U.PORTO
FEP FACULDADE DE ECONOMIA
UNIVERSIDADE DO PORTO

Customers' Bundles' revenue fluctuation in a Telecommunication company: Data Warehouse Construction and Visualization

Cândido Rocha

Dissertation

Master in Modelling, Data Analysis and Decision Support Systems

Supervised by:

PhD Bruno Miguel Delindro Veloso

PhD João Manuel Portela da Gama

Cláudia Isabel Maia Dias

2023

Acknowledgments

I would first like to thank both of my supervisors, Professor João Gama and Bruno Veloso, for their constant feedback and advice regarding both the written and practical parts of this project.

I would also like to acknowledge NOS for giving me the opportunity to do a curricular internship at their Planning and Management Control department, thus allowing me to develop this thesis. Of course, a group of people that should not be forgotten were the colleagues I met at this department who supported me throughout all this project. From them, I would like to give a special thanks to Cláudia Dias (my supervisor), and Liliana Fernandes, who helped me in every step of the process with their patience and expertise.

Another group that could not be forgotten is my family, who supported and encouraged me all the way.

Lastly, I would like to thank everyone that directly or indirectly contributed to the success of this dissertation.

Abstract

Nowadays, a concept that has been gaining traction in all industry sectors is called data-driven companies. This term, simply put, means that a company bases its decision-making on the analysis and interpretation of data, making full use of Business Intelligence to improve its business performance. The telecommunication sector is no exception to this rule, needing it to analyze vast amounts of data to make decisions. However, for a company to be data-driven, it needs well-structured databases to make its analysis.

Therefore, this dissertation is within the scope of a curricular internship in the Planning and Management Control (P&C) Department of NOS. Currently, no unified database can help with a detailed explanation of bundles' revenue fluctuations across the months. With this in mind, this project's main aim is to create a data mart containing the most important information regarding the customers' bundles and design a dashboard that would help its users analyze this data. The data used in this work will be related to the customers' bundle information and the associated billing information, being the period considered in this work ranging from December 2021 to January 2023.

Firstly, the data had to be extracted from an already existing data warehouse, and then it had to be transformed and modeled into multiple tables in SAS. After this, the resulting tables were imported into Power BI, where all the connections between the tables were established, and new variables that would help with the analysis were formed. With these steps done, the dashboard was created to help the P&C with their analysis. The resulting dashboard comprises seven pages: one that explains how the dashboard is supposed to be used; and six more pages that use interactive visual analysis. All the data in this dashboard can be filtered according to the customers' segments, and by the period the user wants to analyze.

Keywords: Data Mart; Relational Database; Dashboard; Telecommunications; Bundles' Revenue Analysis; Interactive Visual Analysis

Resumo

Atualmente, existe um conceito que vem ganhando tração, em várias indústrias, chamado de companhias *data-driven*. Este termo, de forma simples, significa que uma companhia baseia a sua tomada de decisão na análise e interpretação de dados, fazendo uso de Inteligência de Negócios para melhorar o desempenho da sua empresa. O setor das telecomunicações não é exceção à regra, precisando de analisar vastas quantidades de dados para poder tomar decisões. Contudo, para uma companhia ser *data-driven*, precisa de bases de dados bem estruturadas para fazer as suas análises.

Por isso, esta dissertação insere-se no âmbito de um estágio curricular no Departamento de Planeamento e Controlo de Gestão (P&C) da NOS. De momento, não existe uma base de dados unificada que possa ajudar a explicar detalhadamente as flutuações de receita dos pacotes ao longo dos meses. Com isto em mente, o principal objetivo deste projeto é criar um *data mart* que contenha as informações mais importantes acerca dos pacotes dos clientes e projetar uma *dashboard* que ajude os seus usuários a analisar esses dados. Os dados utilizados neste trabalho serão relativos à informação dos pacotes dos clientes e à sua informação de faturação, sendo o período considerado neste trabalho compreendido entre dezembro de 2021 a janeiro de 2023.

Primeiramente, os dados tiveram que ser extraídos de um *data warehouse* já existente, transformados e modelados em várias tabelas, em SAS. Depois disso, as tabelas resultantes foram importadas para Power BI, onde todas as ligações entre tabelas foram estabelicas e novas variáveis, que ajudariam na análise, foram criadas. Com essas etapas concluídas, a *dashboard* foi criada para auxiliar o P&C nas suas análises. A *dashboard* resultante é composta por sete páginas: uma que explica como a *dashboard* deve ser usada; e mais seis páginas, que fazem uso de análise visual interativa.

Palavras-chave: Data Mart; Banco de Dados Relacional; Dashboard; Telecomunicações; Análise de Receita de Pacotes; Análise visual interativa

Glossary

3NF - Third Normal Form
BI - Business Intelligence
BDW - Business Data Warehouse
DM - Data Mining
DW - Data Warehouse
ETL - Extract-Transformation-Load
FTTH - Fiber-to-the-Home
HOLAP - Hybrid Online Analytical Processing
IF - Fixed Internet
IM - Mobile Internet
IOT - Internet of Things
IVA - Interactive Visual Analysis
MOLAP - Multi-dimensional Online Analytical Processing
OLTP - Online Transactional Processing
OLAP - Online Analytical Processing
P&C - Planning and Management Control
ROLAP - Relational Online Analytical Processing
SCD - Slowly Changing Dimensions
TV - Television
VF - Fixed Voice
VM - Mobile Voice

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation and Problem Description

Throughout the years, businesses have employed different approaches to making their decisions, but nowadays, there is a type of decision-making based on data analysis and Business Intelligence (BI) that is becoming the standard in multiple industries. This type of decision-making is called data-driven, and companies use it to make better, more informed, and impactful decisions. As it should, the telecommunication sector, with their vasts amounts of data, is adapting its tasks to incorporate this type of decision-making.

One crucial task this type of company does to enrich its decision process is understanding the customers' bundles' revenue fluctuations between months. A customer bundle is the combination of services a telecommunication company provides, including one or more of the following services: television, fixed voice, fixed internet, mobile internet, mobile voice, and IOT (Internet of Things) gadgets. So, this analysis is comprised of understanding the fluctuations of clients between types of bundles that impacted the bundles' revenues. This is a fundamental analysis for the telecommunication sector as this type of revenue is the primary source of revenue that these companies detain. However, this revenue source can sometimes take time to understand since many variables can influence it. Therefore, the right tools are needed so that the time spent in this process is reasonable and the correct conclusions can be taken.

At NOS, a Portuguese Telecommunications Company, the department with this core task is the Planning and Management Control Department (P&C). Currently, the bundles' monthly revenue analysis is very basic (gives only some of the necessary insights). It is based on a process that involves making complex SQL queries to filter only the necessary data for the analysis, roughly summarizing it, and then transferring this data to Excel to try and understand the implicit trends that influence the bundles' revenues. This analysis is done mostly manually and without the necessary detail, thus causing it to be time-consuming and not return the required insights. The main reasons for these problems are not only the enormous amount of data that this company detains but also the challenge of organizing all this data in an insightful and useful way so that the P&C can add more value to business analysis. Firstly, there needs to be a database with the needed level of detail. Secondly, when some change happens to a bundle, its identification (id) also changes. This last aspect makes the analysis harder because, as the bundles change their

id, it becomes difficult to connect bundles that had alterations made to them and their predecessors (bundles before the changes happened). Not being able to relate these bundles between periods, the understanding of what changes in the bundles impacted the revenues the most is compromised.

To address this problem, this dissertation is being written within the scope of a curricular internship in the Planning and Management Control Department of NOS.

With this in mind, the main objective of this thesis is to build a data mart that facilitates analyzing customers' bundles' revenue fluctuations and understanding their variations between months. Data visualization techniques will also be used to interpret the data better, and some insights about trends and patterns will be taken from them.

1.2 Dissertation Structure

This report is structured as follows: Chapter 2 is compressed of the main insights drawn from the literature review regarding telecommunications, data warehousing, data quality, data visualization, and interactive visual analysis. Chapter 3 includes a description of the data source from which the data will be retrieved, a description of the data itself, the methodology applied in the following chapters, and an explanation of the software used. Chapter 4 will explain the tables implemented in the data mart and how some of their variables were created. In Chapter 5, the attention will be turned toward studying how the composition of the dashboard and its analysis. Chapter 6 includes some conclusions about the work developed in this thesis and some suggestions for future works. At the end of this work, references and appendix is used to support the facts written in this work.

Chapter 2

Related Work

2.1 Data Mining in the telecommunication sector

The telecommunication sector companies need to store a considerable amount of data that can be, for example, related to the time of a transaction, prices of their products, quantities sold, consumer credentials, and many others (Thackeray et al., 2008). This information is very useful because, by analyzing it, these companies can derive conclusions about consumer behavior, patterns, and trends (Yin & Kaynak, 2015). With all this information and the intense competition in this market, implementing better analytical tools can be profitable to these companies (Fatusha & Ktona, 2016).

Two helpful areas that may help in dealing with large amounts of data are Business Intelligence and Data Mining (DM). Most BI applications pass through analyzing immense amounts of data to support companies' decision-making (Qiongwei et al., 2012). On the other hand, according to Silwattananusarn & Tuamsuk (2012), for Data Mining to be applied, it is necessary to have an integrated database with enough quality to help make decisions. Otherwise, the decisions taken from it can be inaccurate and negatively impact the company. Ibrahim et al. (2021) supports the idea of data quality which refers to the need for high-quality data to get more accurate, for example, financial reporting, better performance measurements, and reliable budgeting in companies.

So, a fundamental step to apply either Data Mining or Business Intelligence is to implement a technique capable of structuring usable databases (Habul & Pilav-Velic, 2010). Therefore, the response to this problem should be to find a process that can structure and summarize the needed data while guaranteeing its quality.

Data warehousing is one way, found in the literature, to structure the data. This can be a good approach for this problem, as Data warehousing and Business Intelligence are two areas that highly interact, being BI, as explained, helpful in decision-making. Data warehouses (DWs) are normally used as an information source for Business Intelligence, which causes them to be viewed as one entity. BI combined with data warehousing can provide insights that may not be available through other means (Gonzales et al., 2011). Eventually, data warehousing can also help Data Mining as this process makes data processing easier (Inmon, 1996).

These structures have proven helpful in the telecommunication sector as some works involv-

ing telco companies successfully implemented DWs. Some examples are in the works of Shin et al. (1998); Trisolini et al. (1999); Qiongwei et al. (2012); Calvanese et al. (2006). However, in the literature, there was nowhere to be found an implementation of a warehouse that directly deals with solely the bundles' revenue part of telecommunications.

In the next steps of this work, some literature reviews of data warehousing will be presented.

2.2 Data Warehousing Concepts

DWs were introduced in the 1980s, but their use only got traction in the 1990s. Since then, almost every enterprise has used data warehousing infrastructure to support their decision-making (Reddy et al., 2009). This highly regarded tool is considered very powerful for decision support and Business Intelligence tasks (Draheim, 2013).

DWs are a helpful tool that retrieves information from multiple heterogeneous sources (e.g., data lakes, operational systems, etc.) and consolidate it into a unique database. When this database is implemented, its information can be easily organized, updated, summarized, and may be very helpful for the company that detains it (Sumathi & Sivanandam, 2006). A more formal definition for DW can be that of Bill Inmon, which refers to a data warehouse as a storage system that is subject-oriented (provides information about a topic and not about the whole business), integrated (integrates into a single structure data from different sources), time-variant (keeps track of historical data), non-volatile (data is not changeable) collection of data Singh & Ghate (2017)

A data warehouse can have multiple names depending on its size and how much information it detains about the business. A Business Data Warehouse (BDW) comprises information concerning all corporate levels, such as logistics, controlling, and finance. Contrarily, a Data Warehouse may only be specific to a small part of the business, for example, a department (Köppen et al., 2015). Because sometimes it is challenging to implement a BDW, it is normal for a business to have multiple DWs, but this can be a double-edged sword. Since there are multiple data warehouses, there can be an overlap in their information. This overlap in information can lead to data accuracy and validation problems, so companies must take extra care when creating and using DWs instead of a BDW (Sumathi & Sivanandam, 2006).

The lowest level of data aggregation in data warehousing is a data mart. Data marts are subsets of the DW tailored towards a single business process (requirement-driven). These subdivisions of the DW are fed with granular data that is then reshaped according to its purpose (Bonifati et al., 2001).

2.3 Data Warehouse Architectures

If a company wants to implement a data warehouse or the other two structures (BDW or data mart), there are two main approaches in the literature (Velicanu & Matei, 2007). The first one is called the top-down approach, introduced by Bill Inmon. This method advocates in favour of building a single normalized database for the whole company, a BDW, which keeps the data at the smallest level of granularity, being the departments obligated to follow the centrally designed structure. After the BDW is built with this architecture, some data marts are, posteriorly,

derived to support the individual department's analytical needs (Kwasowiec & Karwowski, n.d.). The central assumption behind this type of warehousing is that the BDW should respond to the whole enterprise's needs (Velicanu & Matei, 2007). The second approach to data warehousing (dimensional modelling or bottom-up approach) was introduced by Ralph Kimball. This more flexible approach does not require a BDW, allowing each department to create its data marts, basing its structures on the star schema. This method implies creating multiple data structures across the company instead of one. Even though dimensional modelling is not focused on the business-wide view, Kimball provides an architecture for implementing an enterprise-wide warehouse. The method to accomplish this is called data warehouse bus architecture, which process, to make it simple, puts together the multiple data marts that the business detains, like the pieces of a puzzle, until the whole data is integrated within the business data warehouse (Kimball & Ross, 2011).

On the one hand, Inmon's approach is more suitable if a company wants to design a single data warehouse from where all the data marts come, thus requiring more rigorous analysis and design than Kimball's. On the other hand, Ralph Kimball's approach is concerned more with optimizing the database for faster data retrieval, making data more understandable, and increasing query performance. However, it should be noticed that Inmon's warehouse approach is easier to maintain, being less prone to redundancies and errors (Velicanu & Matei, 2007).

Even though these approaches have their differences, both have a common underlying process called ETL (Moscoso-Zea et al., 2018). ETL is an acronym that stands for extract-transformation-load and can be summarized in four steps: the identification and extraction of the data from the required data sources and its imputation into the processing system; transformation of the extracted data (cleaning the data, assigning warehouse keys, schema mapping, …); integration of the data into a single format (in this stage values deduplication is done as well); and, the final phase, is the loading of the data into its final destination (Kimball & Ross, 2011), the BDW or data mart depending on the approach taken (Moscoso-Zea et al., 2018).

Vassiliadis et al. (2002) refers that this phase, the ETL, is the most time expensive phase of data warehousing, being possible for it to take up to 80% of the time used for the DW construction.

2.4 OLTP and OLAP

Before the data can be used, it must be processed somehow. For this, there are two main systems: OLTP and OLAP. Typically, databases use online transactional processing (OLTP), which is more customer-oriented, prioritizing the customers' needs over the business' needs. OLTP was built to deal with operations faced daily by an organization. Examples of such operations are sales, inventory checking, accounting, and payroll. In summary, it manages data that is too detailed to be employed in decision-making. Conversely, data warehouses use Online Analytical Processing (OLAP). This system is more oriented towards data analysis, allowing users to quickly analyze large amounts of historical data, making it easier to summarize and aggregate data at different levels of granularity (Sethi, 2012).

A relational or multidimensional structure can organize the data in the data warehouse. De-

pending on the approach chosen, the OLAP applied to the DW will either be called Relational Online Analytical Processing (ROLAP) when the multidimensional data in the data warehouse is represented in a relational format or Multidimensional Online Analytical Processing (MOLAP) when the data is represented in a multidimensional format (also known as a cube) (Moura, 2012).

The advantage of using a ROLAP database is its large data storage capacity. Contrarily, MOLAP systems use pre-computation of data to achieve a faster query retrieval time, resulting in a lower storage capability (Moura, 2012).

To compensate for the problems found (slower processing time versus lower storage capacity), there is an architecture called Hybrid Online Analytical Processing (HOLAP), which has both ROLAP and MOLAP characteristics. This technique stores information on both multidimensional and relational databases. HOLAP was envisioned to have a short response time while supporting a dynamic approach to relational databases (Kazi et al., 2010). This system accomplishes this feat by storing the largest data in relational databases and the densest and most frequently accessed data in multidimensional servers (Pedamkar, 2022).

2.4.1 Schemas and OLAP Cube

To make the concepts of section 2.4 more understandable, in this section, the concepts of schema, OLAP cube, and concepts that are intimately related to them will be explained.

The most common way to represent a relational database is by a schema. A schema is a logical arrangement of data in a data warehouse. It defines how tables and data are organized, stored, and connected in a data warehouse. To build a data warehouse schema, it is necessary to arrange the data between two types of tables: fact tables and dimension tables. A fact table is a table that contains the measurements that are necessary to analyze a business process, such as revenues or sales. Besides the facts, this table possesses foreign keys that enable the connection to the dimension tables' primary keys. Furthermore, each fact table can have a composite key, that was created by combining multiple columns to uniquely identify an entity within the data warehouse (Kimball & Ross, 2011).

A dimension table contains the relevant information about a business. These tables describe the measurements in the fact table (Velicanu & Matei, 2007). They are usually small in terms of rows but can have many columns, having attributes that allow the performance of tasks that are important for Business Intelligence (e.g., querying, grouping, and reporting tables of the facts). Some common examples of the attributes in the dimension tables may be the products, customers, and time (Kimball & Ross, 2011). Kimball & Ross (2011) give much importance to these tables, referring that the DW is only as good as the quality of these. However, that is not all there is to dimension tables. Some dimension attributes may be static in a data warehouse, but others change slowly over time. Since data warehousing is used for analytical purposes, there is, most of the time, the need to maintain historical data in it, so it is then required to specify a strategy to handle the changes that may happen over time within it. To accomplish this, Kimball refers to slowly changing dimensions (SCD). To implement an SCD in a data warehouse, there are mainly three different strategies, being it possible to create hybrid approaches with two or more of the three. The SCD type 1 is simply overwriting the data in the dimensions (used when there is no need to compare with previous events). The SCD type 2, the most commonly used

approach, adds a new record to the dimension and relates this new record to a previously existing record through a key. As storing the values like this would be too confusing (because it would be difficult to differentiate between the most recent and the older records), at least three new columns should be added to the data warehouse, besides the relating key column, having them the following values: row effective date (since when the observation in that row is valid); row expiration date (until when the observation in that row is valid); and a flag indicating the most recent row (Kimball & Ross, 2011).

Since this type spawns new rows in the dimension tables, it can accelerate table growth. This accelerated table growth may imply that there are better methods to use in dimension tables with millions of rows because of scalability problems. However, if some adaptations are made, type 2 can still be implemented to multi-million-row tables. A solution for this is to isolate frequently analyzed or frequently changing attributes by moving them to new dimensions (mini-dimensions), thus eliminating the unnecessary volume in the main dimension. Finally, SCD type 3 adds a new column to a dimension. Instead of just a column for an attribute, there are two columns for the same attribute in each record, one column represents the attribute previously owned by that record, and the other represents the present one. With this last SCD type, it is only possible to compare values between two time periods, being impossible to go further into the past (Kimball & Ross, 2011).

