \times AttributeWeight \tag{9}$$

The process is repeated for each selected attribute until the final increment is achieved and after the instance is evaluated. The increment is positive when the instance verified at least one of the top attributes values, and, therefore, its current relevance level should be increased by the calculated increment. In the opposite case, the increment should be negative to decrease the current relevance grade. This way, the relevance adjustment is calculated using the following expression:

$$CurrentRelevance = \pm Increment \tag{10}$$

Through this process, it is possible to increase the level of specificity of the training sets and improve their quality by refining its knowledge. Having this system to support the previous scoring systems is a great addition, since it is very important to make use of any source of knowledge for improving the results of the solution and reducing possible errors in training sets. The implementation described in this section is just a mere suggestion of a possible system that could be created. Since this option may provide the best source of knowledge for some systems, one must be able for evaluating the problem in which this process is being applied, in order to identify the necessity and purpose for the creation of a system like this. The adaptation is up to each developer since the objectives and scenarios are different from the ones presented. Finally, as to exemplify a behaviour set in XML format readable by the prototype, in Listing 4.4 we present an example of a recommendation set interpretable by the prototype we developed.

```
1. <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2. <UserBehaviours>
3.   <Entry attribute="ConfrontacaoNascente" table="[dbo].[Fichas]" type="
     MENU_INPUT" user="USER_TEST" val=""/>
4.   <Entry attribute="SubDivisao2" table="[dbo].[Fichas]" type="F4_POPUP"
      user="USER_TEST" val=""/>
5.   <Entry attribute="Cedido" table="[dbo].[Fichas]" type="MENU_INPUT"
     user="USER_TEST" val="0"/>
6.   <Entry attribute="TipoBem" table="[dbo].[Fichas]" type="MENU_CLICK"
     user="USER_TEST" val="null"/>
7.   <Entry attribute="NaturezaDireitos" table="[dbo].[Fichas]" type="
     MENU_CLICK" user="USER_TEST" val="null"/>
8.   <Entry attribute="null" table="[dbo].[Fichas]" type="WINDOW" user="
     USER_TEST" val="null"/>
9.   <Entry attribute="AnoUtilizacao" table="[dbo].[Fichas]" type="
     F4_INTEL" user="USER_TEST" val="2010"/>
```

```
10.      <Entry attribute="TxPerdidasEx" table="[dbo].[Fichas]" type="
    F4_POPUP" user="USER_TEST" val="0.0"/>
11 12.   </UserBehaviours>
```

Listing 4.4: Example of a behaviour set interpretable by the prototype.

5

THE CASE STUDY

5.1 GENERAL OVERVIEW

To better understand the system's environment of the proposed solution, it is important having in mind some details about the PRIMAVERA's ERP system architecture, and also about some other particular issues and remarks related to the data that is handled and its management technicalities. The ERP system is a powerful piece of software that is able to support all kinds of business activities. An example of a problem that might eventually endanger a system database is the historical information. The provenance of that type of data might be from previous records of sales, orders, inventories or any other historical information that a company might handle. Another interesting example could be the less frequently used information. The effort to keep those old records stored is quite impactful and unworthy performance wise. Besides that, if the perspective angle is the quality of the data, it certainly will not be any better. For instance, if that happens, some of the analysis services offered by the ERP system might not be giving quality information to users, which might jeopardise the decisions of the company. Furthermore, some of the company's clients may have low-end systems to support the ERP system, due to the fact that some of them may just be a simple interface between business activity stores and the mainframe. Clients that have a huge amount of transactions are certainly going to have some of their machine's performance stalled, which in most cases implies an expensive upgrade in order to run the products of the company. This is entirely undesirable, since one of the most important requirements is that the ERP must be able to be executed in the large majority of the systems with a fairly good performance. Otherwise, clients may choose a different product for supporting their activities. Bigger clients may have better resources to support their businesses, but their transaction volume is huge, which may compromise the performance of some of the client's machines, especially if time is a key factor. Another issue is the limitations of the ERP system. Usually, these issues are related with the legacy version of the software that only allows for using a limited set of system resources. Thus, high-end machines will suffer from this issue as well as if the dimension of the data is extremely high.

In the perspective of the company, the solution of those problems is crucial to hold the present client base, as well as to attract new possible ones due to the performance optimizations. Attending to those reasons, a solution capable of identifying the relevance of each piece of data is extremely valuable, because it would ignite the creation of new and adequate management measures for improving system performance and data quality. Some of the measures that could be implemented in this particular case are related with dimensionality reduction. However, there are other interesting options to be explored. As mentioned before, to improve systems performance on accessing the most valuable information, it could be implemented a cache memory system for favouring the access to data. With an analogue but inverse reasoning, one could export the least important data to a different structure and avoid the over-occupation of the main computational platform. By creating these kind of auxiliary structures, data is not entirely deleted and it is available in the secondary systems if needed, avoiding the complete loss of data. However, if the intention is to remove it completely, it can be easily achieved if there are no problems in terms of data integrity.

Since some of the clients might be struggling with data quality for analysis purposes, the identification of the relevance factor for the information is extremely value in this case as well. By identifying the most relevant data, it is possible to discover new insights about the client's businesses, which is crucial for them. Another interesting observation is that the data for analysis might not always be up to date and contain misguiding information. By mining and analysing the results, the corrupted data may be identified and removed to improve the quality of the business analytics systems of the ERP system. This way, the quality of the analysis is ensured and it confers a feeling of reliability to the analysis of the clients. Another example of what may be discovered from the analysis of the results are patterns amongst the different levels of relevance, which may provide interesting results for business purposes and data management. These data patterns could be mined with association rules techniques and even through clustering methods, which are ideal for that purpose.

Being contextualized the information ecosystem and its issues, it is also important introducing a quick overview about the system data infrastructure. Hence, the ERP is held under a relational database with a SQL Server DBMS. The database schema has more than a thousand tables, which can have up to millions of records, making it a considerable big database. The complexity of these systems is daunting and a crucial objective for them is to be able to maintain a good performance even on low-end systems that some clients might have, as already mentioned. To mitigate some of the problems, the ERP system's infrastructure already has some optimizations and governance measures defined. For handling large

historical records, like the sales history and its values, the system uses index aggregation over the previous records. Therefore, instead of saving every single record, the system only saves the aggregated value from the past instances, avoiding the storage of large amounts of old records that are only used to sum a value. Other example is the case when a particular table is only accessed to get the distinct values from it. Query performance for achieving the distinct values from a table may be quite expensive, especially if the dimension of the data is too big, having a great variety of values. To overcome this, the technique that is being used for solving the case, is to save all the distinct values in a second table offering less latency when queried, preventing the system's effort of calculating all different values every time they are needed. The idea behind this method was to create a sort of materialized view for favouring the access to heavy processing information. In this case, the system will only have to select the entire content of a smaller table that was designed for that purpose. As for the materialized views, these are a very interesting concept that was used in the ERP system in other cases, in order to avoid high latency queries to some of the hardest processing queries like table joins, aggregations and so forth. In the software, there are also other examples where auxiliary tables act as supporting structures to ease the access or the processing of some particular operations, just like the distinct example. Lastly, the system also benefits from some of the traditional DMBS optimizations, such as the query planners, database caching and so on.

All the tunes mentioned above are not the only ones implemented in the system. However, despite having a fair consideration for those measures, some systems might still suffer from a lack of performance due to excessive amount of disposable data. Relating the current data-driven problems with the ERP system ones, a solution that is able to support the management problematic is seen as a great opportunity to engage the main problem, and start the preparation and refinement of the capabilities of the system for facing this next generation of business applications.

5.2 APPLICATION SCENARIOS

In order to understand the tests that were conducted to evaluate the prototype, it is crucial to analyse each database system used. In this case, the idea was to have two examples of distinct ERP's databases and analyse both results to comprehend the usefulness of a solution such as the one purposed for each scenario. Nevertheless, both cases here presented are believed to be good examples of information system that could benefit from having their data classified. Therefore, it is going to be presented each information system.

5.2.1 *A University Department Application Scenario*

In this scenario, the ERP software is serving a department of a university and all its activities. The department manages payments, employees, students, teachers, administrative processes, stocks of essential products, and other entities for support the its activities. The amount of data in this scenario is considerable for some of the tables and the analysis of the classifications could be used to unlock new and valuable insights for manage departmental information. The database is serving the department for a long time and the amount of data has started to generate some performance issues due to some demanding queries and the weak infrastructures that are being used to run the software. This way, it would be very interesting to improve the performance by disposing historical and unused data from the database in order to save resources for not having to upgrade departmental systems where the ERP is installed. Some of the disposable information could come from previous student records, purchases and payments, old processes and their assets and so on. Most of this information could be erased or simply moved to other databases that are used as historical repositories for avoiding the occupation of the main computational platform. Hence, having a solution that is able to identify that potential information is ideal for the task. In addition, the relevance classification of the information, for example, about the payment related entities and from other crucial operations of the department are quite valuable insights. For instance, by identifying what are the least relevant purchases that were made, it should be possible to identify useless items that are being bought and avoid wastes. Through the analysis of the results from the application of the proposed approach, other kind of wrong investments and flaws alike could also be found. Moreover, many other applications for the results can be discovered for improving departmental data and operation management, depending on its needs.

In conclusion, this is an eligible scenario that could benefit from having its data classified, in order to apply new and more specific governance measures for improving information system's management and comply with user demands. For the purpose of this project, this case is going to be used in a comparison between another different example for evaluating the usefulness and correctness of the results of the solution.

