FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Analyzing Chatbots Data with Data Mining

Ana Catarina Dias Amaral

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Carlos Soares

Co-Supervisor: Cláudio Sá

August 2, 2018

Analyzing Chatbots Data with Data Mining

Ana Catarina Dias Amaral

Mestrado Integrado em Engenharia Informática e Computação

August 2, 2018

Abstract

The use of chatbots in business contexts, as a way of communicating with customers is becoming more common nowadays. With this increasing use of machine-to-human contact as a means of connecting with customers, the problem arises of whether communication is being performed correctly. The quality of the questions may not be appropriate, there may be out-of-date questions and even questions that, while correct, may not contribute to the company's goals.

The dialog flow in a chatbot conversation is not homogeneous and there might be a lot of implicit subjectivity. In this way, the analysis of conversations of chatbots is an opportunity to improve the quality of service. However, this task can be quite challenging and time-consuming, so there is a need to find methods to automate it. This information can help to promote the propensity to support the fostering of company sales and satisfaction of their clients.

This dissertation addresses this problem with a combination of two Data Mining topics: Subgroup Discovery and Sequential Pattern Mining. While Sequential Pattern Mining is concerned with finding frequent patterns in sequences, subgroup discovery is the discovery of patterns with unusual behaviour. A chatbot conversation can be represented as a sequence of interactions. In this way, in a context of chatbots, Sequential Pattern Mining can be used to discover sequences of interactions that users go through frequently. In the same context, Subgroup Discovery can be translated as the discovery of interactions between users and bots which are unusual in comparison to the population. By combining these two techniques, it was possible to find frequent unusual sequences, i.e., sequences that reveal unexpected behaviours. These unexpected behaviours can be either positive (i.e. interactions that exceeds expectations) or negative (i.e. interactions below expectations) in terms of business goals.

As a scientific contribution, two different approaches were developed to discover unusual patterns in sequential data. Furthermore, five distinct quality measures were also created. In addition to the scientific contribution, the work developed can also benefit organisations that use chatbots to communicate with their partners, such as customers. Chatbot design and marketing teams can use the results obtained to correct failures and implement the best practices found in other areas or components, both at the system level and at the business level. Within this dissertation, the approaches developed allowed the discovery of several interesting behavioural patterns in chatbots users. Most of them corresponded to design errors in the chatbots under study.

Keywords: Chatbots analytics, Data analysis, Pattern mining, Sequential Pattern Mining, Subgroup discovery

Resumo

A utilização de *chatbots* em contextos empresariais, como forma de comunicação entre cliente e empresa, é cada vez mais comum nos dias de hoje. Com o aumento da utilização deste tipo de interação surge o problema de se a comunicação está a ser efetuada corretamente. A qualidade das questões pode não ser a mais apropriada, podem existir questões deprecadas ou mesmo questões que, apesar de estarem corretas, podem não contribuir para os objetivos do negócio.

O fluxo de diálogo numa conversa de *chatbot* não é algo homogéneo e pode existir bastante subjetividade implícita. Desta forma, a análise de conversas de *chatbots* é uma oportunidade para melhorar a qualidade dos serviços. No entanto, esta tarefa pode-se tornar bastante desafiante e demorada, pelo que existe a necessidade de encontrar métodos de automatização da mesma. Este tipo de informações é algo que pode propiciar o progresso de uma organização e, ao mesmo tempo, a satisfação dos seus clientes.

Esta dissertação aborda este problema como uma combinação entre duas áreas de *Data Mining*: *Subgroup Discovery* e *Sequential Pattern Mining*. Enquanto que *Sequential Pattern Mining* se preocupa com a descoberta de padrões frequentes em dados sequenciais, *Subbroup Discovery* tem como propósito a descoberta de padrões que revelem um comportamento fora do comum. Uma conversa de *chatbot* pode ser representada como uma sequência de interações. Desta forma, num contexto de *chatbots*, *Sequential Pattern Mining* pode ser usado para descobrir sequências de interações que os utilizadores atravessem frequentemente. No mesmo contexto, *Subgroup Discovery* pode ser traduzido como a descoberta de interações entre utilizador e bot consideradas fora do comum quando comparadas com a população. A partir da combinação destas duas abordagens, será possivel encontrar sequências não usuais, ou seja, sequências que revelam comportamentos inesperados. Estes comportamentos podem ser tanto positivos (i.e. interações que excedem as expetativas) como negativos (i.e. interações que ficaram aquém do esperado) em termos de objetivos de negócio.

Como contribuição científica, foram desenvolvidas duas abordagens para descobrir padrões com um comportamento fora do comum em dados sequenciais. Para além disso, foram também criadas cinco medidas de qualidade. O trabalho desenvolvido, para além da contribuição científica, pode também beneficiar organizações que façam uso de *chatbots* como forma de comunicação com os seus parceiros, como clientes. Equipas de *design* de *chatbots* e equipas de *marketing* poderão usar os resultados obtidos para corrigir falhas e extender as melhores práticas encontradas a outras áreas ou componentes, tanto ao nível do sistema como do negócio. No âmbito desta dissertação, as abordagens desenvolvidas possibilitaram a descoberta de vários padrões comportamentais interessantes por parte de utilizadores de chatbots. A maior parte destes padrões corresponderam a erros de *design* nos *chatbots* sob estudo.

Acknowledgements

I would like to start this section by stating my gratitude to my supervisors, Carlos Soares and Cláudio Rebelo. Thanks for all the advice, for the many helpful comments and for the meetings that lasted more than an hour. Thanks also for the ones that did not last so long.

I would also like to thank all my friends, colleagues and teachers who worked with me and helped me throughout my academic path.

Finally, and most importantly, I would like to thank my dear family and my beloved boyfriend for withstanding my frequent absences and, above all, for the unconditional support throughout this journey.

Catarina Amaral

“Be curious.
And however difficult life may seem, there is always something you can do.
It matters that you don’t give up.”