Having the data modeler decided on how the dimensions are composed and organized, these can then be hierarchized. This technique enables the user to visualize the data at different levels of aggregation, thus permitting him/her to make multiple types of analysis. The hierarchization process is nothing more than building hierarchies within the dimension tables, being a hierarchy of the relationship between dimension levels. For example, a hierarchy for geographic locations could be composed of a country, its states/districts, and the cities in each state/district, where a country has the highest level of aggregation in the hierarchy, and the city has the lowest aggregation level (Talwar & Gosain, 2012).

Knowing the meaning of these tables, it may be easier to understand some of the different types of schemas in the literature. The star schema is the first and most used schema in the data warehousing literature and is already mentioned in this work. This schema has a single fact table that refers to multiple dimension tables. These dimension tables, consequently, have a primary key that makes it possible to link them to the fact table. One characteristic of this kind of schema is that it usually does not have an explicit hierarchy and allows faster querying compared to others (Chaudhuri & Dayal, 1997).

Conversely, snowflake schemas have an explicit hierarchy. This type of schema is similar to the star schema. The main distinguisher between the star schema and the snowflake is that the snowflake schema allows normalization of the dimension tables, while the star schema does not. This normalization process removes the dimensions' low cardinality attributes (attributes with many repeated values) by placing them in new tables. These newly created tables are then connected to the dimension tables, from which the normalized values were removed using artificially created keys. If the data modeler applies this schema, he/she will save disk space in exchange for a decrease in query performance (Sen & Sinha, 2005).

However, Moody & Kortink (2000) argues that neither the star schema nor the snowflake schema may be appropriate for data warehousing. On the one hand, the star schema can have

overlapping values in two or more tables; on the other hand, the snowflake schema defeats the purpose of simplicity by creating too many tables. Therefore, this author suggests an intermediate schema (star-cluster schema) that, instead of fully expanding the hierarchies, eliminates overlapping dimensions between tables by normalizing them. These overlapped dimensions are separated into a different table and then connected back to the tables from where they were removed, similarly to the previous schema (Moody & Kortink, 2000).

When referring to normalization, the most extreme case in data warehousing, which goes even further than the snowflake schema, is called the third normal form (3NF) schema. This is a schema where all tables are in the 3NF, being it normally used for BDWs or large DWs (Butt et al., 2012). The third normal form is a database normalization technique that minimizes data redundancy and improves data integrity. It is the third step in a series of normalization processes, having as a requirement the none existence of transitive functional dependencies in the data warehouse. A functional dependency is considered transitive if an indirect relationship exists between attributes in the same table. This means that 3NF enforces direct dependence of non-key columns on the primary key, forbidding the dependence on other non-key columns (Silberschatz et al., 2008). Even though this is an applied schema in data warehousing, many authors are against its use (Martyn, 2004), one of these authors is Ralph Kimball. Kimball thinks that it can become a highly complex schema for data warehouse queries, and it also can turn out challenging to understand or navigate through 3NF schemas, defeating the whole purpose of data warehousing (efficient and user-friendly data retrieval) (Kimball & Ross, 2011).

An additional schema that should be referenced is the Galaxy schema. It is constituted by multiple fact tables and dimension tables, described as a merging of multiple-star schemas. This is a helpful schema for a data warehouse with multiple fact tables that share dimensions (Moalla et al., 2017).

The description for some additional schemas, such as the flat and terraced schema, can be found in the work of (Başaran, 2005). Before going on, it should be noted that not all schemas were addressed in this project, here are only described the most referenced and some additional ones.

After being represented by a relational structure, the data warehouse can be used to form a cube (multidimensional structure), more precisely, an OLAP cube. An OLAP cube, also called a data cube, is a multidimensional database (Kimball & Ross, 2011). In the work of Gupta et al. (1997), the authors, to explain this concept, give the example of car sales. The value of the sales of vehicles is considered to be the metric of interest in the analysis and is then organized by several dimensions. Some examples given for the dimensions are model, color, and day of sale. In this example, the variable sales are deemed to be the whole cube, composed of cells (sub-cubes). Each sub-cube in the OLAP cube represents a unique combination of values across its dimensions, holding the measured value for that combination (Gupta et al., 1997). The big advantage of these structures is that they allow the pre-calculation of fields, which results in a faster query performance (Kimball & Ross, 2011).

By hierarchizing the dimensions, multiple operations can be applied to OLAP cubes, such as drill down, roll up, slice, dice, and pivot. Rolling up means going from a smaller aggregation level into a more complex one (going up in the hierarchy). Drill-down is the opposite of rolling-up when the user ungroups a dimension to get more granular data about it. Slice is when a specific

value of a cube's dimension is selected (e.g., a specific year) to represent a new sub-cube. Dice is similar to slice, but instead of selecting one specific value from a single dimension, values from multiple dimensions are selected to create a new sub-cube. Finally, pivoting is when the user reorients how he/she sees the data. For example, instead of the height axis of the cube being represented by cities, and the length axis by months, pivoting would swap these two axes to provide an alternative presentation of the data. Some other operations that might be used in dimensional modeling but are less referenced are ranking (sorting), selections, and defining computed attributes (Chaudhuri & Dayal, 1997).

2.5 Refreshing

A data warehouse is a system that is usually read-only, meaning that the people that use it cannot update or delete the data in it, in contrast with transactional systems, which allow users to update the system at any time (Rainardi, 2008). So, defining the refreshment strategy is another step in building a data warehouse. After the construction of the DW, there is a need, from time to time, to do this, so the data in the warehouse can be trusted (Chaudhuri & Dayal, 1997). To accomplish this task, it is necessary to execute it through different cleaning, integration, and filtering phases (ETL) (Brajković et al., 2020). As it can be understood, it is an essential task in data warehousing; if not done, the analyst will base their studies on old data, thus deriving wrong conclusions. However, there is the question of how often and how to refresh the data. Most of the time, the refreshing of the data does not need to be constant (every minute) unless the activity that the data is being used for relies on that (e.g., stock market), enabling it to be done periodically (e.g., daily, weekly, monthly) (Chaudhuri & Dayal, 1997). The frequency of refreshments depends on the company's objectives and the purpose of its data warehouse. The data refreshment can be warehouse-driven or user-driven, depending on the answer to these questions. The warehouse-driven policy implies that the refreshes are done less frequently and with more data per refresh, which leads to a more significant load time. On the other hand, with a user-driven policy, the updating is done more often, requiring less loading time (Mannino & Walter, 2006). From a survey performed by Mannino & Walter (2006), it was concluded that most companies, under survey, preferred the application of a mostly warehouse-driven warehouse policy, being the daily refreshment on nonworking hours more than enough to satisfy their company's needs. However, most companies had user-driven policies applied to a minor part of their data. Therefore, having a mixture of both approaches in most companies is useful.

Regarding how the refreshing is done, this can either be achieved by extracting the totality of the data source or incrementally (only new or changed data is updated). This last method allows the refreshment process to be scalable; that is, it allows the process to have a much lower computational time with the increasing data size, making it preferable to use (Chaudhuri & Dayal, 1997).

2.6 Data Quality

When talking about DWs, another topic that should be mentioned is data quality. Data quality is crucial to the users, so they can trust their data warehouses and make correct decisions. If data quality is high, it can translate into company gains; if it is low or has no quality, it may induce customer displeasure and help grow the costs associated with data warehousing projects (Benkhaled & Berrabah, 2019). Since data quality is inherent to all types of data, data warehouses are not an exception to the rule. DWs collect their data from transactional databases, thus being it possible to inherit the data problems present in these data sources. Some of the quality problems that may appear in these databases include incorrectly inputted data, missing values, and data inconsistency (e.g., a value represents different things in different periods) (Golab, 2013). Many data warehousing projects have been halted because of data quality problems, and there is even a study pointing out that 15 % to 20 % of data that most companies detain is barely usable or even unusable (Geiger, 2004).

So, what does it mean to have high-quality data? The simplest way to think about data quality is that it is high when the data meets the user's needs. However, this definition has profound implications. By this description, not all users will have the same opinion about the data, meaning that data quality is subjective to each user (Redman, 2012).

A definition of data quality that turns the previous definition more objective is proposed in the work of Redman (2012). In his work, Redman (2012) characterizes data of high quality as the data that can be utilized in the operations that require it, being this ability achieved when the data has no defects and has all the characteristics needed to complete the required tasks.

Some authors use dimensions to measure data quality, making its definition even more objective, as there are many dimensions in the literature (Loshin, 2009). An example of these can be found in the work of Ballou & Pazer (1985), where the authors defined four dimensions for data quality: consistency, accuracy, timeliness, and completeness. Wang & Strong (1996) went a few steps further and grouped some existing data quality dimensions into four classes. Following Wang & Strong (1996), the four data quality dimension groups are contextual, intrinsic, representational, and accessibility. The contextual quality is related to, as the name suggests, the situation in which the data is used. It comprises the amount of data, completeness, relevance, and timeliness dimensions. The intrinsic quality is related to identifying if the data conforms with the actual values. Its dimensions are accuracy, believability, reputation, and objectivity. Representational quality refers to how understandable the data in the system is. According to this point, the data must be presented so anyone using it can understand it. It relies on understandability, conciseness, and consistency quality dimensions. Lastly, accessibility corresponds to the ease with which the user can obtain the available data. This covers accessibility and security dimensions (Sebastian-Coleman, 2012). Since the literature on data quality is so vast and the number of dimensions that can be found in it is so big, it is normal for companies to choose just a couple of them to implement, considering only some of the existing dimensions while guaranteeing their data's quality (Redman, 2012). However, Redman (2012) pointed out that the previous taxonomy specified by Wang & Strong (1996) is a solid starting place to decide which are the necessary data quality dimensions for any company. Nevertheless, outside the scientific scope, some organizations put forward what they find to be the data quality dimensions most suitable to consider

while data warehousing, an example of this kind of organization is the Data Management Association (DAMA). This association recognizes the most critical dimensions to be completeness, uniqueness, timeliness, validity, accuracy, and consistency (Ramasamy & Chowdhury, 2020).

Completeness refers to the data meeting the user's expectations if the data has all the mandatory fields filled (e.g., first and last names are usually obligatory to fill, but middle names, most of the time, can be optionally filled). Uniqueness imposes that there should not be duplicate values. Timeliness refers to how up-to-date the information is. Validity imposes that the data's syntax (e.g., format, type, range, ...) follows the supposed definition. Accuracy refers to how well the data represents the real world. Consistency is the absence of difference (if the same information is stored and used at multiple instances match) (Ramasamy & Chowdhury, 2020).

Having the definition of data quality and knowing it may be a problem for the data warehouse now remains the question of when someone should check the data quality. For this, Hazen et al. (2014) makes an analogy by comparing the process of producing data to that of producing a physical product. In this analogy, Hazen et al. (2014) concludes that quality verification should be done throughout the entire data production process and not exclusively at the end. This way, errors can be rapidly fixed before they create an unbearable amount of errors to correct (snowball effect). To aid throughout all these phases, Redman (2001) suggests the use of visualizations, such as fishbone diagrams, Pareto charts, and histograms.

However, it is not enough to only apply data quality mechanisms, being it also necessary to implement what is called data governance, which can be defined as "the exercise of authority and control (planning, monitoring, and enforcement) over the management of data assets" (Mosley et al., 2010). Data governance defines which people can use which data, on what occasions they can use it, and by which methods they can access it (International, 2017).

Neff et al. (2013) showed that not having a data governance framework can harm a company, allowing each department to create its financial chart of accounts instead of having a single department in charge of doing that job (time spent in redundant work).

Recently, this concept of data governance has been increasingly accepted by companies because it has multiple benefits, such as confidentiality insurance, integrity, availability, and quality of customers' data (Al-Ruithe et al., 2019). Some other positive aspects of data governance are shown in the work of Brous et al. (2020), where the adoption of data governance by a company proved to enhance operational efficiency, diminish risks, boost profits, and help grow trust in the information overall (Al-Ruithe et al., 2019). Although data governance has been given some importance in recent years, it is still considered an understudied field in the scientific community (McCaig & Rezania, 2021).

2.7 Data visualization

Since the most important data warehousing topics have been addressed, this part of the project will focus on the literature review of data visualization and the method used to understand the data warehouse data.

Data visualization means drawing graphic displays to show data. Depending on the type of graph chosen, every data point in a database can be drawn (e.g., scatterplot), or only a statistical

summary will be presented (e.g., histogram) (Unwin, 2020).

This process is indispensable in any scientific process. If done correctly, it can allow the understanding and posterior communication of insights (e.g., patterns, trends, ...) taken from data (Tukey, 1977).

Data visualization helps understand the data and can be an aid tool for data cleaning, anomaly detection, and many other fields. Because this technique is so useful, it is often underestimated in textbooks (Unwin, 2020).

A common data visualization technique used in BI and decision-making is called dashboard (Yigitbasioglu & Velcu, 2012). A dashboard is nothing more than a collection of graphics. More specifically, a dashboard can be defined as a display composed of different indicators related to a subject, which are then organized in one or multiple visualizations (Schwendimann et al., 2016). A good reason to use such simple tools in decision-making is their capability of compressing vast amounts of information in one place (Holzinger, 2013). For a dashboard to be useful, it should have just the right amount of information because insufficient information in a dashboard can make it worthless (insufficient information does not allow the user to get the full picture) (Kelton et al., 2010). On the other hand, too much information can produce the same effect due to humans' capacity to process what they see (Carrasco, 2011). A person can only pay attention to so many things at a time, so if the dashboard displays too much information, its effect will be similar to one with too little information. This means that the components in a dashboard must be well-defined to avoid overloading the decision-maker with unnecessary aspects (Toreini & Morana, 2017).

Therefore, the dashboard type should be decided before building a dashboard. In the work of Staron et al. (2015), the process of choosing the right dashboard model is divided into seven dimensions: defining the type of dashboard (kind of visualization needed to allow knowing the status of a project); data acquisition (figuring out how the data is going to be extracted and inputted into the dashboard); stakeholders (defining the type of stakeholders to which the dashboard is meant and tailor it based on it); delivery (discover the way the data is going to be provided to the stakeholders); update (define how and with which frequency the data that supports the dashboard is updated); aim (define what the dashboard should accomplish); and data flow (define the key performance indicators keeping in mind that they should be complemented with raw data).

After this process, according to Staron (2015), the final dashboard should have three elements: a heading, visualizations of the metrics to understand the key performance metrics better, and a concise explanation of the information presented. The purpose of the heading is to explain the content presented in the dashboard. Additionally, this same author recommends that everything present in the dashboard should have some relation between them; the number of elements on each dashboard should not surpass seven (at most nine); and there should not be too many colors being used in the dashboard (Staron, 2015).

Since the first step of creating a dashboard is picking the right type of graphic, a brief literature review should be done in this area to decide which are the most appropriate graphic for a multidimensional database.

To represent a multidimensional database, it is necessary to represent two or more dimensions simultaneously. Examples of this type of visualization may be scatter plots, waterfall charts, pie charts, stacked bar graphs, and many more (Islam & Jin, 2019). However, these traditional

visualization tools may only be appropriate when the data size is manageable. So, the user needs to pick better options for when the data size is not manageable (Ali et al., 2016). Some visualization methods suggested by Ali et al. (2016), for this scenario, are treemaps, circle packings, sunbursts, parallel coordinates, stream graphs, and circular network diagrams. Sachinopoulou (2001) suggests the employment of some additional representations, such as trees and glyphs.

2.8 Interactive Visual Analysis

An additional area that can help visualize multidimensional models is interactive visual analysis (IVA). IVA consists of an aggregate of multiple views linked together and integrated with powerful data analysis techniques, which are based on statistical analysis, pattern recognition, machine learning, and other scientific fields. This area allows a balance between human visual inspection, computer-based analysis, and reasoning (Konyha et al., 2009), thus, allowing the identification of relations between elements that are sometimes too difficult to be detected by only one of these fields (Freiler et al., 2008).

With this analysis method, analysts can create visualizations that can be manipulated. Such processes like highlighting patterns, investigating hypotheses, and drilling down data are made easier (Heer & Shneiderman, 2012). The most common methods to enable this are selection, navigation, and coordination. Selection is when the user can manipulate a set of objects in a visualization, so information about a specific topic can be accessed with more detail. This type of manipulation can be translated into actions, such as highlighting or filtering. Navigation permits the user to travel the view more freely by, for example, allowing him to zoom in and out of specific locations or by simply allowing the user to mouse drag (move) across the visualization (Heer & Shneiderman, 2012). The term, coordination of views, infers that, when one user interacts with a visualization, other visualizations will be altered somehow. Some usual coordination techniques are brushing and navigational slaving. Brushing means that when the user selects an element in one view, the same or a related element will be highlighted in the other linked views. Navigational slaving, on the other side, refers to the coordination of navigational actions between linked views (Scherr, 2008).

To validate the usefulness of these visualizations, a standard method requires surveys or interviews. For these surveys, the interviewers let a group of people, at first, utilize the interactive views for a few moments and then ask them to complete a set of tasks, posteriorly questioning them about their user experience. Through this method, the interviewer can derive quantitative and qualitative measures to assess the performance of its interactive visualization(s) (e.g., the accuracy of the answers given, the completion time of each task, and the subject's opinion about the visualizations), while also finding which are the functionalities that need to be improved or added (Viana & Cabral, 2020).

Chapter 3

Methodology

3.1 Data Description

To fulfil the data warehousing project and data visualization creation needed for this work, it is required to understand more about the business and, from this understanding, derive what variables may be needed for the bundles' revenue analysis.

For this, it is essential to describe the services that NOS provides and some of the characteristics related to the customers that might influence the bundles' revenues.

The clients can be divided into two main segments: consumer (e.g., families, individuals, ...) and business (e.g., companies). Additionally, each customer has multiple activation dates (date when the client acquired the services) and can have multiple loyalty end periods (period until which the client has a binding contract with NOS, having to pay the whole contract value if he/she leaves NOS before this period). These two last variables may not be related to the monthly revenues. Nevertheless, they are associated with the entry or possible exit of customers, which may increase or decrease revenues.

Regarding NOS' bundles, they can also be organized differently. The first distinction between bundles is the type of bundle, which values can be standard bundles or custom-made. The first type (standard) refers to bundles, which services are already defined by NOS. Custom-made bundles are those that allow the customer to adjust the services in them.

A second distinguisher is their native offer, which can be professional (products designed for companies) or residential (designed for individuals). Depending on the native offer, there are different distribution channels associated. There are channels more oriented towards selling NOS' professional services (B2B/empresarial) and others more oriented towards NOS' residential services (B2C/residencial). Subsequently, these channels (B2B and B2C) can be subdivided into distribution networks, direct sales, telemarketing inbound, telemarketing outbound, and online sales (web RL). This hierarchy of the distribution channels is represented in Figure 3.1.