5.2.2 *A Retail Company Application Scenario*

In this application case, the ERP system is serving a retail company that has numerous sale points, having each one a client and a database to support their activity. This company is responsible for a great number of sales and purchases, which generate a huge amount of transactions that are stored in the ERP's database. Besides the sales and purchases, the

ERP has to handle other internal operational activities related with customers, products, employees' management, and others activities. This way, it is clear that the information system of each sales point is going to have to deal with vast amounts of data and still answer to the business demands.

The performance factor is crucial in this scenario, since each store cannot afford to have its system stalled during working time, and with the amount of data that some might have to handle, this is a problem that may be a reality. To solve this problem, a solution could be designed using the identification of irrelevant pieces of data that could be disposed for gaining performance. Some databases might be functioning since the store's opening and from then a large amount of data was potentially created. Some of that data might not offer any interest to be kept stored. Thus, it could be identified in order to adequate the most appropriate data management measure for handling it, which could be in this case making its removal or dislocation from the system's main platform, as suggested before. This could offer some performance improvements just from the identification of what is important in the information system.

In addition, the data from each sales point is also being use for data analysis purposes. Therefore, is also crucial to have quality in the data to be analysed in order to provide meaningful results. The elimination of noise in data, such as historical and unused information, or simply irrelevant data is fundamental for ensuring the quality of the data analysis procedures. Again, by using the relevance classification scores from each table instance, it could be possible to identify the noisy information. For this purpose, the use of the process we proposed could significantly improve the quality of what is analysed and save the company from making business decisions based on the results of an analysis with corrupted data. The analysis of the classified data in terms of relevance is also a very interesting perspective, which could be explored for further analysis for discovering new conclusions about data that otherwise could not be achievable. These are worthy insights for business decisions that are valued by the company managers. Hence, it confers an even higher importance for a solution such as the one proposed. Given the information system struggles, the presented case study is also an eligible scenario of larger scale. Therefore, the comparison of the results from each case it is an interesting analysis for evaluating the procedure.

5.3 TEST SPECIFICATION AND VALIDATION

5.3.1 *Tests' Approach and Specification*

The testing strategy thought to carry out the evaluation of the prototype was quite simple, but precise and enlightening. However, there are some fundamental aspects that need to be evaluated in this procedure. For instance, the time that has to be spent to complete the task, the machine learning methods' accuracy and precision, the usefulness of the results, and so forth. In order to achieve the conclusions, it was defined a set of tests that were executed for each presented case study.

As for the tests performed, the strategy for each scenario was to conduct a general execution over the entire database, and another more specific test for analysing the effect of the process in the most impactful table of each information system. The main purpose of the general simulation was to measure the total processing time for the execution, and have a broad idea of the feasibility of the approach in that matter. On the other hand, the objective for the singular table tests was to gather specific performance and accuracy results about an important system's entity, which is used to generalize the idea for the rest of system tables. In both cases, the percentage of data classified with a new meaning is also going to be analysed, since it is relevant to determine the potential usefulness of the procedure. This way, it is believed that it is possible to validate the procedure with an approach that analyses the process on a generalist and specific perspective. This approach is believed to be quite enlightening to imagine a clearer picture of what results could be in other systems. The implemented user behaviour detection system was an experiment that had to be evaluated as well. This time, the purpose for the evaluation is to compare the results of the procedure with the auxiliary system and study its impact and viability for other solutions based in the proposed process. Finally, for comparing the implemented machine learning algorithms, it was conducted a final evaluation. This last test is a single table test run in which both of the algorithms make the classification of the same dataset. The only thing that changes is the data preparation process. The changes in the pre-processing phase are intended to adequate the most appropriate data for each algorithm, since each one required a specific preparation. Through this evaluation, it is possible to compare both algorithms in terms of performance, and acknowledge the real differences between both implementations.

The single table tests were repeated three times for each scenario. With this, it was possible to perform an average of the measured results in the analysis phase. The rules and preferences were defined with the support of an analytical software. Having this kind of programs to assist this process is extremely valuable to unveil important insights about

data to define the most appropriate rules and preferences with ease and efficiency. For instance, determining the most frequent and old values is far easier to obtain using analytical software. Moreover, creating visual representations of data to assist the analysis and consequent preference definition is also another versatile feature that facilitates the process. The preferences and rules were different in each run of the test, and since they were defined a priori, we did not use the recommendation system or the user behaviour detection process. The amount of operational logs is related to the available number of records from each table in the total captured in each system. Finally, the reason behind the decision of the table to be analysed stands on the fact that it was found to be the most frequent in the operational logs, and it is one of the biggest in dimension for both scenarios. Making it a perfect candidate to be thoroughly evaluated in this test. In Table 3, it is presented the specifications of the tests for single table evaluation.

	University Department	**Retail Company**
Table	dbo.Movimentos	dbo.Movimentos
No. of attributes	91	91
No. of instances	1,071,727	363,002
No. of operational logs	102,607	93,262
No. of test executions	3	3
No. of preferences	3	3
No. of rules	2	2

Table 3.: The specifications of the single tests for each scenario.

The strategy defined for performing the general system execution was similar to the one for single table tests, though it had to suffer some changes due to some problems encountered. As for the number of tables analysed by the procedure, it was only considered to be valid the ones that had operational logs registered. This way, it is ensured that only the tables that were used are selected. The Hot Table System performs the selection process, and it has certain criteria that need to be defined. The first one to be defined is the minimum usage factor, 5% of the total logs, and its weight, 60% of the total assigned to define the colour for each table. As for the space criteria, the minimum and maximum space factors weight the same (20% each) in the colour definition decision and their percentages were defined to be of 5% and 70%, respectively. From this process, were select 20 (15%) tables from the University department' information system and 19 (95%) from the retail company's one. These criteria values were equal for both scenarios for ease of analysis afterwards. For this test in particular, the operational logs did not have data for every table in the system, but it was possible to gather enough information for reflecting a precise system usage. Preferences and rules were generated by a utility based on the most frequent attributes and values of each table. This test was executed three times for each scenario,

according to the same reasons presented for the single table tests. In Table 4, it is possible to observe the details of the specifications for each case.

	University Department	Retail Company
Total no. of distinct tables with logs	133	20
No. of selected tables	20	19
Average no. of instances per selected table	24,326	57,553
Average no. of operational logs per table	13,007	76,331
Total no. of operational logs	1,729,906	1,526,636

Table 4.: General System Execution Tests' Specifications For Each Scenario.

As mentioned before, the user behaviour detection auxiliary system is a perfect example of a tool that could be created for supporting the scoring systems. The one here presented has the same specifications as the general execution test of the university department system. The database and operational logs used were the same as well. The only thing that changed in this evaluation was the auxiliary systems. This time, the second scoring phase was only composed by the user behaviour detection system. To create behaviours it was required to develop another generator for simulating the usage of the application. These were based on the same principles as the preference and rule generation, which is to favour the most frequent attributes and values. Through the analysis of the results, it is possible to study the impact of the proposed system and compare it with the other approach.

The fourth and final test was dedicated to the Deep Learning and Naïve Bayes algorithms comparison. The deep learner was implement to cope with the main algorithm's flaws, hence, it is seen as more powerful already. However, the purpose of this evaluation is to compare the actual difference between both of them and study the performance of the deep learning algorithm in particular. As for the actual specification of the test, it is the same as the single table test for the university department. The only change is the data preparation of the training set, which was adapted to fit each algorithm. Another important issue was the machine where the tests were performed. The computer we used was a simple laptop, with modest components. Nevertheless, it was able to serve the demands we imposed for conducting the tests. The computer we used has the following specifications:

- **Model:** HP EliteBook 8440p
- **CPU:** Intel Core i5 520M / 2.4 GHz
- **Max Turbo Speed:** 2.93 GHz
- **Number of Cores:** 2 Cores
- **Memory:** 8GB DDR3 SDRAM

- **Memory Speed:** 1333 MHz
- **Hard Drive:** 250GB HDD
- **Hard Drive Spindle Speed:** 7200 rpm

With the defined testing strategy, it is possible to conduct an evaluation that is able to analyse the feasibility of the created procedure and the developed prototype's viability in various perspectives. These are crucial for validating the project developed.

5.3.2 *Test Validation*

To evaluate the procedure, it is mandatory to analyse the correctness and usefulness of the results before using them. Therefore, the produced data is evaluated in various perspectives related to those matters. One example in particular, is the time that it takes for executing the procedure, since for some scenarios it may be required to finish its execution as quickly as possible to avoid the database occupation. We need also to evaluate the machine learning process correctness. In this validation, the models are analysed through a validation process for determining their viability in the classification of the testing datasets, as referred in a previous chapter. Another relevant aspect defined for evaluating the procedure, is the amount of data that it may be able to differentiate with a new meaning. The reason behind it is that every piece of data is defined neutral before the execution, which is with the medium level of relevance. Therefore, for each instance classified with a different level, it is considered to have a new meaning. This way, it is possible to think of new solutions for handling the differentiated data, think of the usefulness of the procedure, and go beyond the correctness and performance perspectives.

The results from the classifications may not always be correct, since the machine learning methods are not entirely precise and may have an error rate associated with the knowledge that was induced in the process and in data preparation. Hence, it was crucial to evaluate the models that were built to classify the information. Through that procedure, it is possible to confer an even greater degree of quality to the results. In specific, the constructed models were evaluated by a cross-validation method, which is ideal for this kind of algorithms. For the presented tests, it was defined that the minimum average percentage of correct classifications has to be of at least 70% of the total instances of the validation set. Any of the models that fail to achieve that percentage, are then submitted to the deep learner algorithm for improving their accuracy. This percentage is believed to be a reasonable limit for the error associated with the resultant classifications.