Stephen Hawking

Contents

List of Figures

LIST OF FIGURES

List of Tables

Abbreviations

DM	Data Mining
PM	Pattern Mining
SM	Sequence Mining
SPM	Sequential Pattern Mining
SD	Subgroup Discovery
EMM	Exceptional Model Mining
BFS	Breadth-First Search
DFS	Depth-First Search
IDE	Integrated Development Environment

Chapter 1

Introduction

Chatbots, or conversational agents, have been used in a variety of contexts. They have provided a natural language interface to their users with increased sophisticated design [Kerly et al., 2007]. Commerce, entertainment [Shawar and Atwell, 2007a], education [Jia, 2003], security training [Kowalski et al., 2009] and public sector [McNeal and Newyear, 2013] are some examples where the use of chatbots has already been adopted. Chatbots are computer programs designed to simulate conversation with human users using hearing or textual methods. The first chatbot developed was ELIZA in 1966 [Weizenbaum, 1966]. ELIZA analysed the input phrases and returned an output based on reassembly rules associated with the decomposition of the input provided. Its algorithm did not keep the interactions in memory and so it was not possible to develop some type of collaboration or negotiation. However, this algorithm was the first that produced a feeling of concern with users, as it was the first natural language processing computer program. Since ELIZA, many chatbots have been created. Some aimed at improving other algorithms, others developed from scratch. They are often integrated into multi-purpose dialogue systems, including consumer service, information acquisition or even language learning [Fryer and Carpenter, 2006].

In this dissertation, the chatbots taken into account are inserted in a business context. The use of chatbots in business contexts, as a way of communicating with customers, is becoming more common nowadays [Anwar and Abulaish, 2014b]. With this proliferation of machine-to-human contact and increase of complexity in dialogues, the question arises of whether communication is being performed correctly. A better understanding of customer needs and system performance is something that must be streamlined. This can lead to a better achievement of business goals and costumer needs.

The process of creating a chatbot encompasses the design, building, and monitoring of the system. However, many companies avoid the last phase or use poor analysis methods (e.g. number of users and how long a user takes during a session) that do not allow them to do an in-depth analysis and study of their chatbot.

Data mining is one of the scientific areas that can be used to address these types of problems. In particular, in this dissertation a data mining approach is used for the analysis of conversation data from chatbots.

1.1 Motivations and Objectives

The most successful chatbots are constantly adapting and revising their conversation flows in response to their users [Shawar and Atwell, 2007b]. In this way, the analysis of chatbots' conversations is an opportunity to improve the quality of services. However, building a bot that provide a good user experience during dialog is known to be a challenging task [AlHagbani and Khan, 2016]. The dialog flow in a chatbot conversation is not homogeneous (i.e. the flow of interactions of different users may not be the same, since users do not always have the same behaviour in a conversation) and there is a lot of implicit subjectivity. The quality of the questions may not be appropriate or the desired one, there may be out-of-date questions and even questions that, while correct, may not contribute to the company goals and costumer needs. Therefore, understanding the improvements that must be made, or identify the best practices that should be generalized, is of great importance. However, this task can be quite challenging and time-consuming, since the processing and analysis of large amounts of data can become quite complex. Therefore, there is a need to find methods to automate it.

The main objective of this dissertation is the development of an algorithm that allows the discovery of unexpected user behaviours in chatbot data. The flow of a chatbot conversation is represented as a sequence of questions and answers. In this way, a conversation is understood as a system consisting of questions and answers where the questions are asked by the bot and the answers given by the client. This dissertation addresses this problem as a combination of Subgroup Discovery and Sequential Pattern Mining techniques. While Sequential Pattern Mining is concerned with finding frequent patterns in sequences, Subgroup Discovery is the discovery of patterns with unusual behaviour. With this in mind, the purpose of this dissertation is to create a Subgroup Discovery algorithm, which allows the discovery of unexpected user behaviour. A user behaviour can be represented by a set of interactions. In this way, a behaviour can be interpreted as a pattern of interactions.

An unexpected behaviour can be either positive (i.e. interactions that exceeds expectations) or negative (i.e. interactions below expectations) in terms of business goals. A subgroup is evaluated according to the deviation from mean reference values. For example, a subgroup that has a higher than usual proportion of provided email address can be considered a positive subgroup. In the same context, a subgroup that has a proportion of provided email address lower than the average of the reference population can be considered a negative subgroup. The purpose of this dissertation is to find both types of subgroups.

From the business point of view, the main objective of this dissertation is to improve the interactions between company and client in a context of chatbots. This improvement is based on the behavioural patterns of chatbots users. The results from this dissertation can help foster the company's sales and improve customer satisfaction, which can be translated into business success. On the one hand, the results obtained from the negative patterns can be used as decision support for corrections of system failures. On the other hand, the positive patterns found can lead to the discovery of best practices that can be further extended to other components or business areas.

These discoveries can lead companies to achieve, in a more precise and fast way, the increase of their sales and the improvement of the satisfaction of their customers.

1.2 Structure of the Dissertation

The present dissertation is subdivided into six chapters, beginning with the current chapter, the introduction, describing the context, motivation and objectives.

Chapter 2 starts with the definition of a chatbot and a presentation of the different types of analytics in this field. Then, it introduces the concepts of Sequential Pattern Mining and Subgroup Discovery.

In chapter 3 the design and implementation of the various approaches developed to solve this problem are presented and described. This chapter also contains the different measures used to evaluate patterns as unusual or not. Finally, a preliminary analysis of the results with artificial data is presented.

Chapter 4 presents the experimental setup used to obtain and evaluate the results. Some implementation details are also mentioned.

In chapter 5 the results obtained within this dissertation are described and analysed.

In the last chapter, Chapter 6, the conclusions reached throughout this dissertation are described. Finally, the contributions of this project and some possibilities for future work are also presented.

Chapter 2

Literature Review and Background

With the increase in the amount of data available, the interest and need for data mining and data analytics techniques is also growing.

Data Mining (DM) is the extraction of implicit, potentially useful, and previously unknown knowledge from data. It involves the development of computational systems capable of retrieving regularities and patterns from data. This collection process can be automatic or semiautomatic [Witten et al., 2017]. DM applications focus on knowledge acquisition and future predictions based on past data. The domain of these systems is quite extensive, both in scientific and business terms [Kaur and Wasan, 2006].

Analytics is known as the area of discovery, interpretation and communication of patterns in data. Analytics forms an important part of Business Intelligence, which converts raw data into useful information. It is related to solving problems in business fields, enterprise decision management, sales force sizing and optimisation, price and promotion modelling, etc. The methodological foundations for analytics are statistics, mathematics, data mining, programming and operations research, as well as data visualisation in order to communicate insights learned to the relevant stakeholders. There are several branches or domains of analytics. Some examples are marketing analytics, people analytics, risk analytics and web analytics [El-Nasr et al., 2016].