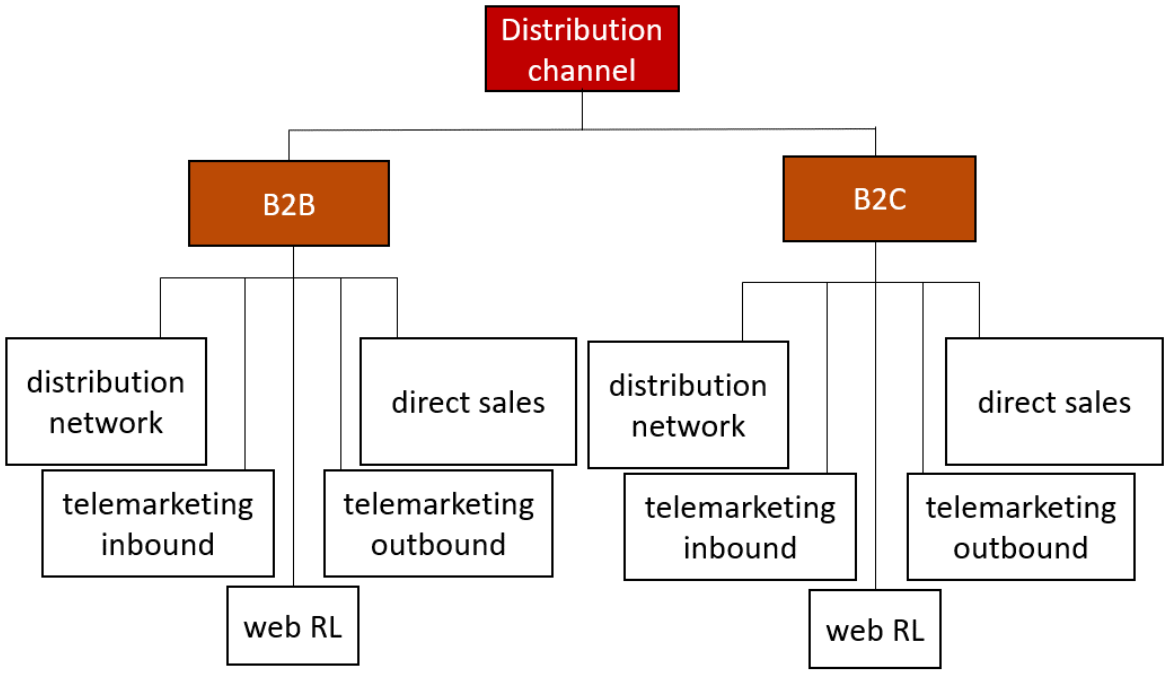

Figure 3.1: Hierarchy of the distribution channels

Another variable that creates heterogeneity between the bundles sold by NOS is how many services are in each bundle and what type of services are included. To this extent, the bundles can have a combination from one to six services, being all the possible available services in a bundle: television, fixed telephone, fixed internet, mobile phone, mobile internet, and IOT (devices that communicate with other devices, thus enabling the access to services, such as telemedicine, localization services, fleet management, tele-security...). These combinations of services correspond to the bundles' profiles and are numbered from 1P to 6P (the number preceding the P refers to the number of services in the bundle). Additionally, if the bundle has any mobile or IOT services, it can have a different number of SIM cards and IOTs associated to it.

To complement the services' description, there is a characteristic referent to the type of technology used by each product. Not every client can have any technology, as this depends on the area where a person lives.

The technology available in a particular area can either be wireless or wired. The wireless technology is based on Satellite (DTH), whereas the wired technology is based on either an FTTH network (optical Fiber-to-the-Home) or NOS' cable network (HFC/cabo). Concerning the FTTH, there is a need to subdivide it into three subcategories: FTTH, FTTH OUTROS, and FTTH DST. This distinction is made because NOS does not have this technology all over Portugal, so it has agreements with two other telecommunication companies that enable it to use its FTTH. This subdivision is unnecessary for the cable since NOS only uses its network.

Consequently, depending on the type of technology a client owns, there can be a user interface associated with it, being the existing user interfaces and their possible combinations with technology displayed in Figure 3.2. Before explaining the rest of the data, it should be clarified

that a user interface is a system that allows the user to access TV services, so some bundles will not have a user interface as they do not have TV services.

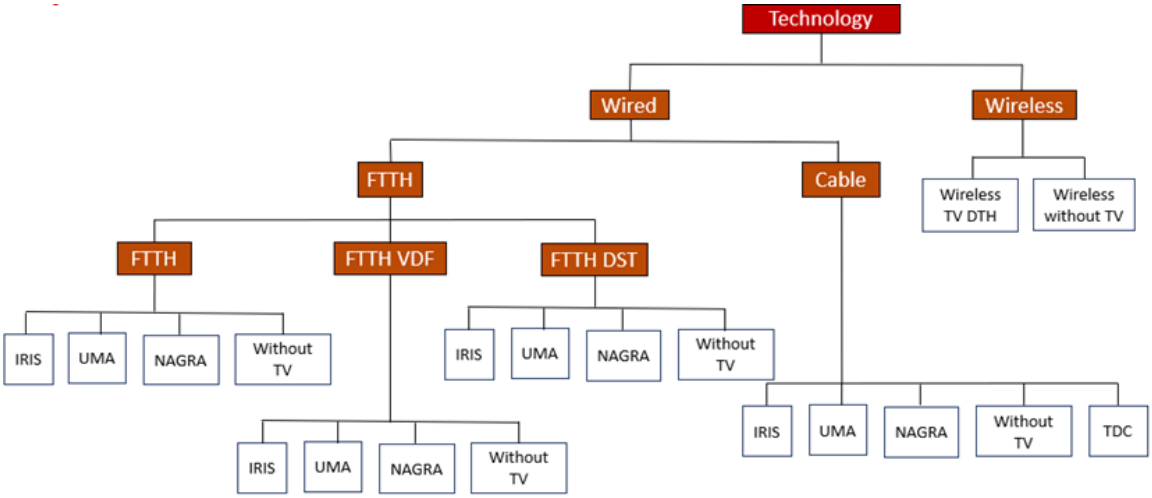

Figure 3.2: Technology and user interface combinations

Besides the variables mentioned above, there will also be two last groups of variables needed in the data warehouse. The first one is the location in which each client installs their services. This dimension will comprise the district, county, parish, and zip code.

Another one is time; as the analysis in this project needs to compare the revenue between months, a variable that organizes the revenue influxes by month and year will be needed. Additionally, it was opted to organize the time by the quarters, as well.

To build the data warehouse, only a sample of the data will be used, which will range from December 2021 until January 2023.

3.2 Data Source

As explained, data marts can retrieve their information from multiple data sources. Therefore, it is important to describe the data sources from which the data disclosed in section 3.1 will be retrieved.

It should be noticed that NOS resulted from a merger of two companies and had to combine the data sources from both. This combination resulted in multiple data sources/warehouses, each detaining information referent to different data domains. However, some sources must be structured to respond to every type of analysis needed. This is the case of the database that NOS detains regarding the services they provide to their clients. This data warehouse has much information related to each NOS' product (not only services within bundles), having information such as where/when the client acquired the service and its revenues, detaining hundreds of columns describing the millions of services NOS provides.

Having this much data, NOS sees this data warehouse as a starting point for every type of analysis that implies the need for customer and bundle information (e.g., ML model creation,

reporting, …), thus being an important tool for developing this work. As mentioned, this data source has the information to complete the task. However, to perform it, it is still necessary that each user builds SQL queries to filter only the needed data, being necessary additional care in the process. This filtering has to be methodically done because if the user does not filter the table according to some rules, he/she can produce an incorrect output. To mitigate this problem, NOS already imposes data governance policies, providing its employees with instructions defining what type of care they should have in this procedure and a glossary defining every field. Another problem in this data set is that if a bundle changes its characteristics, its inherent id also changes. Therefore, a way to relate bundles that had a change in them between periods will be needed in this new warehouse, as mentioned at the start of this project.

Based on this analysis, it can be understood that NOS already has most of the necessary data for the bundles' revenue analysis. However, the data warehouse with this data is not perfect, needing a subset of this data to be restructured to optimize this analysis.

3.3 Methodology

In NOS, a data warehouse is available with the information needed about the bundles and their revenues, as explained, so this project will focus on building a data mart to help explain how client fluctuations between bundles affect revenues by retrieving the necessary information from this source.

Based on the work of Moody & Kortink (2000), the development of a data mart should be based on dimensional modeling because this author argues that this method structures the data in a way that makes the data mart easy to understand and use. So, according to Kimball & Ross (2011), the process of dimensional modeling can be subdivided into four steps: selection of the business process to model, declaration of the grain, choosing the dimensions to apply to each fact table and identifying the facts that will fill each row of the fact table.

A business process is an activity in a company that tries to accomplish one of its goals, usually supported by a data-collection system. So, in the first step, to select the business process to model, the data modeler should derive, from its understanding of the business process and its understanding of the data, the adequate measurements (facts) to be analyzed in the data warehouse. Considering the problem that the Planning and Management Control Department is facing, the business process will be the analysis of the fluctuations that happen to the customers' bundles' revenues, as mentioned (Kimball & Ross, 2011).

Next, Kimball & Ross (2011) suggests that the declaration of the grain of the business process should be made. This is just specifying what a row in a fact table should represent. The grain should characterize the most atomic data captured by the business process, that is, the data with the most detail, which cannot be subdivided further. This data type is optimal in dimensional modeling, as it can be rolled up in every possible way. If the data needs to be granular enough, an unexpected request can appear, requiring the data to be drilled down into further facts that do not exist. In this step, the modeler must also guarantee that all facts should be at the same aggregation level (Kimball & Ross, 2011).

Facing the problem at hand, the grain should be a client's service bundle in each period

(month). This may be a partial definition of grain as it is common for it to be redefined multiple times in the following steps (Kimball & Ross, 2011).

Having the facts and their granularity defined, the next phase is to choose the dimensions to apply to each fact table row. This step should be easy if the grain is plainly defined and the business process is clearly understood, as, in this phase, the modeler only needs to select all possible descriptors (dimensions) of the chosen measurements. After they are chosen, the dimension tables can be created. In this process, attributes correlated with each other should be kept in the same table, and every textual measure should be put in the dimension tables. It is good to place textual attributes in the dimensions because these attributes can be more easily correlated in the dimensions while allowing less space consumption by the warehouse (Kimball & Ross, 2011). Kimball & Ross (2011) also warns that there should not be more than 15 dimensions in a data warehouse/mart as this is typically a sign of having dimensions that are not independent, thus being appropriate to combine dimensions. The problem with having many dimension tables is that it may affect query performance and raise data warehouse maintenance issues. If some necessary dimensions cannot be defined in this phase, it may be necessary to return to the previous step and re-declare the grain.

Lastly, Kimball & Ross (2011) suggests determining the numeric facts that will populate each fact table row. In simple terms, the warehouse designer should pick the numeric facts in each fact table that relate to what the company wants to measure. In this stage, the data modeler should separate the facts by different tables, as the facts should be grouped by the grain they refer to. An additional suggestion is that the chosen facts should usually be additive figures.

According to Kimball & Ross (2011), after all these steps, the dimensional model can be created and represented as a schema, advocating him, as mentioned, in favor of the star schema. So, following these steps, the multidimensional data set will be represented in a schema format, not necessarily a star schema, as seen in Figure 3.3.

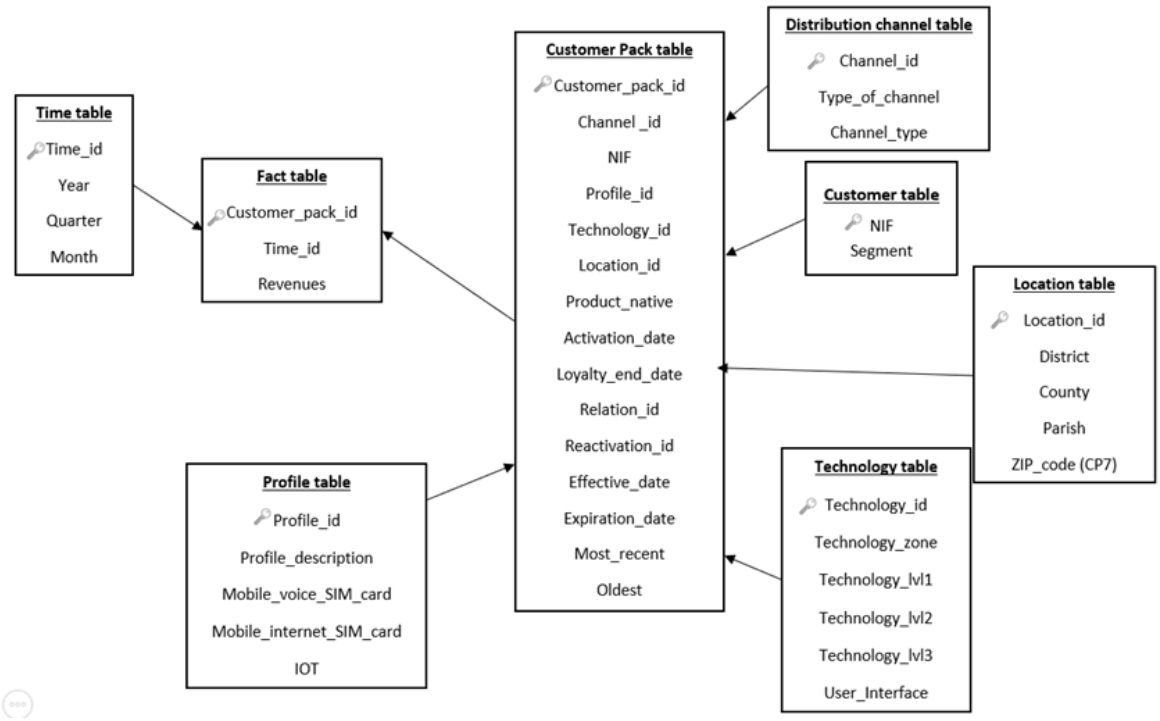

Figure 3.3: Schema design

To implement this schema, it was decided to use SCD type 2 to enable the user to check what characteristics each bundle had throughout their entire lifetime at NOS. To this effect, it is necessary to define a key that allows the relation between bundles that had changes made to them and their predecessors.

Then the ETL process will be performed simultaneously with simple quality checks of the fields. These quality checks will be useful to guarantee that there is no duplication of ids and also that there is consistency in the fields (e.g., misspelling verification and consistency in values meaning) during the ETL process. When done, the data will be imported into Power BI, so interactive dashboards can be created to determine how client fluctuations between bundles affect monthly revenues.

Finally, in the scope of data governance, written instructions on how to operate the dashboard and an explanation about some of its fields will be created to help the understanding of the model.

3.4 Software

As this thesis is being developed within the scope of a company, the pool of software to choose from is limited because the software to handle this data needs to follow NOS' security and privacy rules and should be software its employees are familiar with. To fulfil these restrictions, there are two options: SAS and Power BI.

Starting with SAS, SAS can read almost any data source (e.g., relational database systems, legacy mainframe systems, flat files, …). These files can be accessed natively or by many other

methods, including ODBC/JDBC or open standards supported with SAS Integration Technologies (Grasse & Nelson, 2006).

Still, the whole of SAS cannot be used in this work, as SAS detains multiple licenses built to accomplish specific tasks, being the only available license at the Planning and Management Control department, the SAS Enterprise Guide. SAS Enterprise Guide is a license from SAS that provides a graphical user interface that detains data integration, preparation, analytics, and reporting tools (Tomar et al., 2010). It is a flexible and user-friendly software appropriate for people with any coding level, allowing the operator to decide between writing code or using a point-and-click method, which replaces the need to write code. To employ this last method, the user has to use the query builder and then interact with the SAS' interface by specifying what he/she wants to do. Consequently, the instructions given are then submitted to the SAS server, and in return, an SQL code to perform the required task is generated by SAS, resulting in the implementation of the procedure. Besides being an easy-to-use program, as SAS Enterprise Guide is integrated with the Microsoft Windows Scheduler, it also allows the user to schedule the refreshing of its reports at regular intervals to be up-to-date (SAS, 2017).

Even though it has all these features, Tripathy et al. (2011) states that SAS Enterprise Guide is not the true SAS tool used for ETL, but it can be used for the ETL of projects with smaller sizes, like a data mart. However, this license is not appropriate for constructing dashboards, even though it can create some visualizations (e.g. bar charts, line charts, ...). For that, the BI Dashboard license would be needed (Tripathy et al., 2011).

On the other hand, Power BI is a robust cloud-based business analytics service that supports the development of dashboards, sharing of reports, direct data connection, and its incorporation. By detaining Power BI, the user can also access Power BI Desktop, which enables data to be transformed, report creation, and publish Power BI services (Negrut, 2018). Power Query is the technology that enables Power BI to transform the data needed for data warehousing. Power Query, in short, is a graphical user interface window that enables the user to compile the steps in the ETL phase, such as extracting and reshaping data (Becker & Gould, 2019). However, to not go into much detail about a technology that will not be used. This area of Power BI will not be explored more in-depth because, based on the experience provided by the department in which this thesis is being written, Power BI does not handle well the processing of large amounts of data (the kind of data being used in this work). Therefore, Power BI cannot be used for the data warehousing part of this work, needing the use of SAS, as it can handle databases with millions of rows, unlike Power BI (SAS, 2022).

However, Power BI will not be discarded as an option as it can be useful for the visualization part of the project. Power BI can connect directly to external data sources (Negrut, 2018), being compatible with SAS. It allows the elaboration of reports with multiple pages that supports rich visualizations, which can be created in a matter of minutes (Ali et al., 2016). Its dashboards can provide a 360-degree view of the business by retaining the most critical metrics in one place and updating them in real-time (Chawla et al., 2018). Like SAS Enterprise Guide, this software enables the operator to perform actions, including visualization creation, without requiring the user to know how to code. The graphs are created by a drag-and-drop process comprising different types of techniques. This software also provides an option to run R scripts to create graphs, requiring R coding knowledge (Ali et al., 2016). Because of all these functionalities, it is easy to

create visualizations in this software, and it is also easy to customize the visualizations made in it. Power BI has various fonts, colors, and presentation settings, making this software suitable for dashboard elaboration. Regarding the interactive visualization needed for this work, Power BI also supports linkage between visualizations, allowing visualizations to interact with each other (Becker & Gould, 2019).

Chapter 4

Data Mart Construction

To begin building this data mart, the data had to start being loaded into the dimension tables, beginning with those which are independent from others (with no foreign keys in them), such as the time table, location table, etc. The loading had to start with these tables because their values do not depend on the values on other tables. For example, if the user does not already have other tables ready when inserting data into the customer pack table, an integrity constraint will be activated in SAS, making it impossible to create this table. This integrity constraint specifies the need for the foreign keys to be already loaded into their respective tables in the data mart.

To not be too repetitive during the rest of the work, it should be explained, from the start, that all primary keys that do not have a "natural meaning" (have a real-life meaning) were created by applying an auto-increment method. Auto-increment is a commonly used method to create keys in a data set. This technique automatically assigns a unique numerical value to a new record each time it is added to a database table. This value is typically incremented by a fixed amount (usually 1) for each new record added to the table (Simplilearn, 2023).

Before going in-depth about the composition of these tables, it should be mentioned that, as data is not always perfect, some of the variables explained in section 3.3 may have some different values (e.g., replacement values for missing values, ...) than the exposed ones, in section 3.1.

4.1 Dimension Tables

4.1.1 Time table

The time table is the dimension in the data mart that allows the analysis of the data in it through time. This table comprises a single id (time_id) extracted from the original warehouse and four other time-related fields. The first three fields merely represent the years (Year), quarters (Quarter), and months (Month) in which a bundle revenue existed. The last one (full_date) is a field that shows the complete date when the revenue occurred. However, since no date in the original data specifies the day for this occurrence, this field is only partially accurate, having all its days set to 1. This field may seem redundant initially, but it can be very useful for some analysis in Power BI. With date fields like this, some functions that allow a more straightforward comparison of values between periods, such as the PREVIOUSMONTH function, can be used.