The tests mentioned above, are appropriated for evaluating the correctness and perfor-

mance of the machine learning process that provides the desired classifications above every other perspective. However, and as introduced before, there is another important aspect to be evaluated: the usefulness of the results. This is a quite subjective evaluation, in the way that it has to be the database consumer the one that evaluates the results in a practical point-of-view. This means that the user must find the results somehow useful for solving data management problems occurring in the system. Since this evaluation has to be made in long-term and has plenty of barriers in terms of legal and availability issues, it was not possible to conduct that investigation in this project. Nonetheless, it is viable to study an eventual usefulness of the results for the given scenarios. This way, results are analysed with a hypothetical purpose for them. For example, by removing or moving the least and most important information to different auxiliary structure, it is going to improve system performance. Therefore, the percentage of data classified as such, it is a good performance measure for evaluating the usefulness of the solution. Some tables may not have any data eligible to be moved or removed and thus, the solution did not bring anything new for them and it was not really useful in that particular case and perspective. On the other hand, if the procedure was able to identify plenty of instances with a new meaning, either as relevant or not, then it is useful in terms data management.

Regarding the data removal hypothesis, it is emphasized that it will have to be thoroughly evaluated in each case, because there are records that are functionally given as little relevance. However, in a long run, they represent a lot of value for the company. As such, the removal of records is merely figurative of what could be saved in terms of space in a database. In addition, in most cases what would happen was the decentralization of records eligible for auxiliary structures outside the main system in order to not compromise the integrity and utility of the data. Fundamentally, the key is to make use of the new meaning that was possible to confer over data and adequate appropriate management measures.

There are other ways for evaluating result usefulness, such as through its analysis. For instance, if by analysing the classified information is possible to obtain new conclusions about data quality, data relevance, management and operational insights, or any other kind of important remarks, then results have utility and are meaningful to the user. Another application of the results could be the creation of user customized analytical dashboards for analysing data in terms of relevance and other useful aspects that serve both management and business purposes. This way, having support of visual elements the analysis is simplified and improved. Finally, it is believed after this work that it is possible to discover new ways for optimizing systems, unlock new data quality insights that were not possible before this process, and create a new proactive approach to deal with data management problems.

5.4 RESULT ANALYSIS

5.4.1 *The University Department Scenario*

The case of the university department was the first to be evaluated. Therefore, as for the general execution test, the prepared operational logs have more than 1,700,000 entries. From that total was possible to distinguish 133 different tables. The number of records reveals a small volume of system usage data due to the portion of tables that was possible to identify. This reveals that there is not enough information on most of the structures of the system in these logs. However, this is not a problem, because the remaining tables are not used. Thus, they are not relevant for analysis. From the available operational records, it was possible to build a structure with the queried tables for each user session, in order to create the corresponding Markov chain. After the colouring phase of the chain it is possible to select the most adequate tables to be processed. From the resulting selection, about 15% (20) of the tables with records were selected, which greatly reduced the analysis process. After the selection, each table was processed by the machine learning engine of the prototype to achieve the actual predictions.

Once the procedure was finished, the average time measured was approximately 2 hours and 3 minutes, which is an average of 6.15 minutes for each selected table. The average number of instances distinguished with a new meaning was of 74%, which means that it is possible to create new measures for a huge chunk of the total data. The instances classified as non-relevant is of 9.43% on average. This last percentage represents a more realistic portion of the data that could be removed, since it is not relevant. These results are quite satisfying given the fact that the preferences and rules were generated. Besides, the broad business understanding of the actual information system did not provide enough knowledge to adequate specific user preferences into the auxiliary systems. Nevertheless, the approach followed was effective to overcome the problem. The average accuracy of the model measured was of about 83% from the cross-validation evaluation, which is a very acceptable result, since the data preparation was based on the generic approach defended. The execution time is believed also to be good, since the database has a dimension of about 4 GB of data to be classified, and the average of logs per table is 12,782 entries for preparing and training the models. One of the reasons why the duration was not very long is because the deep learning algorithm was only requested 10% (2) of the times, which avoided the heavy processing most of the times. To better understand the machine learning correctness and performance, the next analysis was intended to cover those matters in detail. In Figures 14 and 15, it is represented, respectively, a chart with the average time and a chart with the

average data classified with a new meaning for the three test runs in which the database was submitted.

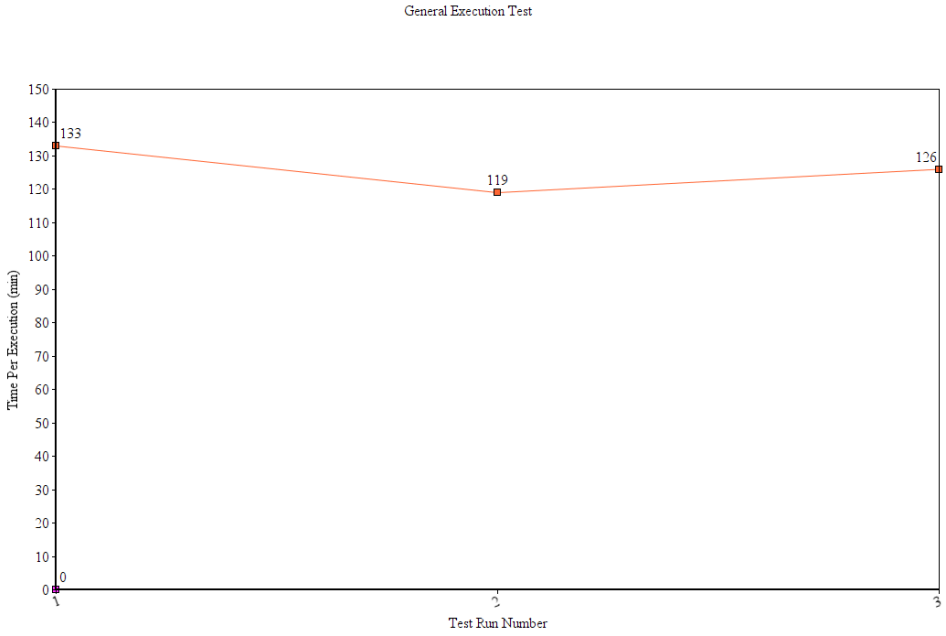


Figure 14.: General Execution Tests: Average Time.

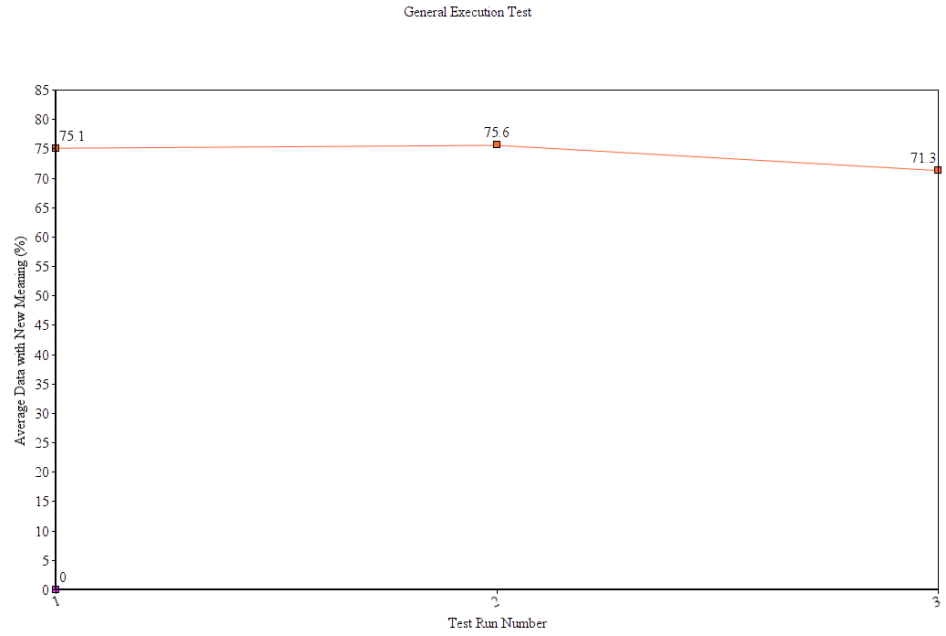


Figure 15.: General Execution Tests: Average Data with New Meaning.

Analysing the results of the single table tests, we see that this table has a total of 1,071,727 instances to be classified and occupies a total space of 629MB. To create the respective training dataset, there was a total of 102,607 operational logs registered. To build the model, it was only necessary to process the training set through the Naïve Bayes algorithm, given its high precision and accuracy results. This was quite profitable in terms of time to process the table, since the heavy processing of the deep learner was avoided in every test run. As for the duration of the execution, was measured an average of 34 minutes and 22 seconds for the three total test that were performed. As for the accuracy of the predictions, it was on an average of 85.64%. This value was obtained from the analysis of the confusion-matrix, which may be generated from the cross-validation process results (Stehman, 1997). This matrix represents the number of correct and incorrect classifications for each level of relevance. Analysing the percentage of space that could be saved, it is concluded that it is possible to save about 11.28% on average with the decentralization of the irrelevant data of the main system. This is a very satisfying result, since about 10% of the total information that was analysed is given as little relevant and could be removed. Besides, 10% is equivalent to 62.9Mb of space that could be saved with this small example. Having this percentage in a much larger scale example, it is possible to foresee a major improvement in the performance of a system. In addition, this projection is just with the possible measure for removing the least relevant instances of each table. As for the total percentage of data classified with a new meaning, it is a little above the global average measured previously and is of approximately 84%. The reason that explains the slight improvement, in comparison with the general execution tests, is the preferences and rules definition, which were based in an analysis that helped to discover the most adequate attributes and values with analytical software. Lastly, the results differ from each run, because the preferences and rules were also varied in each test run. In Figures 16 and 17, it see, respectively, the evaluation results of the models as well as a chart with the percentages of eventual space savings from removing the least relevant data.