From the definitions of both areas, it is possible to perceive that they are essentially equivalent terms. However, analytics is more encompassing and more popular in business than in academia.

This chapter presents the state of the art on the topics covered in this dissertation. The concept of chatbot and the type of analysis that exists nowadays are also presented. Finally, some Pattern Mining themes related to this dissertation are addressed, namely Sequential Pattern Mining and Subgroup Discovery.

2.1 Chatbots

A chatbot is known as a conversational agent that interacts with users using natural language [Shawar and Atwell, 2003]. This interaction is performed taking into account a particular context or topic and can be done using visual and textual commands.

The first chatbot, short for the term *chatterbot*, was developed in the 1960s with the aim of impersonating a human and trying to deceive users about the identity of who would be interacting with them: a human or a machine [Shawar and Atwell, 2007a]. Nowadays, chatbots are created for other purposes. Chatbots are present in several significant areas, as referred in Chapter 1.

Nowadays, chatbots often start the dialogue by making themselves available on websites. Usually, the conversation with a chatbot is initiated by a first approach made by the user. This can be the sending of a message, the click of a button, among others. This first approach indicates the intention of communication with the bot and triggers a response or comment of the bot, which consequently leads to the beginning of a dialog [Huang et al., 2007].

2.1.1 Chatbots Analytics

Chatbots can help brands maintain a high-quality omnichannel presence by guiding customers to the right information. Bot analytics can give marketers information they need to improve the customer experience by improving chatbot performance. Businesses are increasingly using chatbots as conversation interfaces to customers, so the value of chatbots analytics tools is likely to rise. Chatbots analytics is the area of analytics that addresses the problems more closely related to this dissertation. Analytics applied to chatbots is an area where there is not much research known so far. The most common type of statistics used for chatbots analysis are the number of users and how long a user takes during a session. It is an area where the first steps are still being taken, but it has been growing over time. A few analytics tools for chatbots are starting to emerge [CMSWire, 2018].

Recently, a new chatbot developed by Google was released. In November 2017, this company launched Chatbase [Google, 2017], a dedicated chatbot analytics platform. This new tool allows visualisation of the flow of conversations to understand how effectively users interact with the bot. This company has also led the analytics charge years ago with the *Google Analytics* tool. The news was shared by the media.[1] Chatbase allows visualisation of the flow of conversations to understand how efficiently users interact with the bot. It also provides information like number of daily sessions, daily sessions per user, sessions per user and user messages per session. Other metrics are the percentage of users who were present at the time of a given request, and agent response time, representing how long it took the bot to respond. In this way, the evaluation metrics in Chatbase are basically focused on usage volume and response times.

The evaluation metrics used so far do not address behavioural analyses regarding a specific interaction. In addition, the metrics that currently exist also do not allow understanding the behavioural patterns of chatbots users based on their interactions with the bot. Measures that make it possible to discover unusual patterns in chatbots systems is something that is also missing. With this in mind, it is possible to realise that there is a gap in the existing analyses for chatbots.

[1] https://www.cmswire.com/digital-experience/google-chatbase-ushers-in-the-rise-of-chatbot-analytics/

In this dissertation the analytics applied to chatbots are based on their interactions. In the proposed approach, the behavioural analysis of users is made from the discovery of interesting patterns in the interactions between bot and users.

2.2 Pattern Mining

Pattern mining (PM) is a topic of Data Mining which focuses on the discovery of interesting, useful, and unexpected patterns in the data. The interest in pattern mining techniques comes from the ability to discover hidden patterns in large databases that are interpretable by humans and which prove to be useful for understanding data and for making decisions [Fournier-viger and Lin, 2017].

PM has become very popular due to its applications in several fields. PM algorithms can be applied to various data types such as sequence databases, transaction databases [Fournier-viger and Lin, 2017], graphs [Anwar and Abulaish, 2014a], World Wide Web databases [Cooley et al., 1997], among others. The topics of PM that will be addressed in this dissertation are:

- **Sequential Pattern Mining** - Frequent Pattern Mining is concerned with finding patterns with a frequency of occurrence greater than or equal to a certain value. Frequent patterns are item sets that appear in data with a frequency of occurrence not less than a user-specified threshold [Han et al., 2007]. Sequential pattern mining is concerned with the discovery of frequent patterns in sequential databases [Chand et al., 2012]. Data in the form of sequences can be found in many fields such as text analysis, market basket analysis, webpage click-stream analysis, bioinformatics, among others [Fournier-viger and Lin, 2017] (Chapter 2.3).

- **Subgroup Discovery** - Subgroup Discovery is concerned with finding rules describing subsets of the population that are sufficiently large and statistically unusual [Lavrač et al., 2004b] (Chapter 2.4).

2.3 Sequential Pattern Mining

Sequence Data Mining was first introduced in 1995 in the context of market analysis [Agrawal and Srikant, 1995]. This area can be translated as Data Mining applied to sequences. In this way, the purpose of Sequence Mining problems is to get knowledge from sequential data.

Sequential Pattern Mining has emerged as the intersection between Pattern Mining and Sequence Data Mining. It is a technique that aims to find interesting sequential patterns among large databases. Similarly to Frequent Pattern Mining, the patterns obtained are subsequences that occur with a frequency not less than a user-defined threshold [Chand et al., 2012].

Below, the fundamental concepts of the Sequential Pattern Mining problem, required for the understanding of this dissertation, are presented [Boghey and Singh, 2013] [Chand et al., 2012]. Table 2.1 serves as an illustrative example for the following definitions.

Table 2.1: Example of a sequential database [Fournier-viger and Lin, 2017].

Sequence ID	**Sequence**
1	$<\{A,B\},\{C\},\{F,G\},\{G\},\{E\}>$
2	$<\{A,B\},\{C\},\{B\},\{A,B,E,F\}>$
3	$<\{A\},\{B\},\{F,G\},\{E\}>$
4	$<\{B\},\{F,G\}>$

Let us consider $I = \{I_1,...,I_m\}$ a set of m distinct attributes called items. An *itemset* is defined as a non-empty subset of items and an *itemset* with m elements is called a *m-itemset*. For example, in a supermarket sales context, considering that $I = \{A,B,C,D,E,F,G\}$ represents all products in a supermarket, the set $\{A,B\}$ represents an itemset with 2 products (items) of I. This itemset can represent a transaction.