Time_id	Year	Quarter	Month	Full_date
202112	2021	4Q	12	01DEC21
202201	2022	1Q	01	01JAN22
202202	2022	1Q	02	01FEB22
202203	2022	1Q	03	01MAR22
202204	2022	2Q	04	01APR22
202205	2022	2Q	05	01MAY22
202206	2022	2Q	06	01JUN22
202207	2022	3Q	07	01JUL22
202208	2022	3Q	08	01AUG22
202209	2022	3Q	09	01SEP22
202210	2022	4Q	10	01OCT22
202211	2022	4Q	11	01NOV22
202212	2022	4Q	12	01DEC22
202301	2023	1Q	01	01JAN23

Figure 4.1: Time table

4.1.2 Distribution Channel Table

The distribution channel table is formed by the hierarchy that composes the distribution channel. It has a field (type_of_channel) that corresponds to the segment in which each distribution channel directs its sales (empresarial or residencial) and another variable (channel_type) that is comprised of the other sub-segments (distribution network, telemarketing inbound, telemarketing outbound, direct sales, and web RL) which appear on the bottom of the hierarchy represented in Figure 3.1. Besides these, the only remaining field in this table is the channel_id, which is simply its primary key.

Channel_id	Channel_segment	Channel_type
3	EMPRESARIAL	Rede Distribuição
4	EMPRESARIAL	Telemarketing Inbound
5	EMPRESARIAL	Telemarketing Outbound
6	EMPRESARIAL	Vendas Directas
7	EMPRESARIAL	Web RL
13	RESIDENCIAL	Rede Distribuição
14	RESIDENCIAL	Telemarketing Inbound
15	RESIDENCIAL	Telemarketing Outbound
16	RESIDENCIAL	Vendas Directas
17	RESIDENCIAL	Web RL

Figure 4.2: Distribution Channel table

4.1.3 Location Table

Another dimension is the location. Simply put, this table has all the possible combinations of districts, counties, parishes, and zip codes to which each bundle can be related. The zip code could have been used as the primary key in normal circumstances. However, some duplicate values for different location combinations appear in this field. Some attempts to solve this problem were tried, such as using geopy (from Python) or geocode (from SAS), but since there are errors in some addresses, it is very difficult to define zip codes for some cases. Facing this problem, the only alternative was creating a new primary key (location_id).

location_id	ZIP_cod	District	County	Parish
2	3850-001	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
3	3850-002	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
4	3850-003	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
5	3850-004	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
6	3850-005	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
7	3850-008	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
8	3850-009	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
9	3850-010	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
10	3850-011	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
11	3850-012	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
12	3850-014	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
13	3850-016	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
14	3850-017	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
15	3850-019	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
16	3850-022	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
17	3850-024	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
18	3850-027	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
19	3850-030	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
20	3850-031	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
21	3850-034	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior
22	3850-035	Aveiro	Albergaria-A-Velha	Albergaria-A-Velha E Valmaior

Figure 4.3: Location table

4.1.4 Technology Table

The technology table comprises all the technology-related fields and its primary key (technology_id). Here we have the technology zone, which specifies which type of technology is available in each region (technology_zone), the variables technology_lvl1 (characterizes if a bundle has either wired or wireless services), technology_lvl2 (characterizes if the technology in the bundle is wireless, cabo, or FTTH), technology_lvl3 (characterizes if the technology is wireless, cabo, FTTH, FTTH DST, or FTTH OUTROS) and the user interface (user_interface) available in each bundle.

Technology_id	Technology_zone	Technology_lvl1	Technology_lvl2	Technology_lvl3	User_interface
37	DTH	Wireless	Wireless	Wireless	SEM TV
38	DTH	Wireless	Wireless	Wireless	Wireless TV DT...
39	FTTHDST	Wired	CABO	CABO	IRIS
40	FTTHDST	Wired	CABO	CABO	NAGRA
41	FTTHDST	Wired	CABO	CABO	UMA
42	FTTHDST	Wired	FTTH	FTTH DST	IRIS
43	FTTHDST	Wired	FTTH	FTTH DST	NAGRA
44	FTTHDST	Wired	FTTH	FTTH DST	SEM TV
45	FTTHDST	Wired	FTTH	FTTH DST	UMA
46	FTTHDST	Wired	FTTH	FTTH OUTROS	IRIS
47	FTTHDST	Wired	FTTH	FTTH OUTROS	NAGRA
48	FTTHDST	Wired	FTTH	FTTH OUTROS	UMA
51	FTTHNOS	Wired	CABO	CABO	IRIS
52	FTTHNOS	Wired	CABO	CABO	NAGRA
53	FTTHNOS	Wired	CABO	CABO	SEM TV
54	FTTHNOS	Wired	CABO	CABO	TDC
55	FTTHNOS	Wired	CABO	CABO	UMA
56	FTTHNOS	Wired	FTTH	FTTH	IRIS
57	FTTHNOS	Wired	FTTH	FTTH	NAGRA
58	FTTHNOS	Wired	FTTH	FTTH	SEM TV
59	FTTHNOS	Wired	FTTH	FTTH	UMA

Figure 4.4: Technology table

4.1.5 Profile Table

The profile table detains the information referent to each bundle's composition (the available services). It does not only have a variable referring to the combinations of services/products in each bundle (profile_description) but also has variables referring to the number of mobile voice SIM cards (Mobile_voice_SIM_card), mobile internet cards (Mobile_internet_SIM_card) and the number of IOTs (IOT) associated to each bundle.

Profile_id	Profile_description	Mobile_voice_SIM_card	Mobile_internet_SIM_card	IOT
4	2P_IF+IM	0	2	0
5	2P_IF+IM	0	3	0
6	2P_IF+IM	0	4	0
7	2P_IF+IM	0	5	0
8	2P_IF+VM	1	0	0
9	2P_IF+VM	2	0	0
10	2P_IF+VM	3	0	0
11	2P_IF+VM	4	0	0
12	2P_IF+VM	5	0	0
13	2P_IF+VM	6	0	0
14	2P_IF+VM	8	0	0
15	2P_TV+IF	0	0	0

Figure 4.5: Profile table

4.1.6 Customer Table

The customer table is comprised of all the customers that are in the data mart. It comprises the customer's identification number (NIF), which works as the primary key, and the most recent segment in which a customer is inserted (segment). In addition to that, since the customer's segment may change throughout the months, there is also an effective date in this table (it is in number format), indicating when that customer was inserted in that segment. Essentially, the architecture used in this table is a hybrid of the SCD type 1 and 2, having a single line for each client. From SCD type 1, it inherited the substitution of the value in a column by its most recent value. From SCD type 2, it inherited the column effective date.

NIF	Segment	effective_date_segment
[redacted]	CONSUMER	202112
	CONSUMER	202210
	CONSUMER	202208
	CONSUMER	202205
	CONSUMER	202301
	CONSUMER	202301
	CONSUMER	202206
	CONSUMER	202208
	CONSUMER	202112
	CONSUMER	202112
	CONSUMER	202112
	CONSUMER	202112

Figure 4.6: Customer table

4.1.7 Customer Pack Table

The customer pack table is the table that consumed the most time in the ETL process, as most of the fields in it had to go through some of the most time-consuming transformation steps.

This table details all the distinct bundles that exist in the data mart. It not only allows the understanding of how many and which bundles are in it, but it also allows the user to know from which-to-which period each bundle was active and how they relate to each other.

Starting with the customer_pack_id, this is the primary key of this table, to generate this id, it was necessary to determine when a bundle ”started” and ”ended”. The criteria to determine these happenings were a change in the bundle's attributes (not including the segment) and if there was a ”break” in the bundle's existence. For example, regarding the first criteria, if a bundle changed a technology attribute (e.g., wireless to wired), this is recognized as there being two different bundles in the data mart, so both have two distinct customer_pack_ids. To understand the last criteria, it is necessary to recognize that a bundle can be temporarily deactivated and, posteriorly, reactivated. When this happens, in the data mart, it will be considered that these are two distinct bundles with different customer_pack_ids. Instead of attributing new ids for when reactivations happen, other alternatives were considered, such as creating two new columns with dates, one referring to the deactivation date and another indicating the reactivation date. However, this implied that each bundle could only be deactivated and reactivated once in its lifetime. To fix this issue, new columns could be added, each referring to its own deactivation and reactivation date. If this was applied, an indefinite number of columns would be added to the data mart over its lifetime, making these features too complex to handle.

Another id that is very useful in this table is the relation_id. This id allows the relation of bundles to their predecessors between months. With this key, the bundles that were ”disconnected” from each other can again be ”reconnected”. Its creation was based on four keys from the source data warehouse, which purpose were to relate the services between months. To make

use of these keys, a priority level, based on their reliability, was defined, and then a successive relation between months was done. The priority level means that the relation, at first, had to be done with one key. Then, the bundles that did not have compatibility between months with that key were related by a second key, which process was repeated for the last two keys. At the end of this process, the bundles that did not have a relation to a bundle in the previous month were considered to be new bundles (gross adds), and those with no relation to a bundle in the next month were considered to be churned bundles (churn). This procedure was repeated for all the months, being it possible to identify relations between bundles in two consecutive months. After all the months were related, a key (relation_id) was generated with auto increment. However, this id can still be improved.

The next step to improve the relation_id (to make it closer to being perfect) would have been to try and relate months that were not consecutive because of the reactivation problem. Since the relationship was only done between consecutive months if a time "break" happens in a bundle's existence, it will not be able to relate to the most recent bundle to a previous one. To do this, there was just the need to normalize (make equal) the relation_ids of the separate bundles by more than one month while doing the same to the associated bundles. However, as reactivations are not something this department deals with in their daily tasks, there were still some questions on further defining reactivation. Reactivation can be defined as having an upper month limit or no limit. This upper limit means that, even though a bundle has the same attributes as a previous one (technology, NIF, ...), it cannot be related to another bundle that is more than a fixed number of months apart from it. Besides that, there is still the question of which variables should be used to identify a reactivation (should all the variables be considered or just a few). So, as these questions still needed to be answered, the reactivation problem still needed to be fixed in the data mart. Nonetheless, a SAS program to fix this issue was still written. The only details needed for it to be applied are the insertion in the code of the value for the upper month limit and variables to consider in the process. However, even though fixing this problem might be helpful for some analysis, it will not create any issue in the analysis addressed in this work as there is only the need to relate bundles in consecutive months. Nevertheless, one thing to remember is that after this program is implemented, two new flags will appear in the data mart, and two flags mentioned in this section (oldest and most_recent) will need some of their values altered. The first needed flag is called reactivated, which has a value of 1, if the bundle was reactivated; otherwise, it has a value of 0. The other flag is called deactivated and, as the name suggests, has a value of 1, if the bundle suffered deactivation, and 0, if it did not.

Besides the reactivation problem, there was another one SAS raised regarding the relation_id. At first, this key was created with auto-increment. However, SAS Enterprise Guide has a constraint that does not allow similar keys to be in the same table (keys with similar sequences of digits/letters and similar lengths). SAS has this feature that does not allow the user to make correlations within the same table. In this case, the correlation identified was between the customer_pack_id and the relation_id. To avoid this problem, the relation_id was turned into a string with an "r" at the end. This "r" could have been any other letter or symbol chosen arbitrarily.

At the end of this process, it was decided to keep the four keys that originated the relation_id, for data updating purposes. In this data mart, their names were changed to update_id1,

update_id2, update_id3, and update_id4 because of confidentiality reasons.

The rest of the fields that need to be explained in the customer pack table are the effective date; expiration date; most_recent; oldest; product_native; and bundle_type. The effective and expiration dates represent from which period to which period the bundle was in effect (generating revenue), respectively. Both dates are expressed in numbers and not in date format because sometimes it may be needed to compare this value with the time_id. The most_recent is a flag specifying which is the most recent bundle from a series of bundles related by the relation_id (if there is a 1 in the field, it means the bundle is the most recent; otherwise, it is 0). The flag named oldest represents the oldest bundle existing in the series of bundles related to the relation_id (if there is a 1 in the field, it means the bundle is the oldest of its series; otherwise, it is 0). Even though this flag (oldest) was not mentioned in the methodology, it was added, so the identification of gross adds was made easier, as it is going to be explained in section 5.2.2.

The last flag in this table refers to the native offer of the product (if there is a 1 in this field, it means the bundle is professional, else it is residential). The last field (bundle_type) explains the bundle type associated with each bundle if it is either standard or custom-made.

Besides these, all the remaining fields in this dimension represent the foreign keys used to connect with the other dimensions.

Additionally, as it was indicated in the methodology, there were supposed to be some dates in this data mart, such as the activation_date and the loyalty_end_date. However, after some analysis, it was noticed that the quality of the used data source was not good enough to allow them into the data mart. Some of the issues encountered were nonexistent dates in some of the bundles, loyalty end periods that happened before the loyalty start period, loyalty periods that were too small to be considered correct, and some other issues. Therefore, it was not recommended to add them to the final product.

Product_native	Relation_id	effective_date	expiration_date	most_recent	oldest
0	000001636999r	202112	202201	1	1
0	000001133476r	202112	202201	0	1
0	000000723679r	202112	202201	0	1
0	000000725270r	202112	202201	0	1
0	000001261825r	202112	202201	0	1
0	000001589979r	202112	202201	0	1
0	000001519562r	202112	202201	0	1
0	000001643019r	202112	202201	0	1
0	000000786057r	202112	202201	0	1
0	000000624371r	202112	202201	0	1
0	000000104558r	202112	202201	0	1
0	000000104526r	202112	202201	0	1
0	000001644817r	202112	202201	0	1
0	000000209939r	202112	202201	0	1
0	000001632705r	202112	202201	0	1

Figure 4.7: Section of the Customer Pack table

4.2 Fact Table

The fact table will be tackled since the dimension tables were all addressed. The fact table is the simplest of the tables in this data mart. It has the revenues field, which detains the revenue generated by each bundle in each period, and two foreign ids (time_id, and customer_pack_id), that help identify the origin of that revenue by establishing a connection with the time table and the customer pack table.

Customer_pack_id	time_id	Revenues
774475	202201	[redacted]
774475	202204	
1925757	202212	
1925757	202210	
1925757	202211	
774475	202202	
774475	202203	
774475	202112	
774475	202206	
1925757	202207	
1925757	202208	
774475	202205	

Figure 4.8: Fact table

4.3 Updating Strategy

As this data mart has a significant amount of data, it was opted to define an updating strategy that would go in accordance with this amount of data. To compensate for the time that would take to process it, the chosen updating strategy is based on an incremental and warehouse-driven approach; that is, only new or changed data is updated periodically (every month a new month is introduced).

At first, the month's data that will be introduced into the data mart needs to be extracted, cleaned, and transformed. After that, the information about the new month and year introduced needs to be inserted into the time table.

The next update step is relating the bundles from the new month to the ones from the last month in the data mart via the updating ids (to get the relation_id) and via the bundle characteristics (to create the customer_pack_id). Having the bundles related, a mechanism verifies what happened to the bundles. From this mechanism, two main groups can be identified: the bundles that did not change attributes through months and those that did. From the first group (bundles that did not suffer alterations), it is only necessary to add the necessary information to the fact table (time_id, customer_pack_id, and revenues). In the group of bundles that changed attributes, there can be identified three subgroups: the churned bundles (bundles with no relation

to the most recent month), the new/gross add bundles (bundles with no relation to the previous month), and the bundles that changed attributes through months.

The churned bundles are the simplest to update. It is only needed to update the expiration_date, alter its value to be the newly introduced year-month combination, and set the revenue in the fact table for that month to 0. The two other groups need more work to be inserted into the data mart. Both can influence any of the dimensions, unlike the previous groups that could only influence the customer pack table. The new and altered bundles' customer_pack_ids are inserted into the data mart and may add a new combination of attributes that does not already exist. For example, if NOS introduces a new technology to the market, customers can subscribe to that technology, thus influencing the structure of the technology table by adding more rows to it. With this in mind, it is necessary to check if these new bundles have a new combination of attributes in any of the dimensions. If not, it is only attributed to the bundle, the necessary foreign keys to connect with the dimensions. Otherwise, the new combination of attributes is inserted into the respective dimension, and a new primary key is created with auto-increment (the auto-increment starts at one number above the highest number in that dimension), being, then, this id, attributed to the respective bundle(s), as a foreign key. In the case of the customer segment, since it uses a different SCD type, it is verified if any customer has changed its segment, and if it did, the customer segment and segment_effective_date are updated. After this, the updating process varies between groups. The new bundles have the oldest and most_recent flags set to 1; their effective_dates are set to the newly introduced month-year combination; and their relation_id is created like it was explained in section 4.1.7 (the auto-increment starts at one above the highest number in this table and then the resulting number is transformed into a string). The changed bundles inherit their relation_id; have their effective_date set to the newly introduced month/year combination while altering the expiration_date of their correspondents in the previous period to the same value; and have their most_recent flags set to 1, while their related, from the previous month, have this flag set to 0.

After all, this was done, the revenues, time_id, and customer_pack_id of these bundles (new bundles and bundles that changed attributes through months) were added to the fact table. If the reactivation problem correction had been implemented, it would have been applied as the final step.

Chapter 5

Dashboard and Analysis

This part of the work will focus on explaining the dashboard components and how they work. On top of that, through this chapter, the data loaded into the data mart will be analyzed so that conclusions can be taken.

Before starting the description and analysis of the dashboard, it should be mentioned that to protect the confidentiality of NOS information, the values on any graphics will appear in a percentage of the total. In only two of the charts (the two top charts of the Overview page), this percentage value is calculated automatically by changing the output setting of the variable under analysis from ”No calculation” to ”Percent of grand total”. In the other cases, a measure that calculated the percentage values had to be created as some problems arose with the ”Percent of grand total” setting (e.g., values were reversing from positive to negative and vice versa). As it will be seen, most pages and visualizations will only allow the comparison of revenues between two months, so only some months will be analyzed with some depth. One last thing that should be said is that December 2021 and January 2023 were just introduced in Power BI to allow the creation of the next month’s churn and the previous month’s gross adds for all the months, so they will not be analyzed.

5.1 Dashboard’s Aim

As mentioned, a dashboard is a common data visualization technique used in BI and decision-making (Yigitbasioglu & Velcu, 2012). One of the main priorities in defining a dashboard is to identify its aim, as explained by Staron et al. (2015). So, before starting the building of the dashboard, the main questions for the problem had to be defined. Some of the most important questions to create this dashboard were: ”How do the number of bundles and their revenue evolve through the months?”; ”What is the impact of the bundles’ fluctuations (gross adds and churn) in the revenue?”; ”What are the variables (the changes in bundles’ attributes) that influenced the revenue the most?”.

Knowing the questions that needed to be answered, the dashboard could start being created.

5.2 Dashboard Structure

The dashboard was built in Power BI, as it was mentioned in section 3.3. In this program, the data (all the tables mentioned in 4) had to, firstly, be loaded into it, and then the connections between the imported tables had to be organized. Even though this last step can be done manually, it was not, because Power BI can figure out the necessary connections between the tables and implement them (even defining the correct type of connections). Sometimes, adjustments must be made to the schema determined by Power BI, as this program may not have sufficient information to define all the right connections between tables. Therefore, the schema implemented by Power BI was still revised, but no alterations were needed. The resulting schema can be visualized in Figure 5.1.