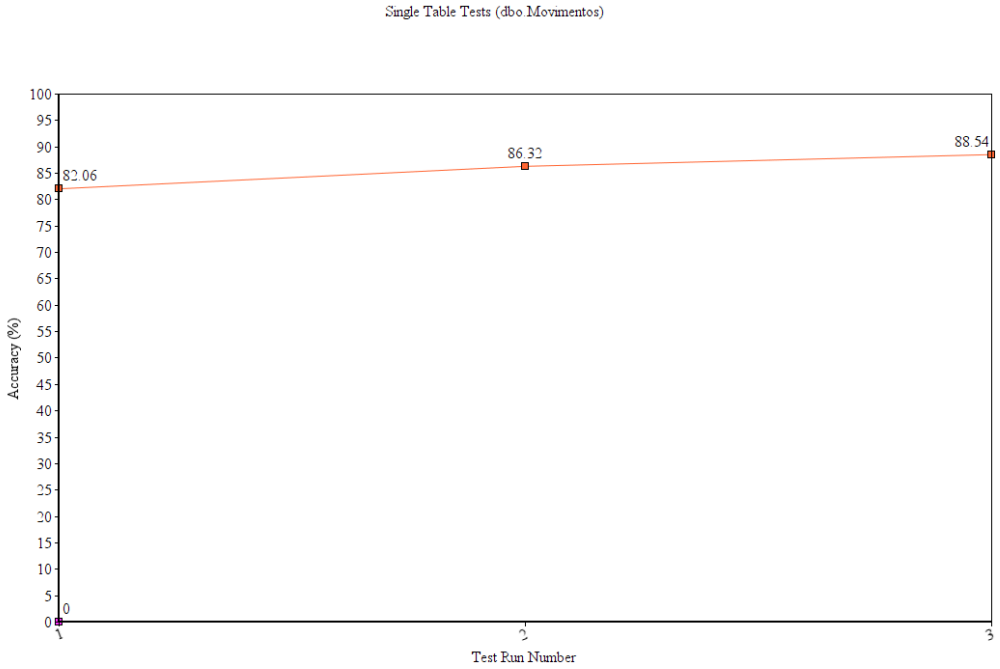


Figure 16.: Single Table Tests: Average Accuracy.

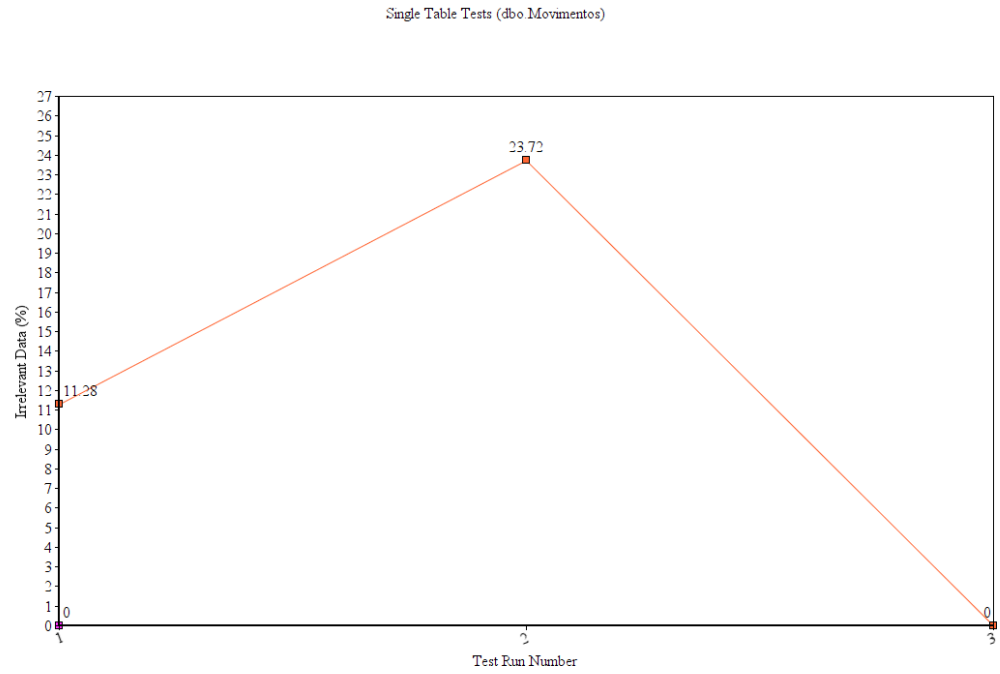


Figure 17.: Single Table Tests: Average Irrelevant Data.

Having this analysis, it is possible to conclude that the procedure is efficient in terms of performance and in this perspective could be used in the company's environment. Analysing the average duration of both tests, it is possible to conclude that the time spent on creating the results was not long, given the machine used to perform the tests and the complexity of the procedure. Moreover, if the results produced are actually useful to solve the company's problems, the time spent on achieving them is inexpensive. As for the correctness, the results revealed that the accuracy of the predictions is quite high, which confers a trustworthy level of confidence of the produced results. The evaluation that the models were submitted is believed to be adequate to support the decision on whether the classifications are worth to be trusted or not. As for the actual classifications, the results are very satisfying because it was possible to define a new meaning to a great majority of the instances of the database. Especially, in the evaluated table, it is possible to observe a very interesting result, which can be used to generalise the idea for the rest of the tables.

With this much data classified, it is possible to think of new management measures to solve the department's problems. As an example, in the single table test, it is possible to identify about 10% of the analysed data as little relevant, which could be removed or moved to save some of the storage space in the main platform. This example, is just one in a sea of options that one must be able to analyse and identify the most adequate to solve its problems. At the end, from the overall analysis we presented, it is possible to affirm that the university department could benefit from having its data classified, in order to improve its data management tasks. Nevertheless, to enforce the validation the process in this case, it is required a long-term evaluation of the evolution of the system with the support of the department.

5.4.2 *The Retail Company Scenario*

The scenario of the retail company was interesting for comparing its results with the previously case study. The database involved with, as reviewed before, is a kind of a small snapshot of the information held in one of the company store's information system. Nevertheless, the dimension of the data is quite similar to the scenario of the university department. Therefore, it is still a good example of a different business activity and its consequent demands, which provides different conclusions. In this case, the time of general execution of the test on average for the three runs was 4 hours, 10 minutes and 12 second, which represents an average of about 13 minutes for each table to be processed. This result was quite satisfying and at the same time expected, since the average number of operational records for each table (76,331) and the average of instances per table (57,553) were higher than in the previous scenario. This leads to an increased demand of the machine learning

engine, which leads to a lower performance when classifying data. Nevertheless, the duration measured is believed to be acceptable and proportional in relation with the other scenario, given the increased effort to process this database. The deep learner algorithm was not required in this case, which favoured the performance registered. The database only had 20 distinct tables in the operational logs, but the percentage of tables selected was much higher than in the previous case, which was of 95% (19) of the tables. This reveals that those tables represent the critical data for this scenario and a great majority of it was processed by the prototype. This means that most of the available data was analysed and there is the possibility to create new management measures and gather insightful knowledge, practically, about the entire database.

As for the classifications results and evaluation of the general execution tests, the average correctness accuracy for the predictions measured was of 97%, using cross-validation. This confirms the efficiency of the Naïve Bayes algorithm, since the alternative was never used in this case. Besides, this excellent result also highlights the efficient data preparation arranged to construct the datasets that trained the models. Ultimately, from this analysis it is also possible to conclude, that the evaluation conducted confers an elevate grade of confidence of the results created. From the classification analysis, it was possible to unveil that about 51% of the total data analysed was given a new meaning. This result is quite lower than the one registered in the previous case. Nonetheless, it still reveals a good amount of data that could be treated to improve the system performance. Besides, the percentage of data classified as little relevant was of about 27%. This is quite good, because the amount of data that could possibly be removed is huge and with the accuracy measured in the models evaluation, this possibility is even more realistic. This result is explained with the amount of historical data that is held in the database in question, for example, from old purchases and sales records. Another factor that could improve these results is the customised user preference for each table, which in this test were simulated based on the same principles mentioned in the previous scenario analysis. Figures 18 and 19 present two charts representing, respectively, the average data classified with a new meaning as well as the average time for the three total executions of this test.

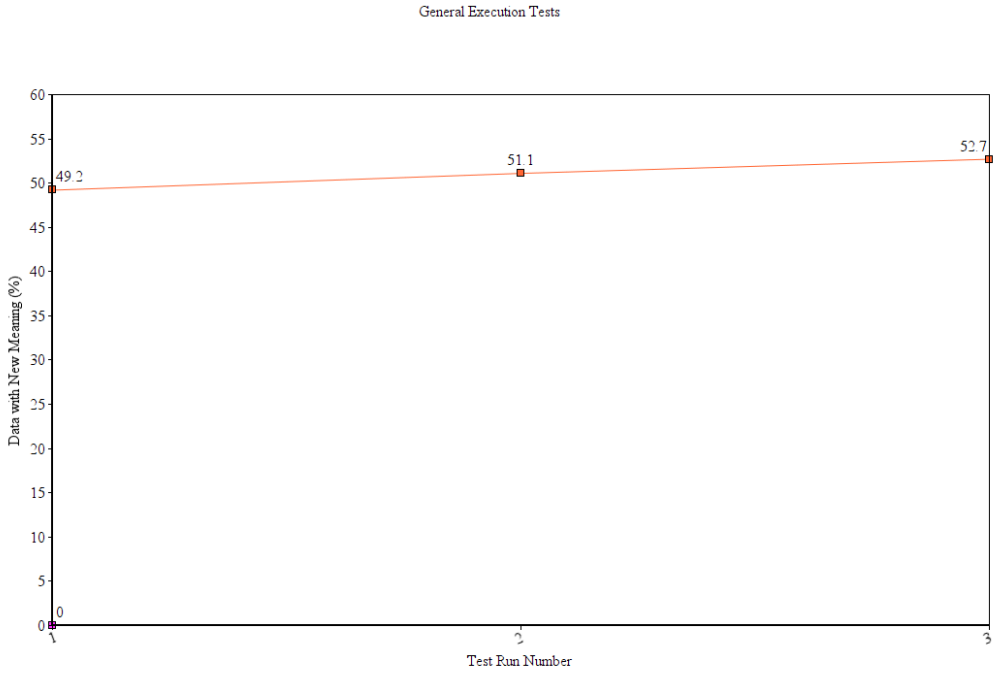


Figure 18.: General Execution Tests: Average Data with New Meaning.