A sequence is an ordered list of *itemsets*. A sequence S with length l is defined as $< s_1, s_2, ..., s_l >$, where each element of the sequence S (s_i) is an *intemset*. In this way, s_i is an element representing a set of items. Since the order of items in a set is not relevant, then the order of the attributes in s_i is also not relevant.

Each row in Table 2.1 is a sequence. In the same context of the previous paragraph, each line can be considered a customer and the set of transactions that each client has already performed. The *length* of a sequence is the number of transactions in it. A sequence with *length* l is a *l-sequence*. For example, the sequence $<\{B\},\{F,G\}>$ has length equal to 2, so it can be called a *2-sequence*. Let S be a *l-sequence* and $len(S)$ the length of the sequence, $len(S) = l$ and the i-th itemset is defined as $S[i]$. An item can only exist once in an *itemset*, but can exist multiple times in various *itemsets* in a sequence.

A sequence $S_a =< s_1, s_2, ..., s_l >$ is a *subsequence* of a sequence $S_b =< y_1, y_2, ..., y_m >$ with $l \leq m$ and S_b is a *supersequence* of S_a if there exists integers $1 \leq i_1 \leq i_2 \leq ... \leq i_l \leq m$ such that $s_1 \subseteq y_{i1}, s_2 \subseteq y_{i2}, ..., s_l \subseteq y_{il}$. For example, in Table 2.1, the sequence $<\{B\},\{F,G\}>$ is contained in the sequence $<\{A,B\},\{C\},\{F,G\},\{G\},\{E\}>$ and the sequence $<\{B\},\{G\},\{F\}>$ is not contained in the sequence $<\{A,B\},\{C\},\{F,G\},\{G\},\{E\}>$. This is due to the fact that each *itemset* of the sequence $<\{B\},\{F,G\}>$ (denominated S_c) is contained in an *itemset* of the sequence $<\{A,B\},\{C\},\{F,G\},\{G\},\{E\}>$ (denominated S_d), in the same order. In this way, *itemset* $\{B\}$ of sequence S_c is contained in the *itemset* $\{A,B\}$ of sequence S_d and the itemset $\{F,G\}$ of sequence S_c is contained in the *itemset* $\{F,G\}$ of sequence S_d. In addition, the corresponding *itemsets* of each sequence occur in the same order, which allows to conclude that the sequence S_c is a *subsequence* of the sequence S_d. This does not happen with respect to the $<\{B\},\{G\},\{F\}>$ (denominated S_e) and S_d sequences. All *itemsets* of the sequence S_e are contained in *itemsets* of the sequence S_d, however the order is not the same in both sequences. Taking this into account, S_e is not a *subsequence* of S_d.

Taking as an example a sequence C, which represents a customer's purchase history, C can be define as $<\{bread\},\{ham, cheese\},\{fruit\}>$. This sequence of transactions represents 3 purchases. In this way, it is possible to verify that the customer first bought bread, then made a purchase where he bought ham and cheese and then another transaction where he bought fruit. The sequence $Y =< \{bread\}, \{ham, cheese\} >$ is a *subsequence* of C, since the transactions of the sequence Y are contained in the transactions of C in the same order. However, the sequence $Z =< \{bread\}, \{ham\} >$ is not contained in C, since there is no purchase with only ham after the transaction $\{bread\}$.

All Sequential Pattern Mining (SPM) algorithms use two types of basic operations for exploring the search space, *s-extensions* and *i-extensions*. The generation of an s-extension sequence S_a of sequence S_b is made from the addition of a new itemset to S_b after all existing itemsets. If s_i is the new itemset, S_a can be defined as $S_a = S_b \cup \{s_i\}$. The generation of an i-extension sequence S_a of sequence S_b is made from the addition of a new item to the last itemset of S_b. This new item is added to the last position of the itemset [Fournier-Viger et al., 2017]. Being i the new item to be added and I_b the last itemset of S_b, after the i-extension operation, I_b can be defined as $I_b = I_b \cup \{i\}$.

When generating or discarding frequent sequential patterns, some types of constraints may be taken into account. The minimum and maximum length (*length constraints*) are constraints relative to the minimum and maximum number of *itemsets* of a sequence. Another example of constraints is the minimum support. This corresponds to the minimum number of times a pattern needs to occur to be considered frequent. There are also the *gap constraints*. These are relative to the minimum and maximum distance between two consecutive *itemsets* of a sequence. This distance is measured in *itemsets*. The above restrictions are some of the easiest and most beneficial to integrate in a pattern mining algorithm, as they can be used to prune the search space [Fournier-Viger et al., 2017].

2.3.1 Existing Algorithms

Over the years there have been quite a few algorithms developed in the area of Sequential Pattern Mining. All Sequential Pattern Mining algorithms receive as input the sequence data from which patterns are to be discovered and a minimum support provided by the user. At the end, the algorithm returns the set of patterns found in the data that have a frequency of occurrence greater than or equal to the minimum support. It is important to note that there is always only one correct answer to a sequential pattern mining task (for a given sequence database and minimum support). Thus, the different SPM algorithms do not differ in their output. The difference between the various algorithms lies in the way they discover the sequential patterns and in the computational complexity. Various algorithms use different strategies and data structures to search for sequential patterns efficiently. As a result, some algorithms are more efficient than others [Fournier-Viger et al., 2017].

In general, SPM algorithms differ in search strategy, in the internal representation of the database, in the way they calculate the support of a given pattern, in the generation of patterns to be explored and in their constraints. In this section the advantages and limitations of the main Sequential Pattern Mining algorithms are presented, according to the mentioned characteristics. The SPAM algorithm will be described in more detail. This is due to the fact that this algorithm was chosen as the base algorithm for the implementation of this dissertation. The reason for this choice is discussed in Section 3.3.1.

In relation to the search strategy an algorithm can be categorised as using breadth-first or depth-first search.

2.3.1.1 Breadth-first Search Algorithms

Breadth-first search algorithms use an approach called a *level-wise approach*. This is due to the fact that this type of algorithm generates patterns in ascending order of their length. In this way, the database is initially scanned in order to find all the frequent patterns with length equal to 1. Subsequently, patterns with 2 items (length equal to 2) are generated using operations *s-extensions* and *i-extensions*. This generation of patterns with incremental size continues until it is not possible to generate more sequences [Fournier-Viger et al., 2017].