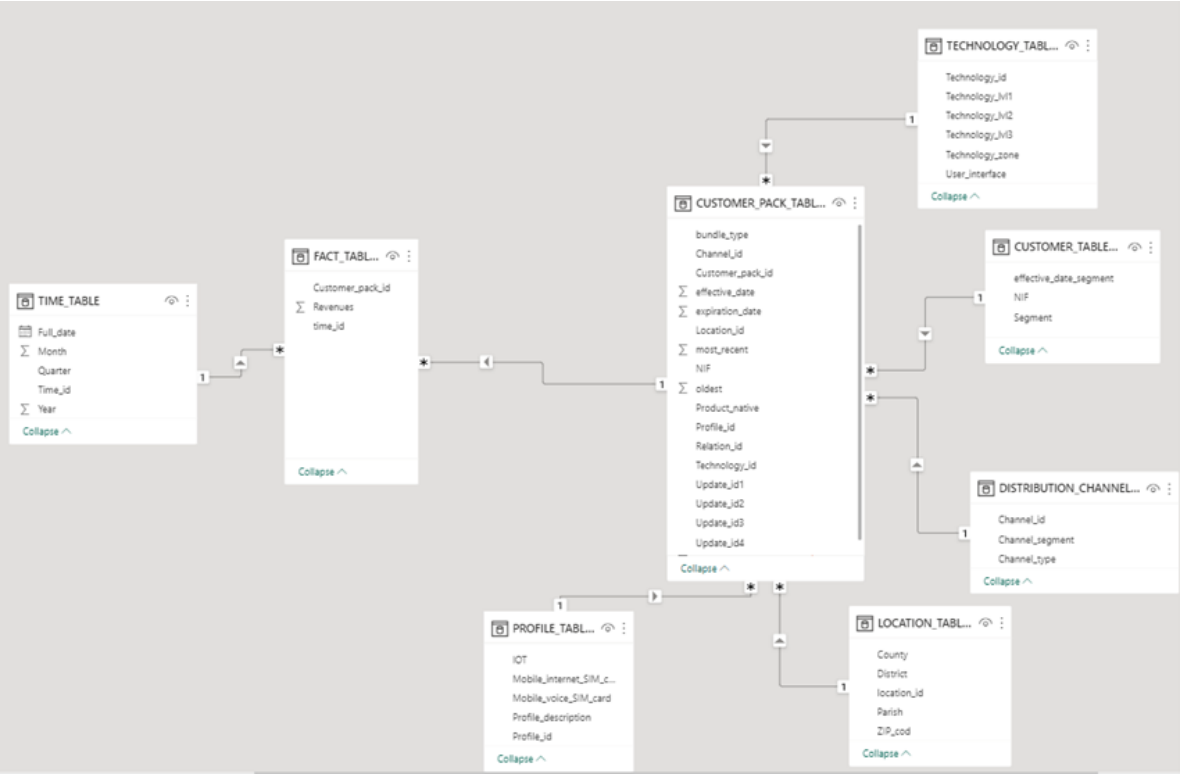

Figure 5.1: Data mart final schema

After this last step, the dashboard was created, resulting in a dashboard with seven pages: the Introduction page, an Overview page, the Decompose page, and four additional "auxiliary" pages (Detailed Overview, Profile & Technology, Profile, and Technology) that were built to help deepen the analysis of the revenue.

Each page was thought to be as flexible, user-friendly, and interactable as possible without penalizing the analysis' understandability.

To make this dashboard, it was necessary to create multiple measures and new fields via DAX (Power BI's programming language), which will be explained in the next sections.

To not be too repetitive through the explanation of the multiple pages some features of the dashboard should be explained right from the start. There are arrows at the bottom of the dashboard's pages that allow easier navigation through the dashboard. All the arrows direct the user to the closest "main" page (Introduction, Overview, and Decompose), which is in that direction on the Power BI's tab (Figure 5.2). For example, the left-pointing arrow on the Detailed Overview will direct the user to the overview, and the right arrow will direct he/she to the Decompose page. Further, on most "auxiliary" pages, there is a button called Reset which is based on a Power BI's bookmark, that allows the user to reset all the filters on that page. Finally, it is possible, on most pages, to cross-filter the graphics (if the user clicks on a figure, such as a bar in a graph, some other graphs are filtered to only take into account observations with that value) or cross-highlight them (similar to cross-filtering but related figures in other graphs become highlighted instead of filtered).

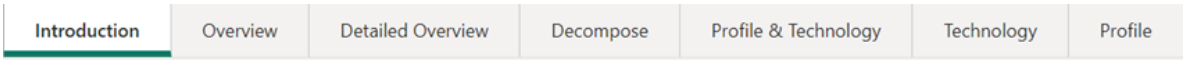

Figure 5.2: Power BI's tab

5.2.1 Introduction Page

The Introduction page has an explanation of the different components in the pages and how they work, also has some advice on how the user can utilize the dashboard to its fullest potential.

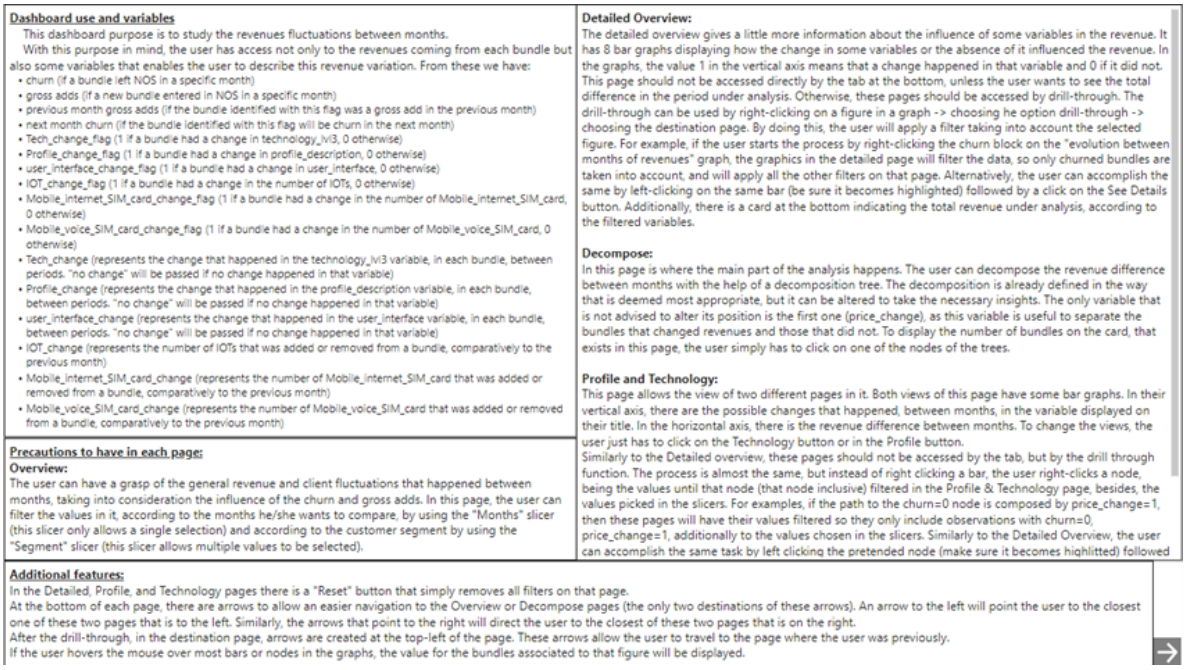

Figure 5.3: Introduction page

5.2.2 Overview Page

The Overview page, which can be seen in Figure 5.4, is essentially a summary of the revenue fluctuations, helping to answer the questions: ”How do the number of bundles and their revenue evolve through the months?”; and, ”What is the impact of the bundles’ fluctuations (gross adds and churn) in the revenue?”. To answer these questions, some graphs show the most basic bundle movements between the months and how they influenced the revenues. This page has a line chart demonstrating the evolution of the number of active bundles through the months and a bar chart illustrating the evolution of their revenues. As the revenue changes between months are proportionally small (big in absolute terms but small when compared to the whole monthly revenue), it was opted to raise the minimum value on the y-axis of the bar chart, so the variations in revenue were more visually noticeable.

An aspect that would have been useful in these charts was a visible distinction between the revenue on each customer segment (in the bar chart) and the number of bundles for each segment (in the line chart). An attempt to use a stacked bar chart to reach the first objective was made, but since the y-axis minimum values were altered, the segment with the highest amount of revenue seemed, visually, to be the one with the least. Therefore, the only other alternative found was to create two new measures (one for each segment) that outputted each segment’s total revenue. These measures were then used as tooltips in the bar chart. These tooltips can be accessed by hovering the mouse over each bar. Doing so will display a black box with some values, including these two. Even though this problem did not exist for the line graph (there could have been added two additional lines, one for each segment), for consistency’s sake, a similar tooltip was applied to the line chart with the bundle count for each segment.

On this page, there are also two waterfall charts. To make these waterfall charts, the design of some measures and calculated fields was necessary. The first one (the value under analysis on the waterfall named ”Evolution between months of bundles”) is called client_variation and is obtained by making the difference between the total gross adds (new bundles in the database in a month) and churn (bundles that left NOS, in the previous month) for the periods under analysis. To calculate the churn and gross adds, some fields were needed in the data warehouse. For a bundle to be churned, the most_recent flag had to be equal to 1, and the expiration_date had to equal the time_id. To create the gross add, a similar thought process was designed. The oldest flag had to be equal to 1, and the expiration_date had to be equal to the time_id. The calculated field that serves as the value under analysis on the chart named ”Evolution between months of revenue” is called revenue difference. This field is obtained by subtracting each bundle’s revenue in one month from its previous month’s revenue.

To describe both of these measures (client_variation and revenue difference), new fields had to be created in both charts, as well. The first two are variants of the previously mentioned churn and gross add flags. The only difference between these fields and their corresponding flags is the output. Instead of their output being 1 or 0, their output is the name of the variable (”Churn” or ”Gross add”) if the observation corresponds to a churn or gross add, otherwise, their output is an empty string. Additionally, it was also necessary a variable called price_change (indicates if a change in the revenue happened in a bundle). This variable was created by comparing the revenue in the bundles, in a month, with their revenue in the previous month, outputting ”Price

changed" if the revenue changed between months or an empty string if it did not. In some of the next pages, it was needed the design of a flag that had the same meaning as this one (outputs 1 when revenues were different and 0 otherwise).

A filter called "Months", which only allows a single selection, was added to enable the user to analyze different months pairings. On this page, there is also a filter for the customers' segments, so an analysis for each segment or a combination of them (this filter does not have a restriction for a single selection) can be made. Furthermore, these slicers are synced with those with the same name on the Decompose page. This was defined so the user does not have the values filtered differently on both pages, thus avoiding the chance of analysis for different periods being made on different pages.

The button called "See Details" will be explained in section 5.2.3.

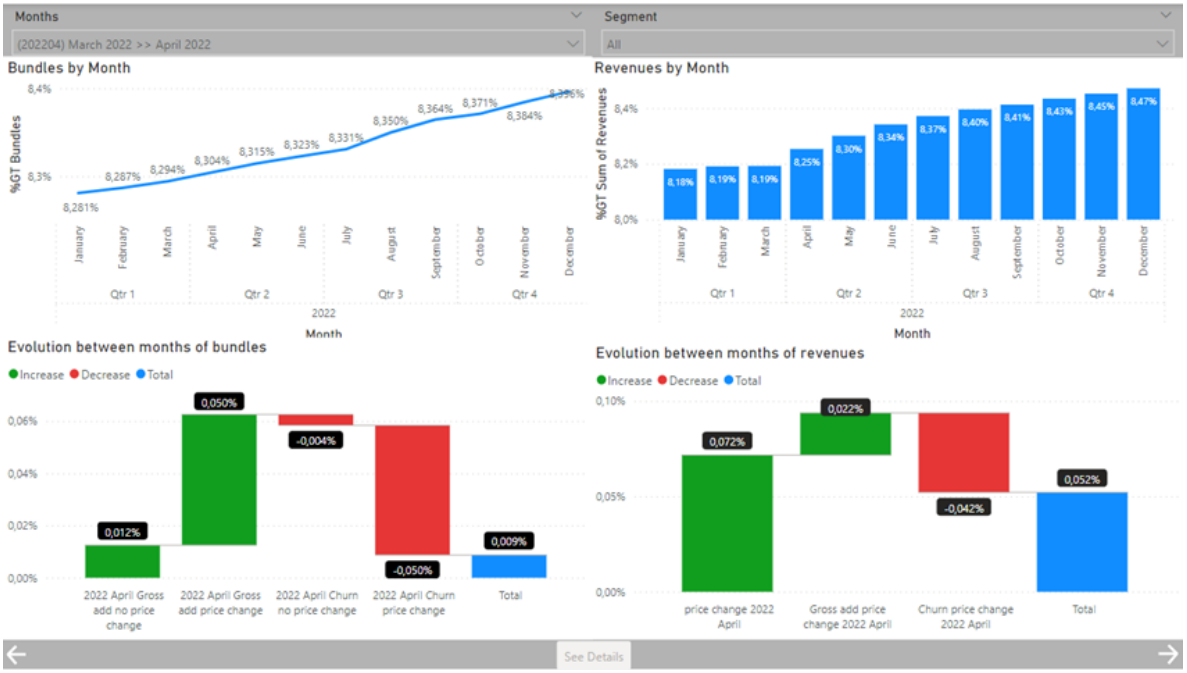

Figure 5.4: Overview page for March » April

Analysis

Starting with the Overview page, the user can see that there is a tendency for the number of bundles and the revenues to grow through 2022, being the first month's revenue representing 8.18% of the total yearly revenue while the last month's revenue represents 8.47% of the total. However, the revenue growth is only partially proportional to the growth in bundles. For example, the revenue growth between March and April and April and May are higher than the others (percentage increased by 0.06 percentile points and 0.05 percentile points, respectively, within each period). However, the increase in the number of bundles is similar, if not lower, in these months, than in others (the increases represent 0.01 percentile points more of the total bundle count than their preceding periods). Since these two periods (between March and April and April

and May) have higher revenue growth values, they will be two of the main focus of this analysis. It is known a priori the primary reason for a higher revenue increase between March and May is due to an annual price increase, but it still should be interesting to analyze other factors that may have affected this increase. Another reason to study these periods is to verify if an analysis done with this dashboard can identify a revenue change influenced by a price increase. Other periods that could be suitable to analyze are the period with the least revenue growth (February » March) and the period with the highest bundle growth (July » August).

Turning the attention towards the waterfall charts, the following analysis can be done by using the period March » April as an example. It can be seen in Figure 5.4 that the type of bundles that impacted the most the evolution of the number of bundles between months are the gross adds (the reason why there is an increase in bundles). The biggest part of these bundles is associated with bundles that did not start their existence with 0 revenue (had a price change). Regarding the revenues' waterfall chart, it can be noticed that even though the number of churned bundles is lower than the gross added (even if only the ones that change the price are considered), the revenue is impacted more by the churn than it is impacted by the gross adds (the revenue lost by churn is bigger than the one gained by the gross adds). However, the revenue coming from the bundles that had an alteration in their revenues (and are neither gross adds nor churn) between months more than compensates for the deficit created by the churned bundles. This conclusion can be taken for the months under analysis and most of them. So, turning the attention towards the other proposed periods to understand why there is a higher or lower variation in terms of bundles or revenues, the following conclusions can be taken: in Figure 5.5, it can be seen that the smaller increase in revenue, between February and March, is mainly due to a higher churn (both in terms of revenue and occurrences); the higher revenue growth between April and May is associated to a higher number of bundles that changed their price (Figure 5.6); and, regarding the period with the highest bundle growth, it seems that the reason for the not so big change in its revenue (when compared with the months with the highest growth) is mainly associated to a lower monetary return on the bundles that suffered alterations (Figure 5.7).

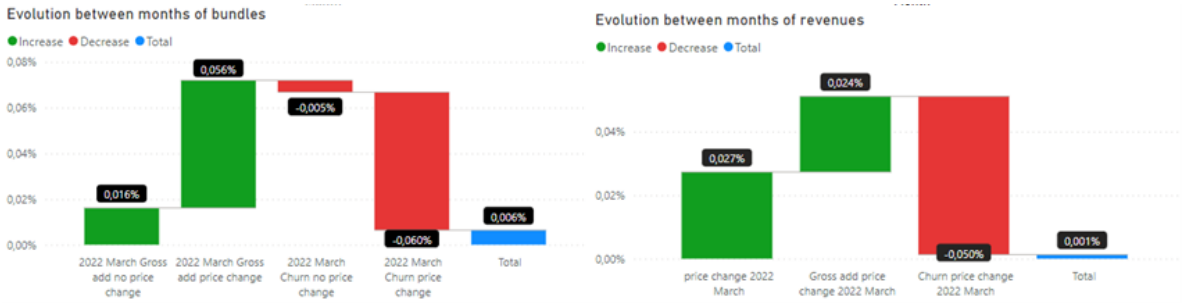

Figure 5.5: Waterfall charts for the period February » March

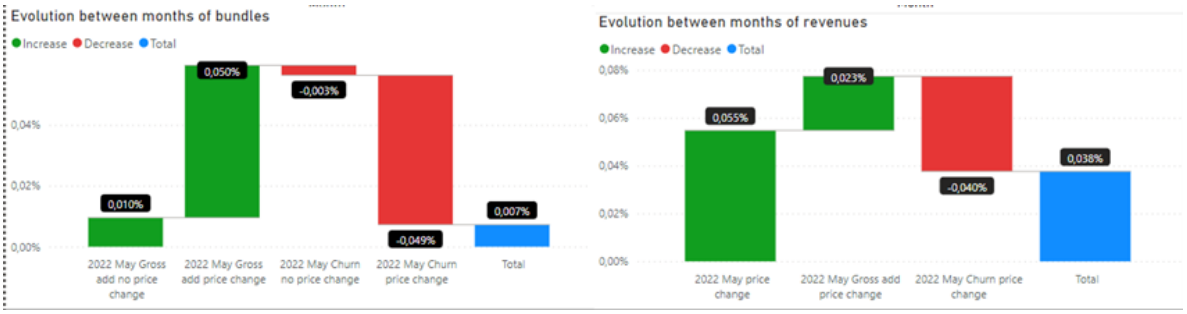

Figure 5.6: Waterfall charts for the period April » May

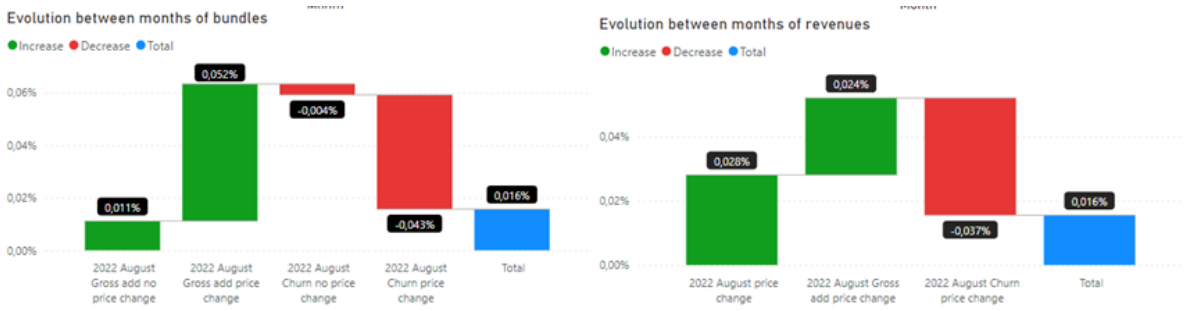

Figure 5.7: Waterfall charts for the period July » August

The last analysis that could be done on this page is relative to the segments. The first insight that can be taken is that the bundle base is mainly composed of consumer bundles, being the business bundles just a minor part of it. The waterfall charts filtered for the consumer segment have a similar analysis as those without a segment filter (the values are not the same, but the conclusions are). Comparing the analysis of the business segment with the one without filters, the most noticeable difference is found in the revenue evolution between January and March (the period with the lowest revenue growth in the none filtered scenario). Between these months, the revenue decreases for the business sector, being it associated with a lower revenue coming from the "price change" bundles (between February and March, this value is negative) and because of a slightly higher churn than usual.