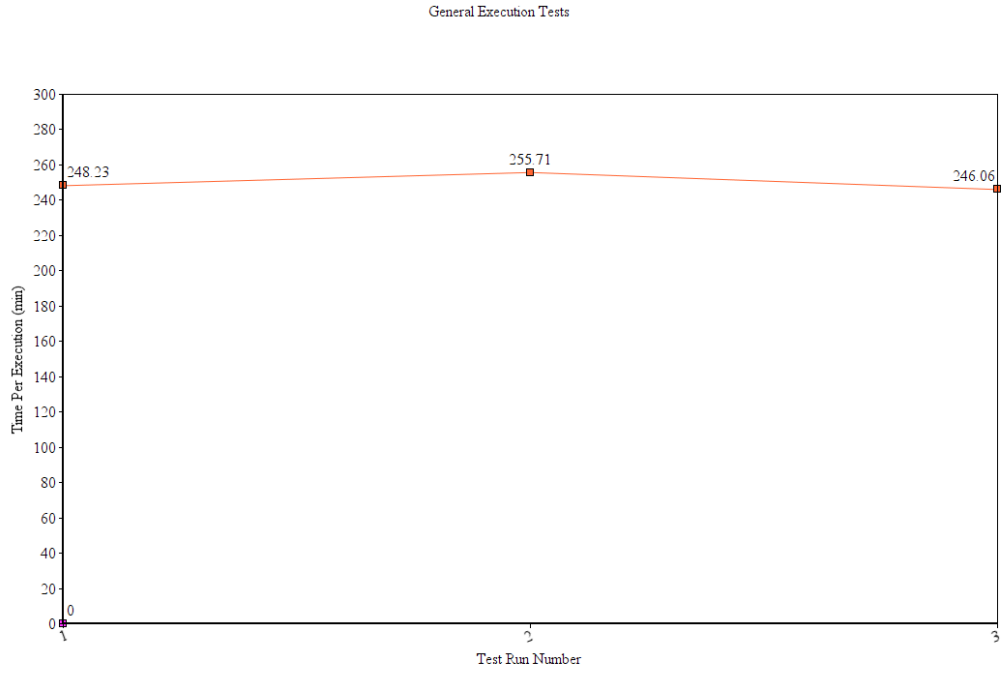


Figure 19.: General Execution Tests: Average Time.

In the single table tests, the evaluated table has 93.252 records of available operational logs and the total of instances to be classified was 363,002. This table is quite smaller in comparison with the same example in the other scenario, but the amount of records for preparing the train of the predictive model is similar. Nevertheless, this table was still found to be the most important and adequate for this evaluation. As for the total execution time was on average 24 minutes and 21 seconds, which is under the duration measured in the previous case due to the fewer instances that had to be classified in this test. However, the result is quite satisfactory, since it is the most important table of the system in analysis.

The classification results were quite impressive in this case, following the previous scenario example. The average accuracy for the model built by the Naïve Bayes classifier is of 88%, which confirms the good performance of the machine learning engine and confers the quality required for the predictions created. In this test, this result in particular is quite more relevant than for the general execution evaluation, since the user preferences were carefully chosen and this reflects a more realistic approach to take on processing each table in this approach. The average data classified with a new meaning was of 89%, which was way above the average registered in the general test execution. This is explained with the improved criteria induced in the knowledge that trained the models. Moreover, 41% of the total information was classified as little relevant and could possibly be removed. This value is also above the general average measured before, but again, it is explained by the user preferences defined, which helped to refine and differentiate data with more criteria.

The usefulness of the results provided is clear and so is the correctness factor. It is possible to adequate new governance measures to most of the data and from the preferences defined, almost 50% of the data could be removed. This evaluation was of a single table, nonetheless, it is possible to generalise the idea of the efficiency improvement from having strong preferences and plenty of operational logs to train the predictive model. This way, it is believed that if the process is well applied, like in this case, it is possible to achieve the so wanted improvement of system data management tasks. Next, in Figures 20 and 21, it is possible to observe two charts representing the average percentage of data found to be irrelevant as well as with a new meaning for the three executions of this tests, respectively.

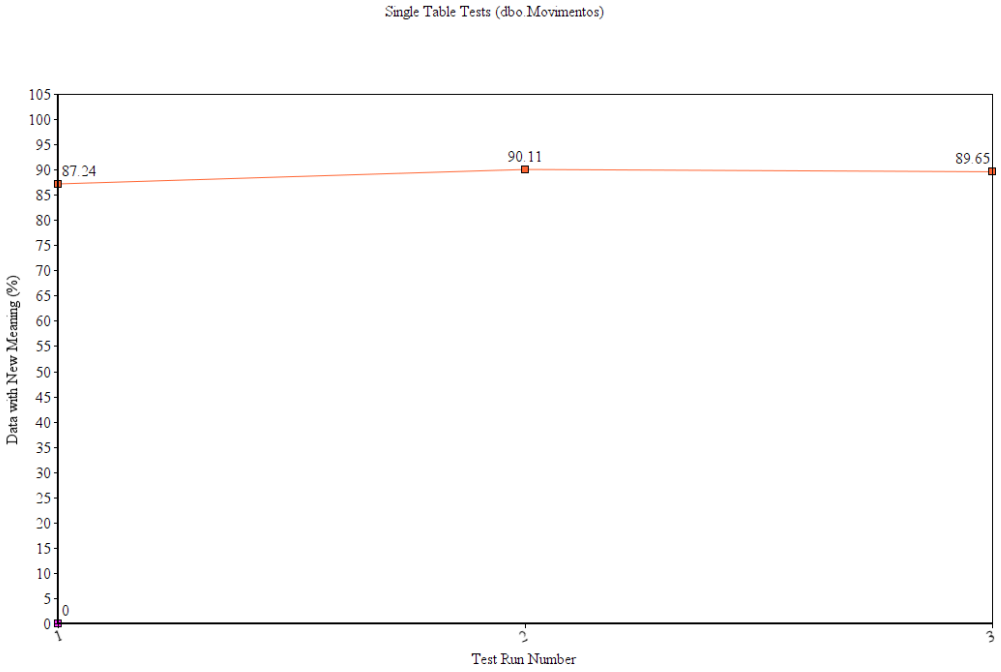


Figure 20.: Single Table Tests: Average Data with New Meaning.

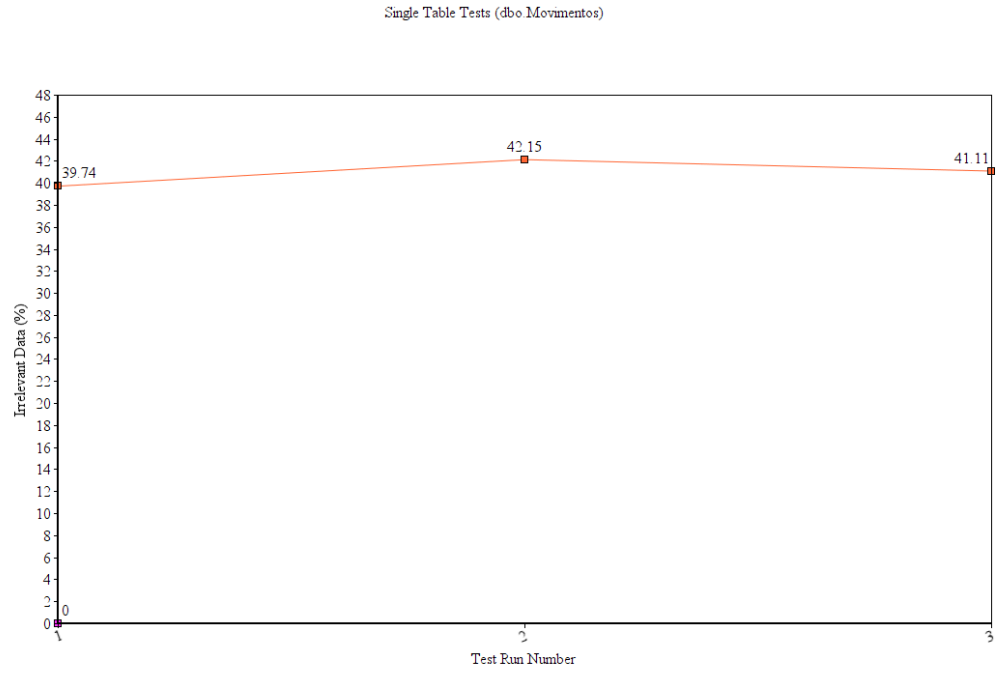


Figure 21.: Single Table Tests: Average Irrelevant Data.