Below are the two most popular breadth-first search algorithms in the SPM area:

- AprioriAll [Agrawal and Srikant, 1995] - The algorithm AprioriAll was one of the algorithms that served as the basis for many other algorithms. This algorithm is based on the Apriori property. This property can also be called downward-closure property or anti-monotonicity. It states that if a sequence is not frequent, then all its extensions will not be either. This property also states that all non-empty subsequences of a given frequent sequence are also frequent. This property is quite useful in pruning the search space. The generation of candidate sequences is done according to the Apriori-generate join procedure. Non-existent candidate patterns are generated and all candidate patterns are kept in memory. [Chand et al., 2012]. This algorithm applies a level-wise search strategy for finding frequent patterns. It uses a horizontal representation of the database. A horizontal representation of the database consists of a table, where for each sequence ID is obtained the entire sequence. Table 2.1 is an example of a horizontal representation of the database. This representation is made from multiple scans to the database. This means that the database is read multiple times. Initially, all 1-sequence patterns are discovered and stored in memory. Subsequently, the size of the frequent sequences to be discovered is increased. At each iteration a new scan is made to the database and the frequent patterns found are stored in memory. The fact that this algorithm performs several scans to the database and maintains the candidate patterns in memory significantly affects the performance. Another limitation of this algorithm is that it generates candidate patterns that may not exist. This limitation may also affect the speed of the algorithm.

- GSP (**G**eneralized **S**equential **P**atterns) [Srikant and Agrawal, 1996] - The authors of the algorithm AprioriAll then proposed an improved version called GSP. This algorithm introduced maximum and minimum interval constraints between two itemsets of a sequence (*gap constraints*).

In Table 2.2, the algorithms described above are summarised.

Table 2.2: Features of the breadth-first search algorithms of SPM.

Algorithm	**Search Strategy**	**Database Representation**	**Support Calculation**	**Generation of Candidate Sequences**	**Constraints**
AprioriAll	Breadth-first search	Horizontal database	Multiple scans to the database	Apriori-Based (*generate-candidate-and-test* approach)	Minimum support
GSP	Breadth-first search	Horizontal database	Multiple scans to the database	Apriori-Based (*generate-candidate-and-test* approach)	Minimum support; *gap constraints*

2.3.1.2 Depth-first Search Algorithms

The difference between algorithms with a breadth-first search strategy and a depth-first search strategy lies in how the candidate sequences are generated. In algorithms that have a DFS strategy the generation of candidate patterns is made from the generation of super-sequences of a certain sequence until it is not possible to expand further. Only after all possible super-sequences of a given sequence (node) have been expanded it is possible to explore another node of the same level. In algorithms with a BFS strategy, first, all nodes of a certain level (all sequences with the same length) are exploited. Subsequently the sequences of the following nodes (sequences with larger sizes) are explored. DFS algorithms allow patterns with a larger size to be generated earlier. This leads to discarding more research paths. In this way, the number of candidate patterns that appear to be non-frequent generated is smaller compared to BFS algorithms [Chand et al., 2012].

Below are some well known depth-first search algorithms used in the SPM area:

- SPADE (**S**equential **PA**ttern **D**iscovery using **E**quivalence classes) [Zaki, 2001] - This algorithm was created to correct the disadvantages of the GSP algorithm. SPADE is based on a frequent itemset mining algorithm, Eclat [Zaki, 2000]. It utilises a vertical database representation rather than a horizontal database representation. The vertical representation of a sequence database indicates the itemsets where each item appears in the sequence database. For a given item, this information is called the *IDList* of the item. This representation greatly facilitates the calculation of support of a sequence. Using a vertical representation of the database causes the database to be scanned only once. From the *IDList* of a given pattern it is possible to know the support of this pattern. This support is given by the number of different sequence identifiers in the table (vertical table rows). Table 2.3 illustrates the vertical representation for item *A* (*IDList* of item *A*) of the database represented in the horizontal table 2.1. Note that for the vertical representation of the entire database, a table is made for each item in table 2.1. This new representation leads to an improvement in performance over BFS algorithms. A disadvantage of this algorithm in relation to GSP is the impossibility of defining *gap constraints*. In this algorithm it is only possible to define the minimum support for the patterns found.

- SPAM (**S**equential **PA**ttern **M**ining) [Ayres et al., 2002] - Like SPADE, the SPAM algorithm

Table 2.3: *IDList* of the item A of the database represented in Table 2.1.

Sequence ID	**Itemsets**
1	1
2	1, 4
3	1
4	

exploits the search space with a DFS strategy. The entire database is scanned only once for the creation of *IDLists* for all items. Subsequently, to create the *IDList* of a pattern it is only necessary to join the items of the pattern. In previous algorithms, when an item appears in many sequences, its IDList was too long. This leads to the joining operation of two IDLists having a high cost. An improvement made by this algorithm was the use of bit vectors as a representation of *IDLists* (Bitmap). The pruning technique followed by this algorithm is based on the Apriori property, both for *i-extentions* and for *s-extensions*. Another advantage of this algorithm in relation to SPADE is that it is possible to define *gap constraints* and *length constraints* [Fournier-Viger et al., 2017].

Algorithm 1 illustrates the pseudocode of the SPAM algorithm. Initially, it is necessary to find all *items* with a support greater than or equal to the minimum support (***minSup***). The database is scanned to create the vertical representation of the database. From the vertical representation the frequent items are identified. Subsequently, for each sequence consisting of an itemset with an item belonging to the list of frequent items an DFS is done to find *s-extensions* and *i-extensions*. The process of generation *s-extensions* corresponds to the *s-extension* step (***S-step*** in Algorithm 1) and the process of generating *i-extensions* corresponds to the *i-extension* step (***I-step*** in Algorithm 1). In this way, it is possible to associate with each sequence $Seq =< s_1, ..., s_{|Seq|} >$, where $|Seq|$ represents the length of the sequence ***Seq***, two sets: $\boldsymbol{S_n}$, the set of candidate items that are considered for possible ***S-step*** extensions of sequence *Seq*, and $\boldsymbol{I_n}$, which identifies the set of candidate items that are considered for a possible ***I-step*** extensions. In each sequence, the support of each *s-extended* child and each *i-extended* child is tested. If the support of a generated sequence ***Seq*** is greater than or equal to ***minSup***, that sequence is saved and then the DFS (function **DFS-Pruning** in the Algorithm 1) is repeated recursively on ***Seq***. If the support of ***Seq*** is less than ***minSup***, then it is not necessary to repeat the DFS on ***Seq*** by the Apriori property, since any child sequence generated from ***Seq*** will not be frequent. If none of the generated children are frequent, then the sequence is a leaf and we can backtrack up the tree [Ayres et al., 2002].