Besides these observations, an attempt was made to identify a pattern between the growth of consumer and business bundles (e.g., if one tends to grow more than the other), but no conclusions could be taken on this matter (at least without a machine learning algorithms).

5.2.3 Detailed Overview Page

The Detailed overview page, which can be seen in Figure 5.8, is where the user may start answering the question: "What are the variables (the changes in bundles' attributes) that influenced the revenue the most?". Eight bar graphs indicate the revenue variation associated with bundles that had a change (or not) in each of the chosen variables (1 indicates that a change happened, and 0 indicates the opposite). Even though there were some variables like county and parish imported into this data mart, not all variables were used for the analysis as they did not have good

explanatory power (this conclusion was taken through trial and error). So, in the final product, it was only created flags for the technology_lvl3 (technology_change_flag), the profile_dsc (profile_change_flag), the user_interface (user_interface_change_flag), the mobile_voice_SIM_card (mobile_voice_SIM_card_change_flag), the mobile_internet_SIM_card (mobile_internet_SIM_card_change_flag), and the IOT (IOT_change_flag) variations. To create all these flags, it was necessary to compare the value in one month to the value in another via an if function. Besides these variables, there are two more: the previous month gross add (if the bundle started existing in the previous month) and the next month churn (if the bundle was churned in the next month). The former flag (the previous month gross add) is important because bundle revenue is proportional to the number of days of the month the bundles existed. Therefore, the revenue coming from a bundle whose existence started in the middle of the month will be less in its first month of existence than in the second. The next month churn is similarly important, as the churn is only accounted for in the month after the bundle deal was terminated; a bundle may contribute less to the revenue preceding its churn month.

To verify the number of bundles in each graph, as explained in section 5.2.2, the user can hover the mouse over each bar, thus displaying the number of bundles. However, it should also be explained that this bundle value is composed of the total number of bundles, including the churned ones, that is, the total distinct bundles in both months, instead of the active monthly bundles. Due to memory issues, the business' bundles and consumers' bundle counts could not be introduced as tooltips. A solution was attempted using stacked bar charts, but even this approach returns memory errors.

To filter the values on this page, the user does not use a slicer as in the Overview, and he/she has to drill through the Overview to filter it. In Power BI and the case of this dashboard, this task is done by right-clicking on one of the bars in the waterfall charts or a value in a slicer, followed by the selection of the drill-through option, which is then followed by the choosing of the target page (the page that the user wants to drill through). By doing so, all the variables on the destination page will filter their values according to the value of the right-clicked bar. Additionally, the drill-through setting called "Keep all filters" was turned on, thus allowing the destination page to inherit all the filters in the drilled-through page. For example, if the user drills through the churn column on the waterfall chart, the values on the destination page will filter its values only to consider observations that have a "churn" value. Alternatively, the user may use the "See Details" button at the bottom of the Overview page to drill through the Overview. The process is similar to the previous one. The user left-clicks on the bar of interest, then clicks the "See Details" button, thus directing him/her to the required page. Even though this last approach may be simpler, it does not allow the drill-through to be directly applied to a slicer. Nonetheless, in both cases, at the top-left of the destination page, an arrow appears after the drill-through operation is done. This arrow directs the user to the page where he/she was previously on (the drilled-through page).

At the bottom of the Detailed Overview page is a card with the total revenue value under analysis for the filtered data, which can be used as a memory aid.

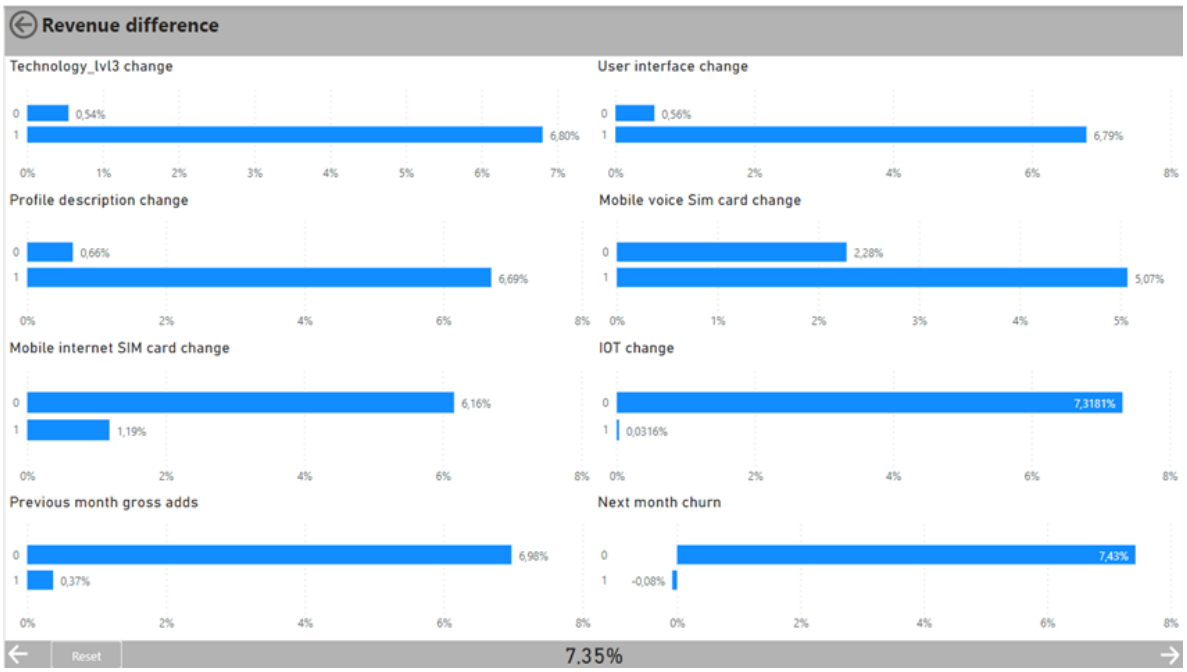

Figure 5.8: Detailed overview page

Analysis

There are multiple visual options on this page, as the user can drill through any of the blocks in the waterfall chart or any of the slicers on the Overview page. However, the most interesting block to analyze is the ”Evolution between months of the revenues” block in the graph, related to the bundles that had their price changed, but neither are churn nor gross adds. By applying the drill-through to this block, for the March » April period and for the April » May period, the views in Figures 5.9 and 5.10 are seen. From these figures, it can be understood that a change in something in a bundle (for the considered variables) decreases the revenue from it. Besides this, for both periods, it can be seen that the biggest contributor to revenue growth (only including the bars represented by 1, and excluding the gross adds and churn) is attributed to when bundles are gross adds in the previous period. Regarding the changes in bundles' characteristics, the most impactful seems to be the change in profile description and mobile voice SIM cards. By applying cross-filtering to the profile description change graph (clicking on the bar represented by 1), for the March » April period, it seems that a change in profile description is normally associated with a change in mobile voice sim cards (77.71% of the changes in profile are associated to changes in this variable). A change in profile is also somewhat related to a change in internet cards (46.53 % of the changes in profile are associated with changes in this variable for the March » April period). Even though these analyses are specific to only the mentioned period, similar conclusions are taken for the rest of the periods (the values may differ a bit).

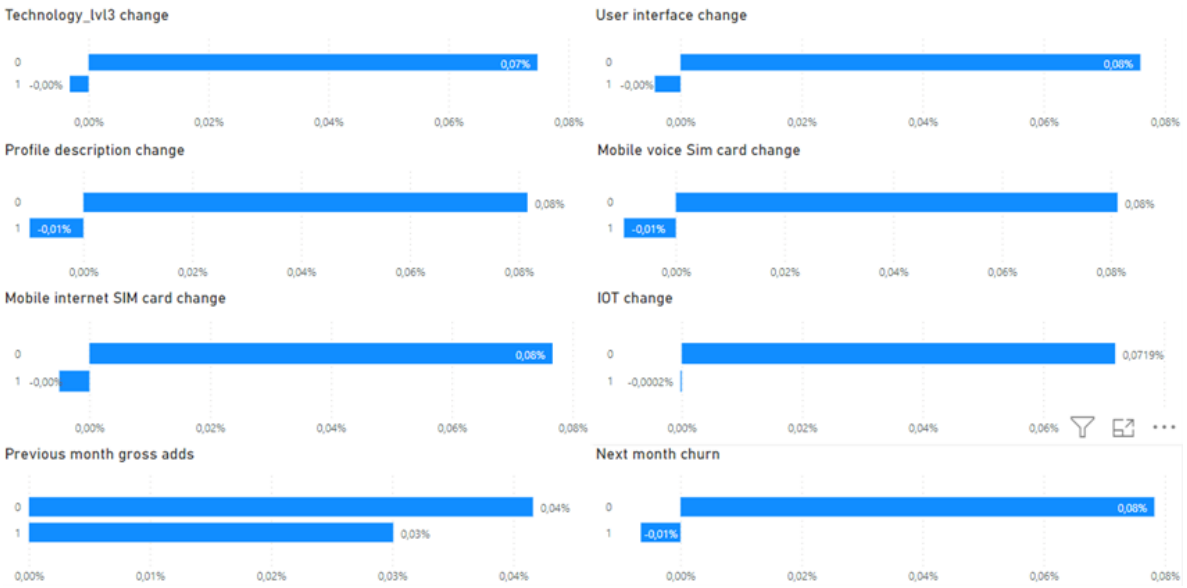

Figure 5.9: Detailed overview view for a drill-through made on the price changed block, in the Evolution between months of revenues chart for the period March » April

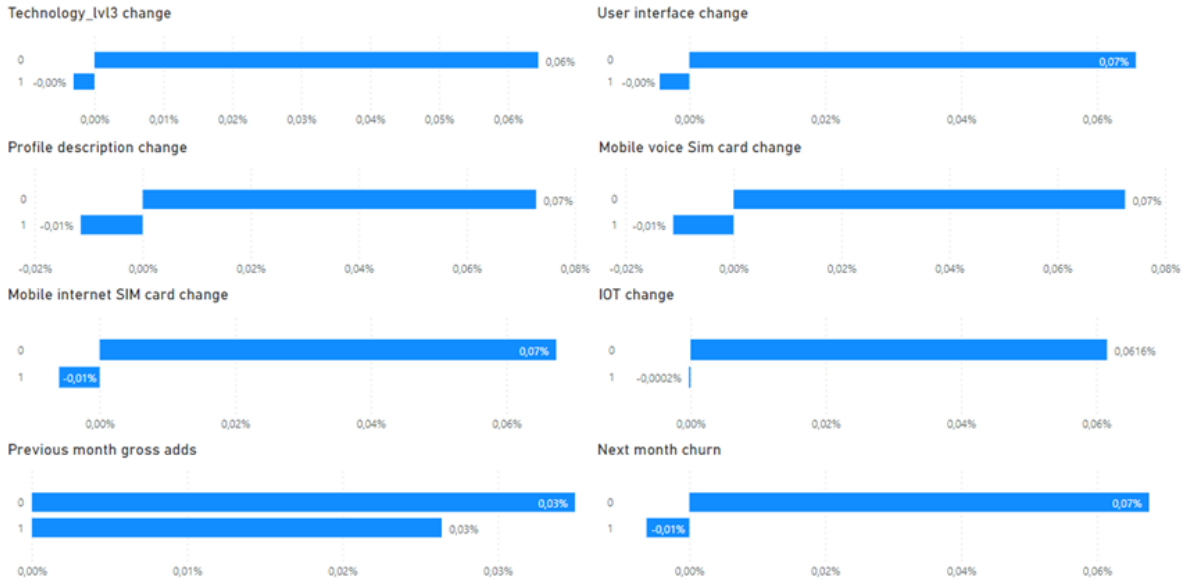

Figure 5.10: Detailed overview view for a drill-through made on the price changed block, in the Evolution between months of revenues chart for the period April » May

Lastly, the variable with the least explanatory power is the IOT change, normally having a minimal impact on the revenue (affected the revenue in about -0.0002 % in the April » May and March » April periods).

Regarding the business segment, for the period between January and February and the one between February and March, it can be seen that when a change happens in a variable, there is a

higher-than-usual relative decrease in revenue in most variables.

5.2.4 Decompose Page

The Decompose page is where the main part of the analysis happens. On this page, and in the pages that will be explained next, the user can complement the answer given to the question: "What are the variables (the changes in bundles' attributes) that influenced the revenue the most?" in the Detailed Overview page. This is accomplished using a decomposition tree visualization with the revenue difference as the value under analysis. The variables used to decompose the difference in revenue are the flags explained in section 5.2.3, plus the gross add, churn, and price_change flags. In alternative to some of these flags, it was considered to use variables that had the combination for all the possible movements in a variable between months (e.g. DTH » UMA, DTH » FTTH, ...). However, this proved to be too confusing to analyze, as there were a lot of nodes in the tree that had many ramifications.

Additionally, on this page, besides the filters referenced in section 5.2.2, there are two different ways to check the total number of distinct bundles on each node of the tree and the number of bundles in each segment. The first way is by clicking on a node of the tree, allowing the number of bundles to be displayed on the cards on the right side of the page (in the case of this project, the values shown are simply the percentage of each type of bundles, in each node). The other way is by hovering the mouse over the node, as explained in section 5.2.3. The tooltip displayed from this last method will also allow the user to verify the revenue for each bundle segment.

At the bottom of this page, there is a button called "See Details", which has a similar purpose as the "See Details" button on the Overview page (further details will be given in section 5.2.6).

The order of the variables in this tree was decided to be price_change, churn, gross add, previous month gross adds, next month churn, technology_change_flag, user_interface_change_flag, profile_change_flag, mobile_voice_SIM_card_change_flag, mobile_internet_SIM_card_change _flag, and IOT_change_flag. One last thing that should be referenced is that even though the order of the variables in the tree is the established, it can be changed by the user. The only reason for the variable order to be like this is that it was deemed the most interesting and useful by NOS. Another way that was considered to organize them was by using AI splits. This kind of split allows the user to split a node by the variable, which will return the node with the highest or lowest value.

Figure 5.11: Decompose page

Analysis

By using a decomposition tree to analyze the revenue change, for the four periods under analysis (February » March; March » April, April » May, and July » August), with the decided variable order, the following analysis can be done. For starters, it can be verified that, usually, the majority of bundles (about 80%) remain the same in their revenue between the months under analysis. However, this value decreases to about 42 % and 58 % in the periods between March and April and between April and May, respectively. This tree can be used to reinforce some of the answers given in section 5.2.3, like, the variable with the least explanatory power is, again, the IOT change, or the change in profile description, normally, seems to be associated to changes of SIM cards. However, something new can be discovered or at least can be understood more easily. At the end of one of the paths of the decomposition trees (the one composed by price_change=1 and 0 on all the other variables), in every period, it can be identified that there is still a value to be explained, being this node still composed by a big majority of bundles that changed the revenue coming from it. This value is probably associated with some change in bundles' attributes that were not introduced in the data mart, which was thought not to have a big influence on the revenue. Some variables that could explain this value were discovered, but given the time to develop this thesis, they could not be introduced in the final product. Nonetheless, instructions on introducing them into the data mart were given to the P&C employees so that the unexplained revenue change could be described. So, this last value cannot be further addressed in this project. However, it should be mentioned that this node may have a monetary value in some analyses, even if all the explanations for the revenue variation are found. In that analysis, this value would result from a generalized price increase/decrease imposed by NOS, like the one that happened

between March and May. Of course, there is also the alternative for this value to be 0 if NOS decides to introduce, into the model, a variable that represents this price increase.

5.2.5 Profile & Technology Page

Technically speaking, the Profile & Technology page is just one page, as it only occupies a tab space in Power BI, but in reality, it works as two pages. Both page views can be accessed via the buttons at the bottom. These buttons are based on bookmarks which, in the case of this page, allow the hiding of certain visualizations without affecting the data or filters in it. By clicking the button named "Profile", the user will visualize four graphs. Each graphic has bars representing the difference in revenues for the changes in one of the following variables: profile_description, Mobile_voice_SIM_card, Mobile_internet_SIM_card, or IOT. For example, imagine that a client changes from a 1P_TV to a 2P_TV+VF; the value "1P_TV » 2P_TV+VF" will appear on the y-axis of the Profile description change graph, while the revenue variation associated to that bundle will appear on the x-axis. Considering that this analysis is made for more than one bundle, the x-axis will have the total sum of the revenue difference, and the y-axis will have all the changes that happened in that variable expressed in a similar way as the one in the previous example. The creation of the descriptive variables, just explained, follows a similar approach to that of the variables explained in section 5.2.3. To build the variable that identified the change in profile description, a comparison between consecutive months is made, and, if a different value between months is detected in a bundle, a string is created with the value from the previous month followed by " » " plus the value it acquires in the next month, otherwise (the value between months is the same), the value "no change" is passed. In alternative to the value "no change", it was considered to simply use the value the bundle had for that variable in those months. The other graphs, present in the Profile view, instead of having a string, like the one on the Profile description change graph's y-axis, have the numerical difference that existed, between months, of the number of mobile cards (voice and internet) and IOTs.

The Technology view of the page has a similar composition to that of the Profile. In this view, instead of graphs expressing the difference in revenue influenced by variables on the profile table, there are graphs for the difference in revenue influenced by the variation of the technology_lvl3 and by the variation of the user_interface variables.

To filter the values in these pages, similarly, as previously mentioned, the user right-clicks on one of the nodes in the decomposition tree, followed by the selection of the drill-through option, which is then followed by the choosing of the target page (the page that the user wants to drill through). By doing so, all the nodes' labels up until that node (that node inclusive) will serve as filters on the targeted dashboard. For example, if the path leading to the node that the user right-clicked passes through a node where price_change=1, the churn=1, and the gross adds=0, as exemplified in Figure 5.11, the targeted page will have the values of these variables filtered according to those values. Alternatively, as in the Detailed Overview page, the user can drill through with the "See Details" button on the Decompose page. As it can be bothersome to memorize the whole path created by the tree, in addition to the mentioned visualizations, both views also have a multi-row card, specifying the path taken to the drilled-through node and a card with the total revenue under analysis.

To further complement this page, as in some analyses, the charts may show many columns, two slicers were created to mitigate this problem. These slicers, found on the bottom-right of the page, allow the user to filter out ranges of values that the user does not want to analyze. The one on the left (Below) controls the values below it, and the one to the right (Above) controls the range of values above it. For instance, if the user changes the value to -500 in the Below slicer, and the rest remains the same, only bars below -500 and bars with values above 0 will appear. To form these slicers, two new tables were created. Both of these tables have a single column composed of a sequence of numbers achieved by incrementing the previous number in the sequence by 1.

Lastly, it should also be mentioned that this page allows the user to verify the number of bundles by hovering the mouse over a bar in the graphics, as expressed in previous sections.