To conclude, the tests conducted in this case were relevant analyse and compare the process's application in different information system with different business activities. Moreover, the quality of the results is very similar in both scenarios, which enlightens the feasibility of having a solution such as the one proposed to support the data management tasks in the future.

5.4.3 *Deep Learning and Naïve Bayes Analysis*

In this test, the main purpose was to compare the performance of both algorithms in the same example. This time, the table tested was the same as the single test evaluation for the university department case. This example is quite complex and has enough information to conduct a good comparison. The specifications were the same as the previous presented test, and, therefore, this analysis highlights the results of the deep learner algorithm.

In the time perspective, the deep learning algorithm is tremendously more expensive than the Naïve Bayes. The average duration of each test execution is 6 hours, 40 minutes and 48 seconds, which represents a drastic difference between the performance of the two algorithms in this matter. The duration of the deep learner is more than ten times higher than the one measured for the other classifier. This is explained with the way each algorithm processes data, and, in particular, the deep learner performs very complex operations that require many resources from the physic machine. In this case, the computer used to run the tests was not the most adequate to perform well on constructing the deep learning model. Besides, the implementation used has mechanisms of parallel-distributed network training implemented and other optimizations. However, these would only improve the performance significantly if the machine used could make full use of them.

The accuracy and the percentage of data classified with a new meaning are also relevant matters to be discussed. This way, the average accuracy measured by the cross-validation evaluation was of 96.31%, which represents a great result that demonstrates the amazing power of this classifier and its implementation features. This value quite a bit above 10% higher than the results measured in the Naïve Bayes application case. This enforces the confirmation that the deep learning is superior in this matter. As for the percentage of data classified with a new meaning, the difference is not so notorious. The average value measured for this algorithm was of 88.34%, which is only about 4% higher than the previous one. This indicates that both algorithms are able to provide satisfactory results due to the efficient user preferences defined, which are able to confer more criteria to the data used to train the models. Thus, from this analysis, it is possible to conclude that both algorithms are able to provide the desired results with efficiency. However, the deep learner is notori-

ously more expensive in terms of performance than its rival is. Besides, the gain achieved from the heavy processing may not be worth to pursue for some cases at the expense of ten times more time. The deep learner is an excellent option to confer the best possible results one can have with the implementation of our proposal. Having more than 95% chance of having a correct prediction is entirely desirable and may be required to be this high most of the times. Nonetheless, it is advisable to pursue this option, only when required due to the abrupt difference in performance, which may also be at stake. Finally, it is important to emphasise that the computer used in the tests was not the most suited for this task. This greatly affected the results. The following two figures (Figure 22 and Figure 23) present the average accuracy measured and the average data found with a new meaning for the total executions of this test.

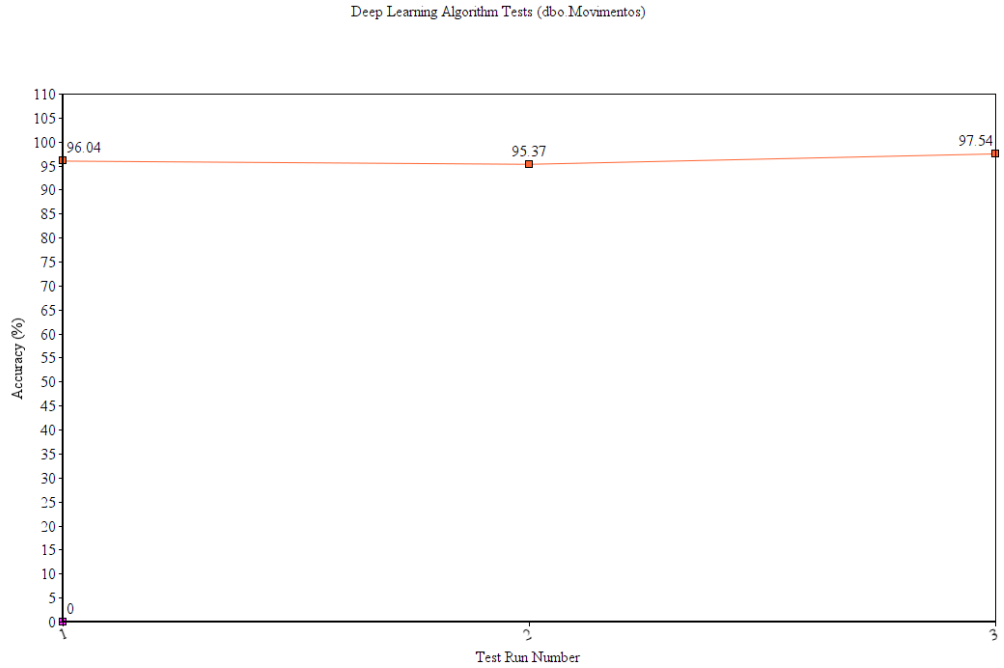


Figure 22.: Deep Learning Algorithm Tests: Average Accuracy.

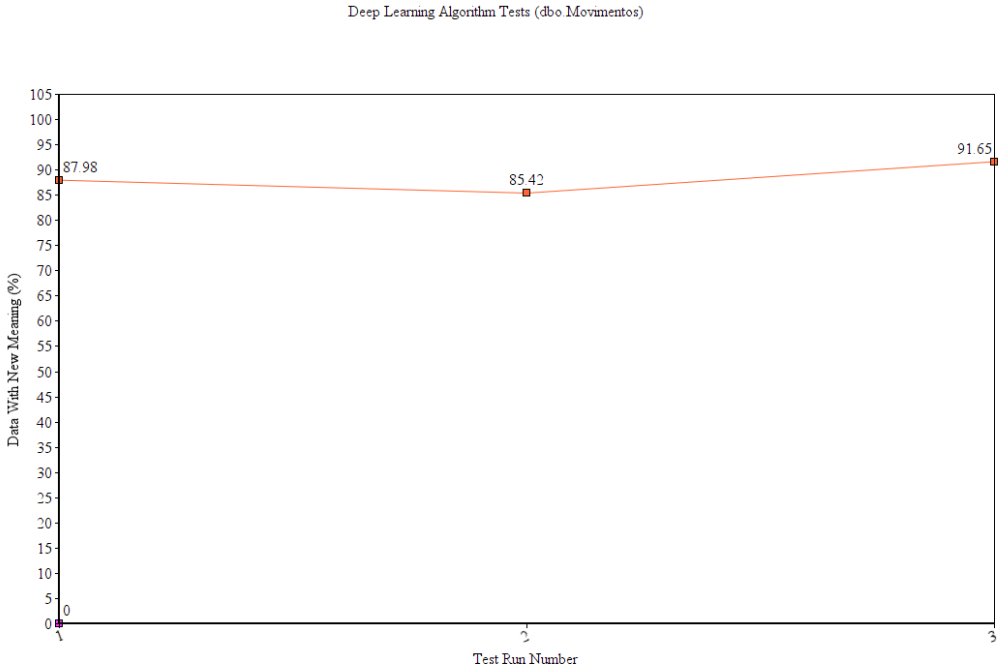


Figure 23.: Deep Learning Algorithm Tests: Average Data With New Meaning.

5.4.4 *User Behaviour Detection System Analysis*

In this evaluation, the objective was to conduct a comparison of the general execution tests for the university department against the same test specification, but with the user behaviour system as the only auxiliary scoring system. This way, the analysis will focus on the same perspectives as the previously conducted before. As for the results, the average execution time was quite similar to analogue general execution tests, since the average duration measured was of 2 hours and 23 minutes. The duration equivalence between this test and the general execution is explained by the fact that there was only one modification between the two test specifications, which were the auxiliary scoring systems. The processing time from of scoring system is quite low in comparison with the machine learning engine. Therefore, the performance was not affected significantly. With that comes another interesting fact, which is the good performance of the implemented scoring systems. Another factor that influenced the total duration was the request of the deep learner one more time (3 times on average) than in the general test execution.

The average accuracy of the models measured by cross-validation evaluation was of 87%, which reveals the same efficiency as the one measured before. This also indicates that the implemented auxiliary systems did not affect the machine learning methods performance

on building robust models. However, the total percentage of data that was classified with a new meaning decreased a little. This time, the average value measured was of 61%, which reveals that the knowledge induced by the behaviours was not decisive enough. The reason for this was that behaviours were simulated and did not provide conclusive user preferences to refine the training sets that trained the models. To improve these results, it is required to capture real behaviours for reflecting real usage knowledge and increase the user preferences by combining this auxiliary system with the others. Nevertheless, the results achieved are quite enlightening to realise the feasibility of the system and its capability for improving the knowledge gathering that supports the defended approach and the prototype in this case. As for the duration of the procedure and average accuracy of the model, they demonstrate that the designed architecture for the process for building the predictive models is efficient and light in terms of performance.

Finally, and to conclude, it is believed that this system, in particular, is a very interesting and efficient method that may provide the most accurate knowledge one can have to implement this procedure, along with the operational logs. In other cases, this system should be implemented and well thought to provide insightful data that can be used to classify data, as should the logging systems. In the end, with these systems implemented and well arranged, it is possible to gather valuable knowledge to fuel this process. Figures 24 and 25 present two charts representing the execution time and the average data, respectively, with new meaning found in the three total executions of this test.