It should also be noted that in algorithm 1, an item i is greater than an item j in case j occurs after i. As an example, in the sequence $< \{A, B, C\}, \{A, C\} >$ B and C are greater than A and C greater than B. This is because B and C occur sequentially after A and C occur sequentially after B.

```
SPAM(Sequential Database, minSup)
Vertical Representation ← Vertical Representation of the Sequential Database
Frequent Items ← list of frequent items taking into account the Vertical Representation
for each item $i \in$ Frequent Items do
    DFS-Pruning($< \{i\} >$, Frequent Items, all elements in Frequent Items greater than $s$,
      minSup)
end

DFS-Pruning($Seq =< s_1, ..., s_{|Seq|} >$, $S_n$, $I_n$, minSup)
$S_{temp} = \emptyset$
$I_{temp} = \emptyset$
for each item $i \in S_n$ do
    if $< s_1, ..., s_{|Seq|}, \{i\} >$ is frequent then
        $S_{temp} = S_{temp} \cup \{i\}$
end
for each item $i \in S_{temp}$ do
    DFS-Pruning($< s_1, ..., s_{|Seq|}, \{i\} >$, $S_{temp}$, all elements in $S_{temp}$ greater than i, minSup)
end
for each item $i \in I_n$ do
    if $< s_1, ..., s_{|Seq|} \cup \{i\} >$ is frequent then
        $I_{temp} = I_{temp} \cup \{i\}$
end
for each item $j \in I_{temp}$ do
    DFS-Pruning($< s_1, ..., s_{|Seq|} \cup \{i\} >$, $S_{temp}$, all elements in $I_{temp}$ greater than i, minSup)
end
```

Algorithm 1: The pseudocode of the SPAM algorithm [Ayres et al., 2002].

- Fast [Salvemini et al., 2011] - Fast was inspired by the SPAM algorithm. This algorithm introduces the concept of indexed sparse IDLists. This new type of storage structure was introduced in order to reduce the time required to calculate the support of a pattern and the storage memory required.

- CM-SPADE (**C**o-occurrence **MAP** - **SPADE**) and CM-SPAM (**C**o-occurrence **MAP** - **SPAM** [Fournier-Viger et al., 2014]) - One of the disadvantages of SPAM and SPADE is that they follow a *generate-candidate-and-test* approach. In this way, the patterns are generated first and then tested, in order to verify if they are frequent or not. Although the generation of non-frequent candidates is less than in horizontal representations of the database, non-frequent patterns continue to be generated. As a way of trying to improve this, Fournier-Viger et al. [2014] proposed a new approach. This new approach involves the creation of a new structure to store co-occurrence information, the Co-occurrence Map (CMAP), which allows a better pruning of the search space. This new structure is created from the initial scan of the database. In this step all the sequences with the size of two (2-sequences) are saved. Subsequently, for each pattern to be considered, it is checked if its last two items are frequent or not. If they are not, this pattern is no longer considered and is not generated [Fournier-Viger et al., 2017].

In Table 2.4, the algorithms described above are summarised.

2.3.1.3 Pattern-growth Algorithms

In addition to the DFS and BFS algorithms, there are also the pattern-growth algorithms. One of the problems of the DFS and BFS algorithms is the generation of candidate sequences that may not exist in the database. These algorithms use a *generate-candidate-and-test* approach. Generating patterns is done by joining smaller patterns. This happens because the database is scanned only once. In this way, in the pattern generation phase it is not verified whether a pattern is possible or not. It is only in the testing phase that the invalid sequences are disregarded.

Pattern-growth algorithms are algorithms that have a depth-first search strategy and that attempt to solve the referred problem. These algorithms avoid the problem described above by recursively scanning the database to find larger patterns (*divide-and-conquer* approach). Thus, they only consider patterns actually appearing in the database. However, one disadvantage of this type of algorithm is the cost in terms of time and space, since many scans and database projections are being made. A projection of the database for a sequential pattern S contains all and only the necessary information for mining the sequential patterns that can grow from S. In terms of memory, creating database projections can consume a huge amount of memory if it is naively implemented, as in the worst case it requires to copy almost the whole database for each database projection [Fournier-Viger et al., 2017].

Below are some relevant pattern-growth algorithms used in the SPM area:

- FreeSpan (**Fre**quent pattern projected **S**equential **pat**ter**n** mining) [Han et al., 2000a] - FreeSpan was developed to reduce the cost associated with the generation of non-frequent candidate patterns, which was based on the Apriori property. This algorithm performs several projections of the database recursively from the existing frequent items. From each projection several subsequences are formulated. In this way, the size of the projections of the databases is decreasing. Given this, the tests to be performed are also becoming more and more specific as several subsequences are being created. The disadvantage of this algorithm is the amount of repeated patterns it can generate [Chand et al., 2012]. Moreover, in terms of runtime, performing multiple scans to the database leads to a high cost [Fournier-Viger et al., 2017].

- PrefixSpan (**Prefix**-projected **S**equential **pat**ter**n** mining) [Pei et al., 2004] - This algorithm was based on the FPGrowth algorithm [Han et al., 2004], proposed in the area of itemset mining. Prefix-Span comes as an optimized version of the FreeSpan algorithm. As with FreeSpan, Prefix-Span only scans existing patterns in the database. However, FreeSpan has a high cost associated with performing multiple scans in the database. As a way of trying to improve runtime, this algorithm introduces the concept of *pseudo-projection*. This concept is based on the projection of the database from a set of pointers to the initial database [Fournier-Viger et al., 2017].

Table 2.4: Features of the depth-first search algorithms of SPM.