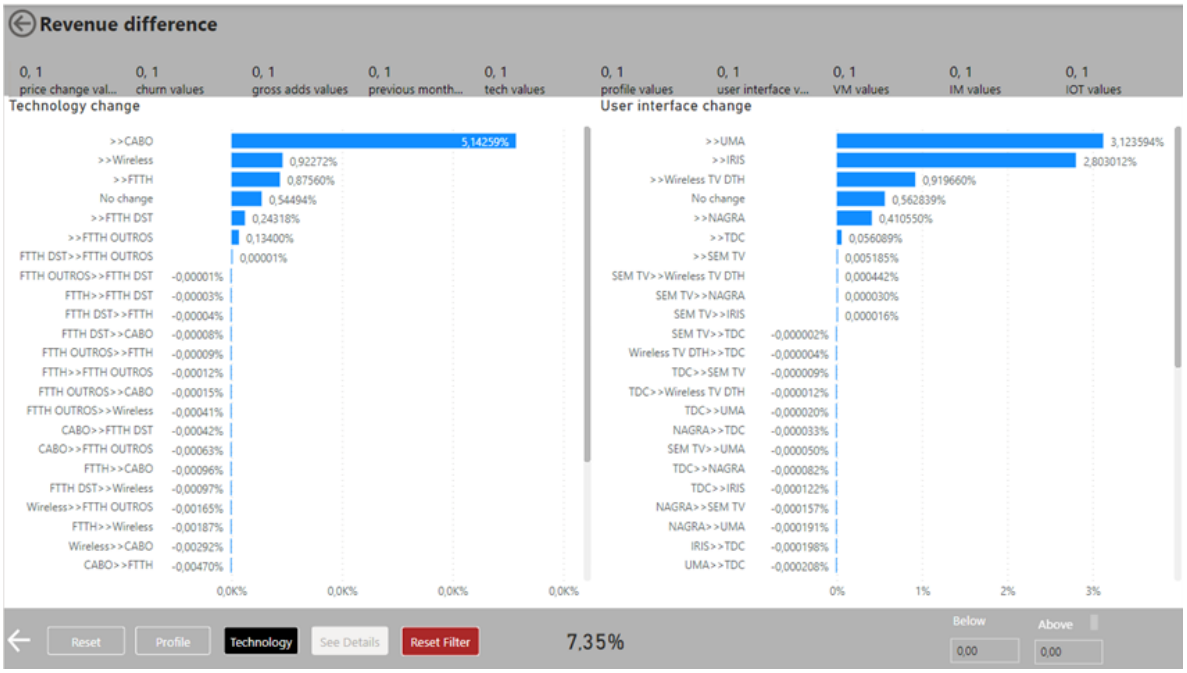

Figure 5.12: Technology View of the Profile & Technology page

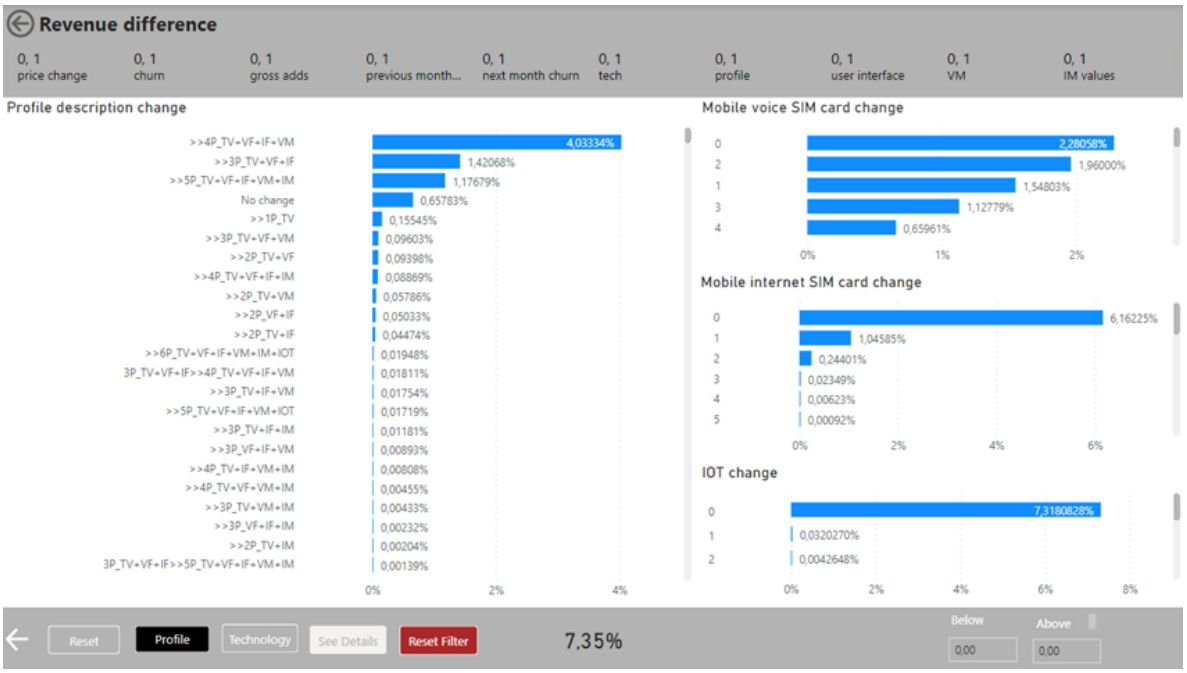

Figure 5.13: Profile View of the Profile & Technology page

Analysis

Now, in this section, it is ideal to find the specific type of bundle variations affecting the revenue.

To eliminate the interference of gross adds, churn, and other measures from the analysis of the profile variables, a drill-through is applied on the node where user_interface_change_flag equals 0 and where all the preceding variables, besides the price_change flag, are equal to 0, as well. Starting with the period with the highest revenue growth (March » April), it is understood that the revenue is affected negatively, mainly by a decrease of 1 mobile voice card in the bundles (this decrease is equivalent to 0.00357% of the total revenue on this month).

On this page, it can also be confirmed that some of the most impactful changes in the profile (changes that increase/decrease the revenue the most) are associated with the total removal or addition of sim cards. For example, the removal of the IM component from the 5P_TV+VF+IF+VM+IM is associated with the second biggest decrease in revenue, while the removal of the VM component from the 4P_TV+VF+IF+VM is associated with the biggest decrease in revenue. In some months, the change associated with the 5P may decrease the revenue more than the one associated with the 4P. Following these two, most of the negative changes that impact the revenues are associated with bundles that have removed one or both of the mobile services, being possible that other services were removed in addition to them. On the other hand, the change that had the biggest contribution to an increase in revenue is always associated with adding voice SIM cards (VM) to a bundle of the type 3P_TV+VF+IF. Reorganizing the tree in a way the tech_change_flag and the user_interface_change_flag are at the end, it is possible to make a drill-through that captures all the movements that occur in both of these variables and are

not influenced by the rest of the changes. Applying this drill-through that ignores the effect of all other types of changes besides the ones in technology and user interface (for March » April), it is noticed (in Figure 5.14) that not many types of changes increase the revenue (only the FTTH DST » FTTH OUTROS does). On the other hand, many changes decrease the revenue, being the lowest revenue difference associated with the change from IRIS to UMA, in all months. In the technology change, no unique category normally impacts the most through the months. So, regarding the periods under analysis, the change with the most impact is the change from Wireless to FTTH DST (for February » March, April » May, and July » August), and CABO » FTTH (for March » April). It was also found that the most impactful change in user interface could be the change from Wireless to FTTH, in some months.

Having analyzed bundles that are neither gross adds nor churn, the rest of this section will be dedicated to analyzing the gross adds, churn, and the business segment. Making the drill-through for the February » March period, on the variable next month churn, in a way that only the churn and the price_change influence are captured until then, the following analysis can be done. Starting by the profile, from Figure 5.16, it can be understood that the lowest values encountered (both for revenues and count of bundles), by far, are the total values associated with the churn of 3P_TV+VF+IF, followed by the values related to the churn of 4P_TV+VF+IF+VM. In other months, the 4P may impact the revenue more. Regarding the technology (Figure 5.17), the most churned type of bundles and with the highest revenue lost are the bundles that use cable (CABO). Relatively to the user interface, the highest revenue lost is attributed to bundles with UMA.

This section also tried to compare this period (the one with the lowest revenue growth) with others to find differences in the type of bundles that affected its revenue the most. However, it was just mostly found some insignificant differences, being the most important difference the higher number of bundles associated with each change (higher overall number of churn).

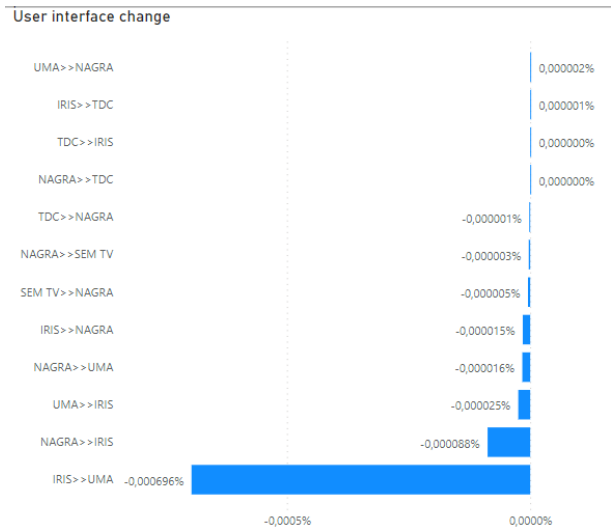

Figure 5.14: User interface bar chart for March » April

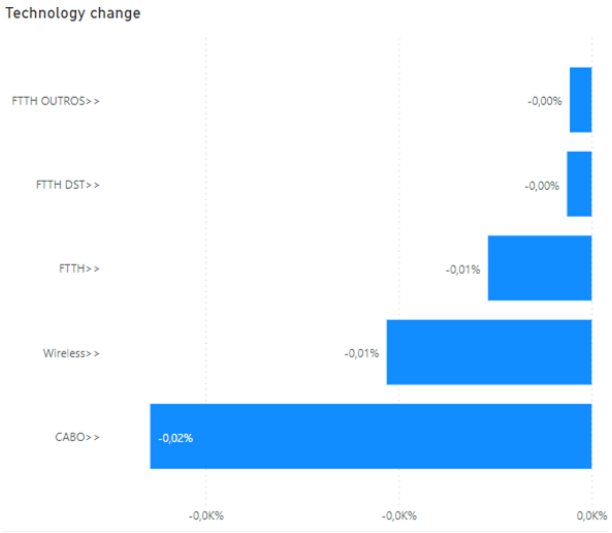

Figure 5.15: Technology bar chart for March » April

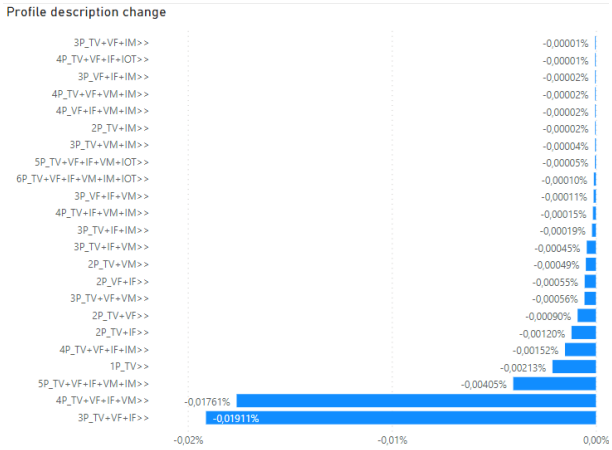

Figure 5.16: Profile description bar chart for February » March

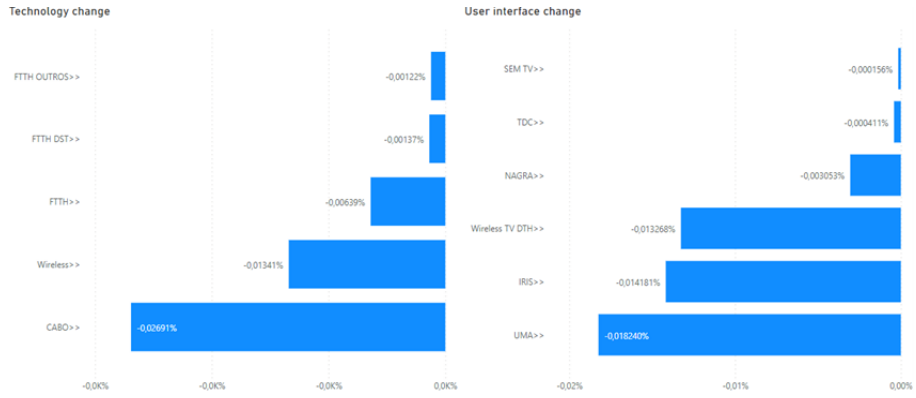

Figure 5.17: Technology and user interface bar charts for February » March

Relatively to the gross adds, usually, the cable is the most prevalent in all months (both in revenues and amount). Following the cable, the highest values are associated with the gross add of bundles with FTTH. So, the gross adds have a ”similar” variation when compared to the churned bundles (the most churned are the most gross added). However, there is a higher churn of cable than gross adds, meaning that this type of technology is decreasing through the months. Contrarily, the FTTH has the opposite behavior (more gross adds than churn), inducing a growth in customers' use of this type of technology. This is to be expected since, at the moment, NOS has more cable installed all over Portugal than FTTH, but is slowly changing the paradigm.

Lastly, regarding the business segment, the conclusions are similar. Therefore, there seems to be an almost constant behavior of what type of changes happen in gross added, churned, or price changed bundles between months.

5.2.6 Profile and Technology Pages

Both the Profile and Technology pages have the same structure as the views with the same names, explained in section 5.2.5, being them almost a ”continuation” of the views with the opposite names. The Profile page has the same format as the Profile view, and the Technology page has the same composition as the Technology view. The only difference between these pages and corresponding views is that they do not have some of the buttons on the bottom-left corner.

These last two pages are used to see more details about the values seen in the graphs of the Profile&Technology page. For example, if the user wants to see the changes in IOTs, within a technology change, such as ”FTTH » DTH”, then he/she just has to drill-through that bar, thus applying filters to the Profile page. Other solutions for this problem were attempted, such as bookmarks in conjunction with cross-filtering or cross-highlighting. Nevertheless, Power BI's bookmarks do not detect that cross-filtering or cross-highlighting are applied when bookmarks are activated.

Nonetheless, because of the implemented solution, something still needed to be added to these pages. As drill-through was passing all the filters from the Profile & Technology page, the filters originated from the Below and Above slicers were also being passed through the pages. To contradict this, a new feature was added to the Profile & Technology page. This new feature

includes a new bookmark that resets the Above and Below slicers. To implement this new bookmark, a new button had to be created on that page, with the name "Reset Filter". To remind the user to click this button before drill-through, this button was painted with a color that draws attention (red).

Analysis

The analysis made on the Profile and Technology pages helps consolidate the conclusion found so far, being its use optional in most analyses. To analyze this page, it would be useful to drill through a path in the decompose page that captures changes from variables that are on different views of the Profile & Technology page. An example could be drilling through the next month's churn node so that only a price_change is detected in the bundles (all the variables in the drill-through until the next month's churn node are 0, besides the price_change). After this drill-through is made, if another drill-through is done to the bar corresponding to 5P_TV+VF+IF+VM+IM » 4P_TV+VF+IF+VM (the profile with the lowest value for the revenue difference for the March » April period), it is seen in Figure 5.18 that the most impactful technology change becomes Wireless » FTTH DST, instead of CABO » FTTH. In regards to the user interface change, the conclusions remain the same. The conclusions are very similar if the drill-through is applied to the second most impactful profile change (4P_TV+VF+IF+VM » 3P_TV+VF+IF). Regarding the rest of the months, the conclusions are identical.

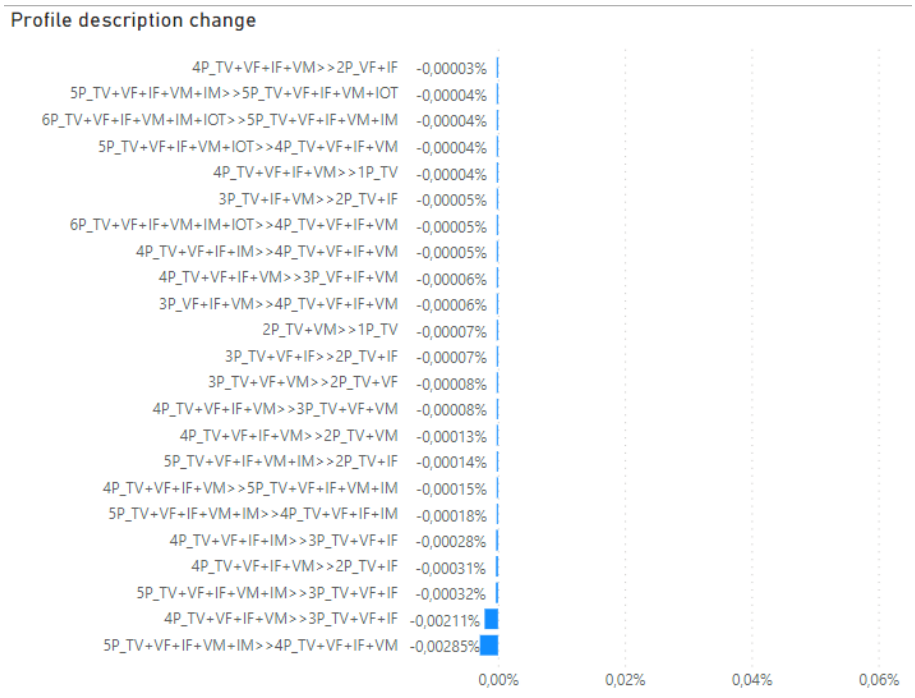

Figure 5.18: Profile bar chart for March April

All in all, the changes between months in the bundles are very similar. Most of the time, if not every, the most impactful differences in revenue are associated with churn, gross adds, or

related variables (previous month gross adds and next month churn), while the changes on the other considered variables tend to impact the revenue, yet in a lesser extent. The profile and SIM cards are the most impactful from the changes on other variables, while the IOTs are the least. Regarding the changes, it looks like the ones that affect the most revenue between months are similar (e.g., the most churned technology is cable), even if they do not always appear in the same order (in some occasions, the most impactful change may be different but will rarely drop from the top 3 most impactful). Nonetheless, even though this dashboard can explain a big part of the revenue, there is still a small part of it that still needs justification.

Chapter 6

Conclusion

This dissertation was written within the scope of a curricular internship at NOS' Planning and Management Control department.

Currently, the study of the bundles' revenue fluctuations through the months is quite troublesome because there is no unified database with the needed detail to help this analysis. This detail not only implied that there needed to be a database with the characteristics of each bundle, but it also implied that there needed to exist a way (a single key), in the database that allowed the relation of bundles to their predecessors (same bundle, in a different period that has different characteristics). As there is no such database to rely on for this analysis, it is difficult to visualize the needed data and draw conclusions on the bundle's alterations/movements that affected the revenue the most. Therefore, the main goals of this project were to build a data mart capable of allowing the study of the bundles' revenue fluctuations and a dashboard that would allow the interpretation of the data in the data mart. These two tools will enable the P&C to do the required analysis.