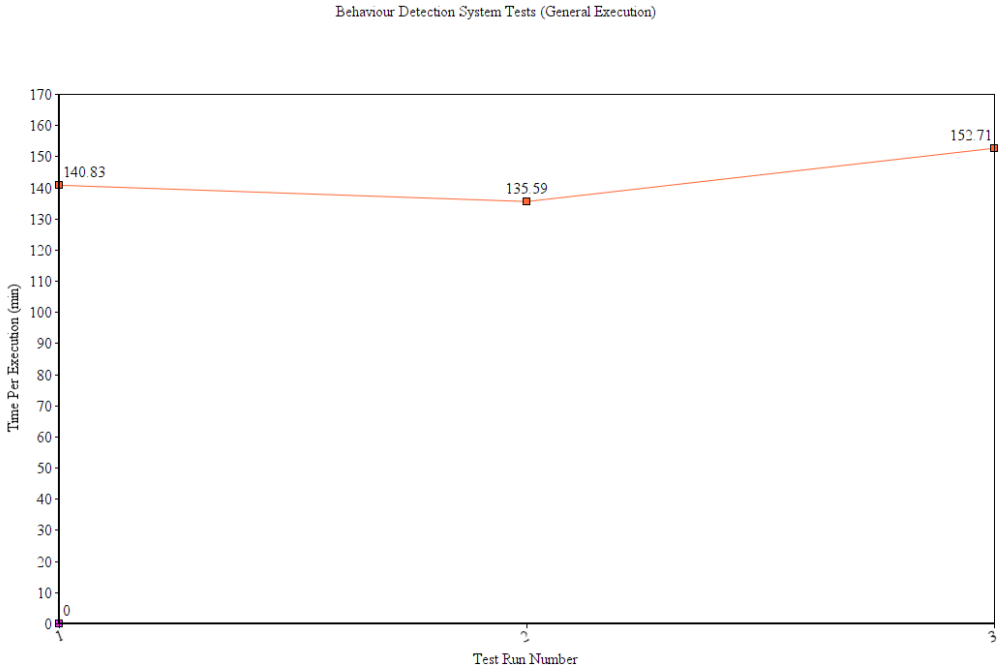


Figure 24.: Behaviour Detection System Tests: Execution Time.

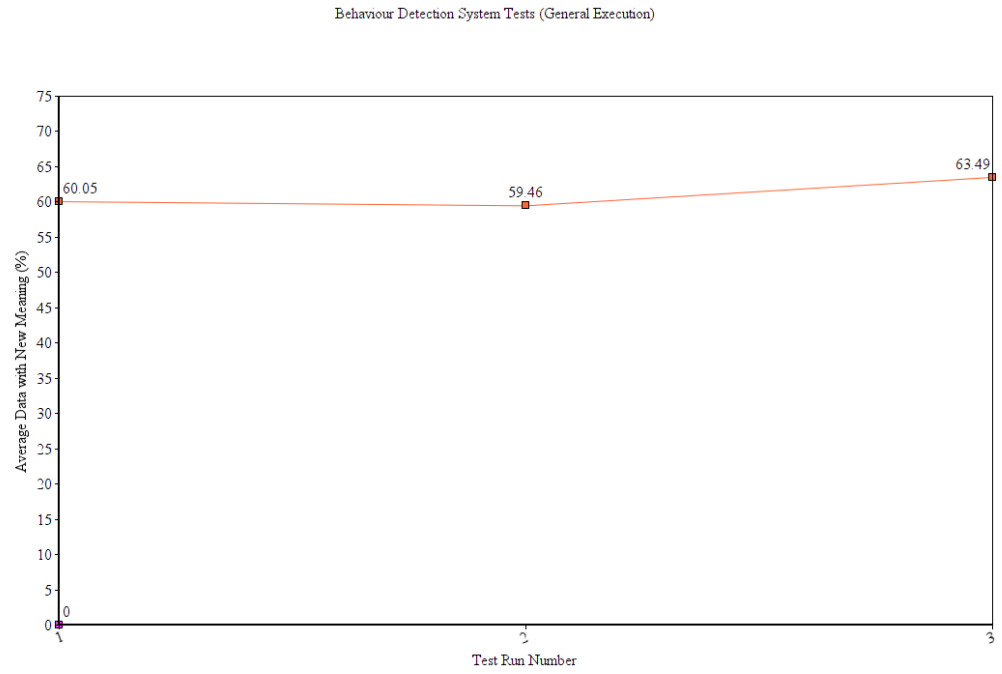


Figure 25.: Behaviour Detection System Tests: Average Data with New Meaning.

6

CONCLUSIONS AND FUTURE WORK

Now, having the work developed, it is believed that the process that is being defended in this work can improve data management tasks in any conventional database scenario. Given the current state of the data management in big databases, it is required to discover new ways for handling data and improve the governance measures for each system. A solution, as proposed here, is seen as a valuable approach due to the important insights that it provides, in order to create and adapt management measures for implementing in a system. From the conclusions that were able to be drawn throughout this research, it is believed that having data classified according to relevance of its system users is an excellent way for discovering new insights about data, which can be used to refine the data management of an information system. This overview is a proper way to be accurate in determining the quality of the data of a particular system. This work was led by the theory that what matters to the system users is what is important in that particular system. This way, it is also believed that discovering the meaningful data to a user, implicitly is being defined what data quality is in a given system. Therefore, it is safe to say that the created prototype, which applies the proposed data management process may serve as an example of a great tool for supporting data management in most information systems.

The study of the impact of the application of this data management process in a long-term analysis represents a future work that has to be conducted. Understanding the impact in data quality and management is a valuable validation of the work we presented and discussed in this work. Unfortunately, this was not possible to achieve since it is a very demanding task in terms of time for gathering the information for performing this analysis. Besides, it would also require an improved implementation of a solution alike the one presented and its maintenance, in order to secure its evolution and success. As for the data management suggestion presented in the case studies, which was to decentralize the least relevant information, it is meant to be an example of what could be achieved with the results achieved. The further exploration of these measures was not part of the presented work, since it depends on each system management requirements. Thus, this should be defined by system administrators that are in charge of the data infrastructures of a business.

In the case studies we presented, that factor was not possible to incorporate. Nevertheless, the important thought to be highlighted is the fact that it is possible to look at data in a distinct way and from the perspective of who uses the system for improving the existent measures and creating new ones.

This work was developed within a traineeship offered by PRIMAVERA B.S.S.. Therefore, the company had some objectives in mind with this work as well. So, the development of a prototype was quite relevant, because it was able to process an entire database and deliver meaningful results. At this point, the developed solution is just a mere example of what can be created for applying the proposed data management process. In this case, the development requirements of the prototype were established having the problems of the company's ERP system in mind, and it was also adapted to that software in particular. However, based on the results, it is possible to conclude that it is a viable and worthy option to be pursued in any database scenario. Nonetheless, for other cases, it is advisable to evaluate and determine the best approach to implement the referred process and develop an appropriate solution. As for the application of a solution based in the presented prototype in a real scenario, it is required to rebuild the prototype into a sort of modular application that the final user may customize every processing feature and create new modules through extensibility facilities. For instance, to improve data processing procedures or to implement new auxiliary systems, the user should be able to implement them with ease and customize the data's processing procedure.

The generic data processing is not the best approach to accomplish the optimal results. Thus, new ways to improve it also need to be explored. An alternative to the generic approach is by simply investing time to investigate the most adequate data processing procedures for every dataset and develop an automatism based on those conclusions. Exploring new ways for enhancing the approach of obtaining knowledge to train models is also quite relevant to explore in the future. Still, and related to the training of the model, it is also crucial to create re-training strategies, in order to cope with the continuous growth of the information used in the learning phase. Moreover, the initial scoring phase is achieved based on a quite naive approach and therefore, it is important to improve it. In order to do so, a specific logging system should be designed and created for this use only. It should be thought to reflect the most important information that can be related to the databases. An improved user behaviour detection system, for instance, could be an alternative to the database logs to gather the initial knowledge and should be considered. In the presented case study, the logging system used was the one already implemented in the ERP system and, therefore, the whole initial scoring approach was based in those logs. Nevertheless, the operational registry was able to provide enough information to achieve the desired re-

sults, as reviewed before.

Based on the interpretation of the results we got, it is possible to conclude that these are satisfactory and enlightening. Although the dimensions of the databases were not perfect examples of the biggest databases that represent the best candidates, it was possible to perform individual tests for a rather large and important table from both systems. The amount of operational records for those tables, in particular, was quite large, which gave an acceptable degree of reliability to the classification results. From those, the generalised idea is also possible to be created and think of the performance for other tables and even systems. As for the quality of the classifications, it is safe to say that it is very satisfying, since it was possible to confer a new meaning for a majority of the total data with a high degree of accuracy in both scenarios.

The two scenarios we presented here were quite useful for determining the impact of the application of the process in different activities, and that largely enriched the investigation. From the comparison of the results, it was possible to conclude that the prototype developed was able to perform with efficiency in both scenarios. This confirms the quality of the approach taken to implement the process in the evaluated systems. As for the test accomplished, these were useful and efficient for evaluating every perspective defined in the validation. Therefore, it is believed that through this evaluation, it was possible to strengthen the investigation made and proposal here presented.

The classification of the information allowed for the discovering of a performance improvement in the system that could be achieved through the application of a new management measure. As expected, observing the percentage of information that could be decentralized from the results got with the tests, it could offer a huge gain in space savings. Thus it represents a way of reducing the dimension of a system. Which often entails quite high costs, both in monetary terms as well as in terms of information processing. Given the previous fact, it can be affirmed that the main objective of this work was accomplished, since it is possible to create a new dynamism in the way the data systems are managed today with the proposed methodical process. Nevertheless, the solution presented here was only evaluated in a couple of real situations and with tests that only allow a view of the situation in the present. Which means, it is not possible to accurately predict the future impact of the application of this procedure in a system. From the previous conclusions, the sight is quite enlightening though it still may be deceiving for some cases. Therefore, it is important to enforce that in the future, this is a relevant matter to be further explored.