Algorithm	**Search Strategy**	**Database Representation**	**Support Calculation**	**Generation of Candidate Sequences**	**Constraints**
SPADE	Depth-first search	Vertical database	*IDList* allows direct calculation of the support of a pattern (only one scan to the database)	Apriori-Based (*generate-candidate-and-test* approach)	Minimum support
bitSPADE	Depth-first search	Vertical database	It uses bit vectors to represent IDLists (Bitmap)	Apriori-Based (*generate-candidate-and-test* approach)	Minimum support
SPAM	Depth-first search	Vertical database	It uses bit vectors to represent IDLists (Bitmap)	Apriori-Based (*generate-candidate-and-test* approach)	Minimum support; minimum and maximum pattern lengths (*length constraints*); *gap constraints*
Fast	Depth-first search	Vertical database	It uses indexed sparse *IDLists* in order to reduce the time and the storage memory required to calculate the support of a pattern	Apriori-Based (*generate-candidate-and-test* approach)	Minimum support
CM-SPADE	Depth-first search with Co-occurrence Map	Vertical database	*IDList* like the SPADE algorithm	Apriori-Based (*generate-candidate-and-test* approach) with a new structure to store co-occurrence information, the Co-occurrence Map (CMAP)	Minimum support
CM-SPAM	Depth-first search with Co-occurrence Map	Vertical database	Bitmap like the SPAM algorithm	Apriori-Based (*generate-candidate-and-test* approach) with a new structure to store co-occurrence information, the Co-occurrence Map (CMAP)	Minimum support; minimum and maximum pattern lengths (*length constraints*); *gap constraints*

In Table 2.5, the algorithms described above are summarised.

Although this type of algorithm only generates existing patterns in the database, its runtime is not very good. Algorithms like the CM-SPADE proved to be faster [Fournier-Viger et al., 2017].

Table 2.5: Features of the pattern-growth algorithms of SPM.

Algorithm	**Search Strategy**	**Database Representation**	**Support Calculation**	**Generation of Candidate Sequences**	**Constraints**
FreeSpan	Depth-first search	Database projections	Elaborates several projections of the database	Pattern-Growth-Based (*divide-and-conquer* approach)	Minimum support
PrefixSpan	Depth-first search	*Pseudo-projections*	Elaborates several *pseudo-projections* of the database	Pattern-Growth-Based (*divide-and-conquer* approach)	Minimum support; maximum pattern length

Table A.1 lists all Sequential Pattern Mining algorithms mentioned.

2.4 Subgroup Discovery

Subgroup discovery is a data mining technique that aims to discover interesting relationships between objects relative to a particular property or variable. The patterns found are called subgroups and are usually represented in the form of rules. These patterns combine a component of interest relative to a certain value and another component related to the frequency of occurrence of the pattern. The interest component of a pattern is associated with its deviation from the population.

The concept of finding interesting subgroups in data was first introduced in the 1990s as Data Surveying. According to Herrera et al. [2011], Subgroup Discovery can be defined as:

> In subgroup discovery, we assume we are given a so-called population of individuals (objects, customers, ...) and a property of those individuals we are interested in. The task of subgroup discovery is then to discover the subgroups of the population that are statistically “most interesting”, i.e. are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest.

A rule R, that represent a subgroup, can be defined as [Lavrač et al., 2004a]:

$$R : Conjunction \rightarrow Target_{Class} \tag{2.1}$$

The antecedent of the rule R, *Conjunction*, is a conjunction of features that represents pairs of attributes and values. The consequent of the rule, $Target_{Class}$, is the target class that represents the property or variable of interest for a Subgroup Discovery task.

As an example, let D be a dataset with three variables: $Age = \{$Less than 25, 25 to 60, More than 60$\}$, $Sex = \{M, F\}$ and $Country = \{Spain, USA, France, German\}$,and a variable of interest target variable $Money = \{Poor, Normal, Rich\}$. Some possible rules containing subgroup descriptions are:

$$R_1 : (\text{Age = Less than 25 AND Country = German}) \rightarrow Money = Rich \tag{2.2}$$

$$R_2 : (\text{Age = More than 60 AND Sex = F}) \rightarrow Money = Normal \tag{2.3}$$

Rule R_1 (Eq. 2.2) represents a subgroup of german people with less than 25 years old for which the probability of being rich is unusually high with respect to the rest of the population. According to rule R_2 (Eq. 2.3), women with more than 60 years old are more likely to have a normal economy than the rest of the population.

Subgroup Discovery is based on local exceptionality detection. In this way it seeks to know how locally exceptional a subgroup is relative to a target population. The target population is usually the total population. An area that also aims to detect this type of locally interesting patterns in contrast to global models is Local Pattern Mining [Atzmueller, 2015].

2.4.1 Main Elements of a Subgroup Discovery Algorithm

Subgroup Discovery (SD) takes into account several elements when applying an SD approach. These elements are related to the characteristics of the problem and the algorithm. It is possible to refer as main elements of an SD approach the following aspects [Herrera et al., 2011]:

- **Type of the Target Variable -** The analysis of the problem depends on the nature of the variable. The variables under study in an SD problem can have different types: binary, numerical or nominal or categorical.

  In case of a binary problem, the variables will have only two values (True or False). In this case it will only be necessary to find a subgroup for each of the two values, since the task is focused on providing interesting subgroups for the possible values, True or False.

  A Nominal problem is treated in the same way as a Binary problem, but with the difference that the target variable can have a undetermined number of values.

  The numerical problem is a bit more complex than the previous ones. In this case, the variable can be studied in different ways such as dividing the variable in two ranges with respect to the average, discretizing the target variable in a determined number of intervals [Moreland and Truemper, 2009], or searching for significant deviations of the mean, among others.

- **Description Language -** The description language refers to how the subgroups or rules are represented. These should be simple and suitable for obtaining interesting rules. Therefore, these are represented as *attribute-value* pairs in conjunctive or disjunctive normal form in general. A formula is in conjunctive normal form if it is a conjunction of one or more clauses, where a clause is a disjunction of literals. It can also be described as an AND of ORs. A literal is an atomic formula (atom) or its negation. A logical formula is considered to be in disjunctive normal form if and only if it is a disjunction of one or more conjunctions of one or more literals. It can also be described as an OR of ANDs.

- **Quality measures -** A quality measure of an SD algorithm is an evaluation measure of some particular aspect of a subset of individuals in relation to the total population. These measures represent the parameters for the evaluation and extraction of the rules. Furthermore, they provide the expert with the importance and interest of the subgroups obtained. A quality measure is a function that assigns a numeric value to a subgroup taking into account specific parameters [Duivesteijn and Knobbe, 2011]. There is no consensus about the best quality measures for SD, since different measures represent different types of interest. Some of the most commonly used quality measures in SD are presented in Chapter 2.4.2.