The data used to develop this dissertation was real data provided by NOS, ranging from December 2021 to January 2023. This data had a lower level of granularity than the one needed for the analysis (the granularity of the data was specific to services, not bundles), so it went through some transformation steps to achieve its final form.

The methodology used in this work followed that of Kimball & Ross (2011). This methodology starts with understanding the business process and the data, followed by the basic design of the schema that will be used in the data mart.

The resulting dimensional model originated a schema with one fact table and seven dimension tables. Two-dimension tables (time table and customer pack table) are directly connected through foreign keys to the fact table, while the other five tables are related to the customer pack table by the same means.

After creating the data mart, it was imported into Power BI, where some calculated fields and measures were designed. From these, some of the most important were: the revenue difference, which calculated the revenue difference of the bundles between months; the flags and variables that indicated if an alteration had been done to a bundle's characteristics; and the previous month's gross add, next month churn, gross add, and churn flags.

When all the measures and fields were finally created, the dashboard was implemented.

The final step of this project passed through the analysis of the information in the dashboard, which led to the discovery of some conclusions. Even though there is room for improvement in terms of additional explanatory variables (a part of revenue variation is still to be explained), the bundles' alterations that affect the most revenue through the months are their churn and gross adds. Following these, the bundles that were gross adds in the previous month normally are the third alteration with the biggest impact on the revenue. Relatively to the changes that happen within a bundle (if the gross add, churn, and related variables are excluded), it is noticed that the most impactful change on the revenue is in the profile. Most of the time, this change is influenced by a decrease in the number of mobile voice cards and, sometimes, by the decrease in internet voice cards.

Even though only some of the variables from the data mart were used, they may be helpful for future projects that need this kind of data.

This project also helped find some minor areas for improvement in the source data warehouse and helped the P&C department improve their knowledge about the data referent to the bundles. Moreover, it also helped this department improve its knowledge of BI analytical tools, as Power BI is not very used in this department. To confirm the benefits that this project brought to NOS, the reader can go through the feedback letter in appendix A.1.

In future works, a deeper study of the variables that may influence the revenue is proposed to complement and improve the analysis done in this work. After this improvement, it would be advisable to make this data mart available to the whole company and not only the P&C department.

Bibliography

Ali, S. M., Gupta, N., Nayak, G. K., & Lenka, R. K. (2016). Big data visualization: Tools and challenges. In *2016 2nd international conference on contemporary computing and informatics (ic3i)* (pp. 656–660).

Al-Ruithe, M., Benkhelifa, E., & Hameed, K. (2019). A systematic literature review of data governance and cloud data governance. *Personal and Ubiquitous Computing*, *23*(5), 839–859.

Ballou, D. P., & Pazer, H. L. (1985). Modeling data and process quality in multi-input, multi-output information systems. *Management science*, *31*(2), 150–162.

Başaran, B. P. (2005). *A comparison of data warehouse design models* (Unpublished doctoral dissertation). Atilim University.

Becker, L. T., & Gould, E. M. (2019). Microsoft power bi: extending excel to manipulate, analyze, and visualize diverse data. *Serials Review*, *45*(3), 184–188.

Benkhaled, H. N., & Berrabah, D. (2019). Data quality management for data warehouse systems: State of the art. *JERI*.

Bonifati, A., Cattaneo, F., Fuggetta, A., & Paraboschi, S. (2001, 10). Designing data marts for data warehouses. *ACM Trans. Softw. Eng. Methodol.*, *10*, 452-483. doi: 10.1145/384189.384190

Brajković, H., Jakšić, D., & Poščić, P. (2020). Data warehouse and data quality-an overview. In *Central european conference on information and intelligent systems* (pp. 17–24).

Brous, P., Janssen, M., & Krans, R. (2020). Data governance as success factor for data science. In *Conference on e-business, e-services and e-society* (pp. 431–442).

Butt, E. M. A., Quadri, S., & Zaman, E. M. (2012). Star schema implementation for automation of examination records. In *Proceedings of the international conference on frontiers in education: computer science and computer engineering (fecs)* (p. 1).

Calvanese, D., Dragone, L., Nardi, D., Rosati, R., & Trisolini, S. M. (2006). Enterprise modeling and data warehousing in telecom italia. *Information Systems*, *31*(1), 1-32. Retrieved from https://www.sciencedirect.com/science/article/pii/S0306437904000730 doi: https://doi.org/10.1016/j.is.2004.07.002

Carrasco, M. (2011). Visual attention: The past 25 years. *Vision Research*, *51*(13), 1484-1525. Retrieved from https://www.sciencedirect.com/science/article/pii/S0042698911001544 (Vision Research 50th Anniversary Issue: Part 2) doi: https://doi.org/10.1016/j.visres.2011.04.012

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and olap technology. *ACM Sigmod record*, *26*(1), 65–74.

Chawla, G., Bamal, S., & Khatana, R. (2018). Big data analytics for data visualization: Review of techniques. *International Journal of Computer Applications*, *182*(21), 37–40.

Draheim, D. (2013). Towards total budgeting and the interactive budget warehouse. In F. Piazolo & M. Felderer (Eds.), *Innovation and future of enterprise information systems* (pp. 271–286). Berlin, Heidelberg: Springer Berlin Heidelberg.

Fatusha, D., & Ktona, A. (2016). Big data in telco. In *Rta-csit* (pp. 116–120).

Freiler, W., Matkovic, K., & Hauser, H. (2008). Interactive visual analysis of set-typed data. *IEEE Transactions on Visualization and Computer Graphics*, *14*(6), 1340-1347. doi: 10.1109/TVCG.2008.144

Geiger, J. G. (2004). Data quality management, the most critical initiative you can implement. *Data Warehousing, Management and Quality, Paper*, 098–29.

Golab, L. (2013). Data warehouse quality: Summary and outlook. In S. Sadiq (Ed.), *Handbook of data quality: Research and practice* (pp. 121–140). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-36257-6_6 doi: 10.1007/978-3-642-36257-6_6

Gonzales, M. L., Bagchi, K., Udo, G., & Kirs, P. (2011). Diffusion of business intelligence and data warehousing: An exploratory investigation of research and practice. In *2011 44th hawaii international conference on system sciences* (pp. 1–9).

Grasse, D., & Nelson, G. (2006). Base sas® vs. sas® data integration studio: Understanding etl and the sas tools used to support it. *SAS Users Group International*.

Gupta, H., Harinarayan, V., Rajaraman, A., & Ullman, J. D. (1997). Index selection for olap. In *Proceedings 13th international conference on data engineering* (pp. 208–219).

Habul, A., & Pilav-Velic, A. (2010). Business intelligence and customer relationship management. In *Proceedings of the iti 2010, 32nd international conference on information technology interfaces* (p. 169-174).

Hazen, B. T., Boone, C. A., Ezell, J. D., & Jones-Farmer, L. A. (2014). Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. *International Journal of Production Economics*, *154*, 72–80.

Heer, J., & Shneiderman, B. (2012). Interactive dynamics for visual analysis. *Communications of the ACM*, *55*(4), 45–54.

Holzinger, A. (2013). Human-computer interaction and knowledge discovery (hci-kdd): What is the benefit of bringing those two fields to work together? In A. Cuzzocrea, C. Kittl, D. E. Simos, E. Weippl, & L. Xu (Eds.), *Availability, reliability, and security in information systems and hci* (pp. 319–328). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ibrahim, A. E. A., Elamer, A. A., & Ezat, A. N. (2021). The convergence of big data and accounting: innovative research opportunities. *Technological Forecasting and Social Change*, *173*, 121171. Retrieved from https://www.sciencedirect.com/science/article/pii/S0040162521006041 doi: https://doi.org/10.1016/j.techfore.2021.121171

Inmon, W. H. (1996). The data warehouse and data mining. *Communications of the ACM*, *39*(11), 49–51.

International, D. (2017). *Dama-dmbok: Data management body of knowledge (2nd edition)*. Denville, NJ, USA: Technics Publications, LLC.

Islam, M., & Jin, S. (2019). An overview of data visualization. In *2019 international conference on information science and communications technologies (icisct)* (pp. 1–7).

Kazi, Z., Radulovic, B., Radovanovic, D., & Kazi, L. (2010). Molap data warehouse of a software products servicing call center. In *The 33rd international convention mipro* (pp. 1283–1287).

Kelton, A. S., Pennington, R. R., & Tuttle, B. M. (2010). The effects of information presentation format on judgment and decision making: A review of the information systems research. *Journal of Information Systems*, *24*(2), 79–105.

Kimball, R., & Ross, M. (2011). *The data warehouse toolkit: the complete guide to dimensional modeling*. John Wiley & Sons.

Konyha, Z., Matkovic, K., & Hauser, H. (2009). Interactive visual analysis in engineering: A survey. *Posters at SCCG*, *2009*, 31–38.

Kwasowiec, W., & Karwowski, W. (n.d.). Data warehouse design. *INFORMATION SYSTEMS IN MANAGEMENT IV* , 46.

Köppen, V., Winsemann, T., & Saake, G. (2015, 06). An analytical model for data persistence in business data warehouses. *Proceedings - International Conference on Research Challenges in Information Science*, *2015*, 351-362. doi: 10.1109/RCIS.2015.7128896

Loshin, D. (2009). Chapter 5 - data quality and mdm. In D. Loshin (Ed.), *Master data management* (p. 87-103). Boston: Morgan Kaufmann. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780123742254000059 doi: https://doi.org/10.1016/B978-0-12-374225-4.00005-9

Mannino, M. V., & Walter, Z. (2006). A framework for data warehouse refresh policies. *Decision Support Systems*, *42*(1), 121–143.

Martyn, T. (2004). Reconsidering multi-dimensional schemas. *ACM Sigmod Record*, *33*(1), 83–88.

McCaig, M., & Rezania, D. (2021). A scoping review on data governance. *Available at SSRN 3882450*.

Moalla, I., Nabli, A., Bouzguenda, L., & Hammami, M. (2017). Data warehouse design approaches from social media: review and comparison. *Social Network Analysis and Mining*, *7*(1), 1–14.

Moody, D. L., & Kortink, M. A. (2000). From enterprise models to dimensional models: a methodology for data warehouse and data mart design. In *Dmdw* (p. 5).

Moscoso-Zea, O., Paredes-Gualtor, J., & Luján-Mora, S. (2018). A holistic view of data warehousing in education. *IEEE access*, *6*, 64659–64673.

Mosley, M., Brackett, M. H., Earley, S., & Henderson, D. (2010). *Dama guide to the data management body of knowledge*. Technics Publications.

Moura, S. V. (2012). *Rolap e molap: Comparação e avaliação prática* (Unpublished doctoral dissertation). Universidade Nova de Lisboa.

Neff, A., Schosser, M., Zelt, S., Uebernickel, F., & Brenner, W. (2013). Explicating performance impacts of it governance and data governance in multi-business organisations. In (p. 1 - 11). Melbourne, Australia: RMIT University.

Negrut, V. (2018). Power bi: Effective data aggregation. *Quaestus*, *13*, 146–152.

Pedamkar, P. (2022). *HOLAP | Architecture of HOLAP with Advantages & Disadvantages — educba.com*. https://www.educba.com/holap/. ([Accessed 19-Jan-2023])

Qiongwei, Y., Yaotang, L., & Qiuyun, N. (2012, 12). Research on the application model of business intelligence (bi) in e-business from the perspective of chinese culture. In (p. 907-911). doi: 10.1109/ICCSNT.2012.6526074

Rainardi, V. (2008). *Building a data warehouse*. Apress.

Ramasamy, A., & Chowdhury, S. (2020). Big data quality dimensions: a systematic literature review. *JISTEM-Journal of Information Systems and Technology Management*, *17*.

Reddy, S. S. S., Lavanya, A., Khanna, V., & Reddy, L. (2009). Research issues on data warehouse maintenance. In *2009 international conference on advanced computer control* (p. 623-627). doi: 10 .1109/ICACC.2009.144

Redman, T. C. (2001). *Data quality: the field guide*. Digital press.

Redman, T. C. (2012). Data quality management past, present, and future: Towards a management system for data. In *Handbook of data quality* (pp. 15–40). Springer.

Sachinopoulou, A. (2001). *Multidimensional visualization*. VTT Technical Research Centre of Finland.

SAS. (2017). *Delivering the power of sas® analytics and reporting from an easy-to-use, point-and-click windows interface* (Tech. Rep.). Author.

SAS. (2022). *SAS Help Center — documentation.sas.com.* https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/proc/p13kvtl8ezj13in17i6m99jypcwi.htm. ([Accessed 14-Dec-2022])

Scherr, M. (2008). Multiple and coordinated views in information visualization. *Trends in information visualization*, *38*, 1–33.

Schwendimann, B. A., Rodriguez-Triana, M. J., Vozniuk, A., Prieto, L. P., Boroujeni, M. S., Holzer, A., … Dillenbourg, P. (2016). Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Transactions on Learning Technologies*, *10*(1), 30–41.

Sebastian-Coleman, L. (2012). *Measuring data quality for ongoing improvement: a data quality assessment framework*. Newnes.

Sen, A., & Sinha, A. P. (2005, mar). A comparison of data warehousing methodologies. *Commun. ACM*, *48*(3), 79–84. Retrieved from https://doi.org/10.1145/1047671.1047673 doi: 10.1145/1047671.1047673

Sethi, M. (2012). Data warehousing and olap technology. *International Journal of Engineering Research and Applications (IJERA)*, *2*(2), 955–960.

Shin, S., Park, G., Lee, W., & Lee, S. (1998). Case study: how to make telecom pricing strategy using data warehouse approach. In *Proceedings of the thirty-first hawaii international conference on system sciences* (Vol. 6, p. 55-60 vol.6). doi: 10.1109/HICSS.1998.654758

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2008). System concepts.

Silwattananusarn, T., & Tuamsuk, K. (2012, 10). Data mining and its applications for knowledge management: A literature review from 2007 to 2012. *International Journal of Data Mining & Knowledge Management Process*, *2*. doi: 10.5121/ijdkp.2012.2502

Simplilearn. (2023). *What is Auto Increment in SQL and How to Set Up Auto Increment — simplilearn.com.* https://www.simplilearn.com/tutorials/sql-tutorial/auto-increment-in-sql. ([Accessed 17-Apr-2023])

Singh, V., & Ghate, A. (2017). A review: Analysis on data warehousing and data mining. *Int. J. Res. Dev. Technol*, *8*(1), 336–341.

Staron, M. (2015). Dashboard development guide how to build sustainable and useful dashboards to support software development and maintenance..

Staron, M., Niesel, K., & Meding, W. (2015). Selecting the right visualization of indicators and measures–dashboard selection model. In *Software measurement* (pp. 130–143). Springer.

Sumathi, S., & Sivanandam, S. (2006). Data marts and data warehouse. *Introduction to Data Mining and its Applications*, 75–150.

Talwar, K., & Gosain, A. (2012). Hierarchy classification for data warehouse: A survey. *Procedia Technology*, *6*, 460–468.

Thackeray, R., Neiger, B., Hanson, C., & McKenzie, J. (2008, 11). Enhancing promotional strategies within social marketing programs: Use of web 2.0 social media. *Health promotion practice*, *9*, 338-43. doi: 10.1177/1524839908325335

Tomar, R., Parsad, R., & Jaggi, S. (2010). *Sas enterprise guide: an overview* (Tech. Rep.). SAS.

Toreini, P., & Morana, S. (2017). Designing attention-aware business intelligence and analytics dashboards. In *Designing the digital transformation: Desrist 2017 research in progress proceedings of the 12th international conference on design science research in information systems and technology. karlsruhe, germany. 30 may-1 jun.* (pp. 64–72).

Tripathy, A. S., Das, K., & Swarnkar, T. (2011). Pervasive sas techniques for designing a data warehouse for an integrated enterprise : An approach towards business process..

Trisolini, S. M., Lenzerini, M., & Nardi, D. (1999, jun). Data integration and warehousing in telecom italia. *SIGMOD Rec.*, *28*(2), 538–539. Retrieved from https://doi.org/10.1145/304181.304569 doi: 10.1145/304181.304569

Tukey, J. W. (1977). *Exploratory data analysis* (Vol. 2). Reading, MA.

Unwin, A. (2020). Why is data visualization important? what is important in data visualization? *Harvard Data Science Review*, *2*(1), 1.

Vassiliadis, P., Simitsis, A., & Skiadopoulos, S. (2002). Conceptual modeling for etl processes. In *Proceedings of the 5th acm international workshop on data warehousing and olap* (pp. 14–21).

Velicanu, M., & Matei, G. (2007). Building a data warehouse step by step. *Economic Informatics, Forthcoming*.

Viana, M. d. M., & Cabral, P. (2020). How to effectively use interactivity to improve visual analysis in groups of novices or experts. In *Capsi 2020. 20ª conferência da associação portuguesa de sistemas de informação," artificialização, humanização: Os desafios dos sistemas de informação na transformação da sociedade".[20th portuguese association of information systems conference]* (pp. 1–18).

Wang, R. Y., & Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, *12*(4), 5–33.

Yigitbasioglu, O. M., & Velcu, O. (2012). A review of dashboards in performance management: Implications for design and research. *International Journal of Accounting Information Systems*, *13*(1), 41-59. Retrieved from https://www.sciencedirect.com/science/article/pii/S1467089511000443 doi: https://doi.org/10.1016/j.accinf.2011.08.002

Yin, S., & Kaynak, O. (2015). Big data for modern industry: Challenges and trends [point of view]. *Proceedings of the IEEE*, *103*(2), 143-146. doi: 10.1109/JPROC.2015.2388958

Appendix A

Appendix

A.1 Feedback letter

Direção de Planeamento e Controlo de Gestão
NOS Corporate Center

Porto, 30th June 2023

Óscar Afonso
Director of Faculdade de Economia do Porto

Dear Professor,

At this final moment of the project, we would like to give our feedback from the experience with the curricular internship of Cândido Rafael Rocha, within the scope of her master's thesis in Modelling, Data Analysis and Decision Support Systems (MADSAD) at FEP.

The project carried out by Cândido Rafael Rocha is part of our strategic journey of digital and analytics transformation of NOS's Planning and Management Control Department, and we would like to highlight the following messages:

- The project added clear value to the department, by enabling a more detailed and deeper explanation of NOS' revenues evolution, based on two relevant outputs:
    1. Construction of unified data mart of NOS Bundle's revenues, joining the different data sources in a smart and efficient way, and with the key dimensions to enable a more insightful analysis;
    2. Construction of analytical Power BI dashboard, with a very simple and user-friendly approach to the complex analysis of our revenues evolution.
- The success of this challenging project was only possible because of Cândido Rafael Rocha's determination, resilience, natural curiosity, problem solving and critical thinking skills and, foremost, hard work and dedication towards finding the best solution.
- Moreover, I would like to refer that it was a pleasure for me to guide Cândido Rafael Rocha and witness his growth within our team, adding value by challenging "our way of doing things".
- For all the experience and competence shown, Cândido Rafael Rocha was recommended internally to apply to our NOS Alfa program, and we look forward to working with Rafael again in the near future.

Last but not the least, we would like to express our gratitude towards Professors João Gama and Bruno Veloso, who once again for a third consecutive year in this partnership, have allowed us to keep in touch with young talent and state-of-the-art methodologies and academic knowledge, essential to our transformation and innovation journey.

Kind regards,

Cláudia Dias
Head of NOS' TELCO Management Control & Corporate BI