As for other applications of the results, it is probably the most exciting part of the whole

process due to the variety of possibilities. From the classified data sets, many practical measures can be taken for improving the management of an information system. These measures depend on each system requirements and may be immediate to be found. If datasets are analysed appropriately, then more specific measures can be prepared to manage that information. Reinforcing a previous statement, it is crucial that the data quality and management requirements are well defined, in order to adequate the best possible governance measures.

Throughout this work, there were some difficulties that had to be dealt with. However, there were also some major advantages that made things simpler. For instance, there is not a single approach similar to the one presented available for investigating and that initially indicated that it could fail. Nevertheless, there was plenty of information and case studies available regarding the machine learning subject and the data management scene, which definitively helped the investigation and decision making process. The information studied was valuable in the way it made solutions for the problems encountered possible and it helped to better understand the reality that the information systems are going through nowadays. Another adversity was the complexity of the project, both in terms of development infrastructure and difficulty to apply the whole process. To overcome these problems, the tools used to support the development of the prototype, such as the machine learning libraries and other data processing tools, played a huge role to assist the development. As for the infrastructures, the company provided the best equipment and support throughout the entire development. Besides those assistances, it is vital to mention the project coordinator and supervisor, who were also crucial in the decision making process, the business understanding and provided the most valuable knowledge throughout every phase of the project. Therefore, it is believed that in the end, the combination of all the presented factors, good and bad, carved the success of the proposed data management process. Nonetheless, the future path presented should be further explored, in order to improve the overall approach and correct the many possible mistakes that were not possible to foresee or emend.

BIBLIOGRAPHY

Agrawal, R., T. Imieliński, and A. Swami

1993. Mining Association Rules Between Sets of Items in Large Databases. *SIGMOD Rec.*, 22(2):207–216.

Ahmed, K. and T. Jesmin

2014. Comparative Analysis of Data Mining Classification Algorithms in Type-2 Diabetes Prediction Data Using WEKA Approach. *International Journal of Science and Engineering*, 7(2).

Ajinkya, K. and S. Basil

2017. A Survey on Machine Learning Algorithms for Building Smart Systems. *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 5(1).

Arnold, K., J. Gosling, and D. Holmes

2006. *The Java programming language*, 4th ed edition. Upper Saddle River, NJ: Addison-Wesley.

Candel, A., E. LeDell, V. Parmar, and A. Arora

2017. *Deep Learning with H2O - Booklet*, 5th Edition. H2O.ai, Inc.

Chapelle, O., B. Schölkopf, and A. Zien, eds.

2006. *Semi-supervised learning*, Adaptive computation and machine learning. Cambridge, Mass: MIT Press. OCLC: ocm64898359.

Connolly, T. M. and C. E. Begg

2005. *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson Education.

Gagniuc, P. A.

2017. *Markov chains: from theory to implementation and experimentation*. Hoboken, NJ: John Wiley & Sons.

Gantz, J. and D. Reinsel

2012. THE DIGITAL UNIVERSE IN 2020: Big Data, Bigger Digi tal Shadows, and Biggest Growth in the Far East. IDC.

Gertz, M., T. Özsu, G. Saake, and K.-U. Sattler
2004. Report on the Dagstuhl Seminar "Data Quality on the Web". *ACM SIGMOD Record*, 33:127–132.

Golfarelli, M. and S. Rizzi
2009. Expressing OLAP Preferences. In *Scientific and Statistical Database Management*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, and M. Winslett, eds., volume 5566, Pp. 83–91. Berlin, Heidelberg: Springer Berlin Heidelberg.

Grolinger, K., W. A. Higashino, A. Tiwari, and M. A. Capretz
2013. Data Management in Cloud Environments: NoSQL and NewSQL Data Stores. *J. Cloud Comput.*, 2(1):49:1–49:24.

Halkidi, M., Y. Batistakis, and M. Vazirgiannis
2001. On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, 17:107–145.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten
2009. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10.

Ioannidis, Y. E.
1996. Query optimization. *ACM Computing Surveys*, 28(1):121–123.

Jarke, M. and J. Koch
1984. Query Optimization in Database Systems. *ACM Comput. Surv.*, 16(2):111–152.

Kohavi, R.
1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Pp. 1137–1143. Morgan, Kaufmann.

Kohonen, T.
2001. *Self-Organizing Maps*. Berlin, Heidelberg: Springer Berlin Heidelberg. OCLC: 851768192.

LaBrie, R. and L. Ye
2002. A PARADIGM SHIFT IN DATABASE OPTIMIZATION: FROM INDICES TO AGGREGATES. *5*.

LeCun, Y., Y. Bengio, and G. Hinton
2015. Deep learning. *Nature*, 521(7553):436–444.

Madasamy, B. and J. J. Tamilselvi
. Improving classification Accuracy of Neural Network through Clustering Algorithms. *International Journal of Computer Trends and Technology*, Pp. 3242–3246.

Marakas, G. and J. O'Brien
2010. *Management Information Systems*. McGraw-Hill Education.

Marr, B.
2016a. Big Data: 20 Mind-Boggling Facts Everyone Must Read. *Forbes*.

Marr, B.
2016b. Big Data Overload: Why Most Companies Can't Deal With The Data Explosion. *Forbes*.

Portela, M.
2013. Preface - Admirável Mundo Novo (Brave New World by Aldous Huxley). In *Admirável Mundo Novo*, volume 1, Pp. 7–17. Antígona.

Redman, T. C.
2008. *Data driven: profiting from your most important business asset*. OCLC: 865508809.

Rocha, D. and O. Belo
2015. Integrating usage analysis on cube view selection - an alternative method. *International Journal of Decision Support Systems*, 1(2):228.

Russell, S. J. and P. Norvig
2003. *Artificial intelligence: a modern approach*, Prentice Hall series in artificial intelligence, 2nd ed edition. Upper Saddle River, N.J: Prentice Hall/Pearson Education.

Sakr, S., A. Liu, D. M. Batista, and M. Alomari
2011. A Survey of Large Scale Data Management Approaches in Cloud Environments. *IEEE Communications Surveys Tutorials*, 13(3):311–336.

Stehman, S. V.
1997. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1):77–89.

Sumithra, R. and S. Paul
2010. Using distributed apriori association rule and classical apriori mining algorithms for grid based knowledge discovery. Pp. 1–5. IEEE.

Tan, P.-N., M. Steinbach, and V. Kumar
2005. *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Valpola, H.
2000. *Bayesian Ensemble Learning for Nonlinear Factor Analysis. Acta*. PhD thesis, Polytechnica Scandinavica, Mathematics and Computing Series, No. 108.

Vapnik, V. N.
2000. *The nature of statistical learning theory*, Statistics for engineering and information science, 2nd ed edition. New York: Springer.

Wang, R. Y. and D. M. Strong
1996. Beyond Accuracy: What Data Quality Means to Data Consumers. *J. Manage. Inf. Syst.*, 12(4):5–33.

Witten, I. H. and E. Frank
2005. *Data mining: practical machine learning tools and techniques*, Morgan Kaufmann series in data management systems, 2nd ed edition. Amsterdam ; Boston, MA: Morgan Kaufman.

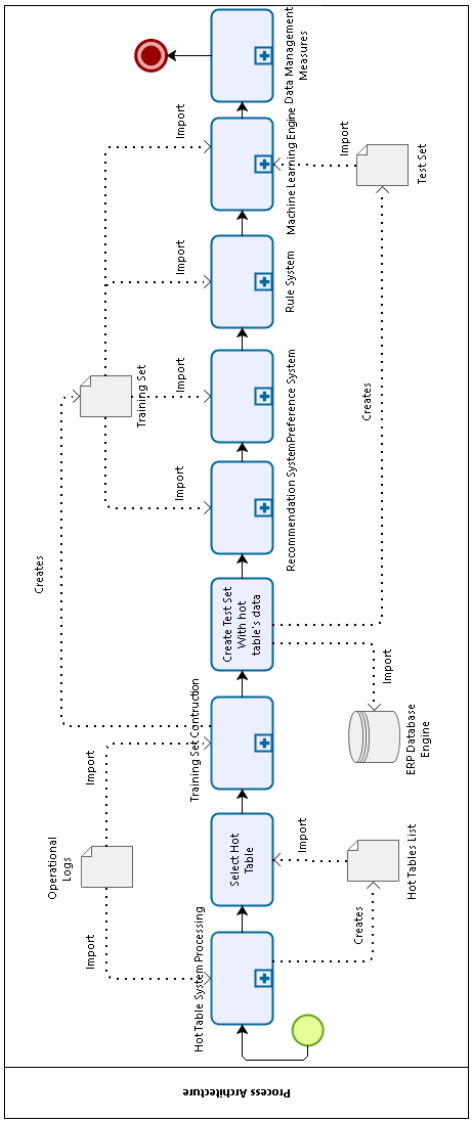
Zhang, S., C. Zhang, and Q. Yang
2003. Data preparation for data mining. *Applied Artificial Intelligence*, 17(5-6):375–381.

Zhihai Wang and G. Webb
2002. Comparison of lazy Bayesian rule, and tree-augmented Bayesian learning. Pp. 490–497. IEEE Comput. Soc.

Zhu, Y., N. Zhong, and Y. Xiong
2009. Data Explosion, Data Nature and Dataology. In *Brain Informatics: International Conference, BI 2009 Beijing, China, October 22-24, 2009 Proceedings*, N. Zhong, K. Li, S. Lu, and L. Chen, eds., Pp. 147–158. Berlin, Heidelberg: Springer Berlin Heidelberg.

A

ANNEXES



The processing schema of the prototype.