- **Search strategy -** The number of features and values to be considered in a subgroup discovery algorithm has an exponential relation with the dimension of the search space. With this in mind, the search strategy is something important that should be taken into account. The different search strategies and the implemented algorithms are presented in Chapter 2.4.3.

2.4.2 Quality Measures

Following are the most commonly used quality measures in Subgroup Discovery. These are classified according to their main purpose such as complexity, generality, precision and interest. There are also the hybrid measures, which aim to achieve a tradeoff between distinct types of quality measures.

The notation used in this section follows the notation used in Formula 2.1. In addition, n_{Total} represents the total number of examples in the database and $n_{Variables}$ the total number of existing variables.

Measures of Complexity

These measures are related to the simplicity and interpretability of the problem. These measures include [Herrera et al., 2011]:

- *Number of rules* - Number of rules induced from the problem.

- *Number of variables* - Number of possible variables in the antecedent of the rule.

Measures of Generality

The measures of generality are used to quantify the quality of individual rules according to the individual patterns of interest covered. Some quality measures for this purpose are:

- *Coverage* - This measure represents the percentage of examples covered by a rule [Lavrač et al., 2004b]. It can be defined as:

  $$Coverage(R) = \frac{n(Conjunction)}{n_{Total}} \tag{2.4}$$

  In this formulation, *n(Conjunction)* is the number of examples that verify the conditions determined by the antecedent of the rule and n_{Total}, as mentioned earlier, is the total number of examples.

- *Support* - The support of a rule measures the frequency (percentage) of correctly classified examples covered by the rule (True Positives) [Lavrač et al., 2004b]. The true positives are the examples that verify the condition and that are correctly labeled by the classifier. This can be computed as:

  $$Support(R) = \frac{n(Target_{Value} \times Conjunction)}{n_{Total}} \tag{2.5}$$

  In the above formula, $n(Target_{Value} \times Conjunction)$ is the number of examples that satisfy the condition and that belong to the value of the target variable. n_{Total} has the same meaning as in Eq. 2.4.

Measures of Precision

These quality measures show the precision of the subgroups and are widely used in the extraction of association rules and classification. They are related to the precision of the subgroups in terms of tradeoff between correctly and not correctly classified examples or examples that satisfy or not satisfy a rule, totally or partially. Within this group can be found:

- *Confidence* - It measures the relative frequency of examples that satisfy the complete rule among those satisfying only the antecedent [Herrera et al., 2011]:

  $$Confidence(R) = \frac{n(Target_{Value} \times Conjunction)}{n(Conjunction)} \tag{2.6}$$

  The necessary definitions for the understanding of this equation are mentioned in Eq. 2.4 and Eq. 2.5.

- *Precision measure Qg* - It measures the tradeoff of a subgroup between the number of examples classified correctly and the unusualness of their distribution [Herrera et al., 2011]. This can be computed as:

  $$Qg(R) = \frac{TP}{FP+g} = \frac{n(Target_{Value} \times Conjunction)}{n(\overline{Target_{Value}} \times Conjunction)+g} \tag{2.7}$$

TP is the number of true positives 2.5, and FP is the number of false positives. The negative negatives are the ones that do not verify the antecedent of the rule and that were incorrectly labeled. The variable g is used as a generalisation parameter. It is usually configured between 0.50 and 100.

Measures of Interest

Measures of interest are intended for selecting and ranking patterns according to their potential interest to the user. These measures aim to select and rank the patterns found [Herrera et al., 2011].

- *Interest* - It evaluates the interest of a rule taking into account its antecedent and consequent. It can be formulated as:

$$Interest(R) = \frac{\sum_{i=1}^{n_{Variables}} Gain(A_i)}{n_{Variables} \cdot \log_2(|Target_{Value}|)} \tag{2.8}$$

  Gain(A_i) is the information gain relative to the number of values or ranges of the variable A_i [Herrera et al., 2011]. The Information Gain function has its origin in Information Theory [Cover and Thomas, 2012]. It is based on the notion of entropy, which characterises the impurity of an arbitrary set of examples [Raileanu and Stoffel, 2004]. This measure is used to reduce a bias towards attribute values. The variable $n_{Variables}$ represents the total number of variables and $|Target_{Variable}|$ is the cardinality of the target variable.

- *Novelty* - This measure detects the interestingness or unusualness of a rule. Since the interestingness of a group depends both on its unusualness and size, this measure combines both factors. In this way, the unusualness of a rule is obtained from the difference between the number of examples that satisfy the condition and that belong to the value of the target variable and the examples that satisfy the condition or that belong to the value of the target variable. It can be computed as:

$$Novelty(R) = n(Target_{Value} \times Conjunction) - (n(Target_{Value}) \times n(Conjunction)) \tag{2.9}$$

  The variable $n(Target_{Value})$ is the total number of examples of the target variable. The remaining definitions are mentioned in Eq. 2.4 and Eq. 2.5

- *Significance* - This measure indicates the significance of a finding, if measured by the likelihood ratio of a rule [Lavrač et al., 2004b]. This measure can be formulated as:

$$Significance(R) = 2 \times \sum_{k=1}^{n_{Variables}} n(Target_{Value} \times Conjunction) \times \log \frac{n(Target_{ValueK} \times Conjunction)}{n(Target_{Value}) \times p(Conjunction)} \tag{2.10}$$

In the above equation, $p(Conjunction)$, computed as $\frac{n(Conjunction)}{n_{Total}}$, is used as a normalised factor. Other necessary definitions for the complete understanding of this measure are mentioned in Eq. 2.4, 2.5 and 2.8.

- *Piatetsky-Shapiro* - In 1991, Piatetsky-Shapiro [1991] suggested that any quality measure M defined to quantify the interest of an association within a pattern should verify three specific properties in order to separate strong and weak rules so high and low values can be assigned, respectively. These properties are related to the independence of occurrence between the antecedent and the consequent of the association rule and the increase or decrease of the value of the quality measure. This quality measure can also be defined as a test.

  Knowing that $p(Conjunction)$ stands for the probability or relative frequency of the antecedent of the Rule 2.1, $p(Target_{Value})$ describes the relative frequency of the consequent of the rule and *Support(R)* the support of the rule (Eq. 2.5), these properties can be described as follows:

  Property 1: $M(Conjunction \rightarrow Target_{Value}) = 0$ when $Support(R) = p(Conjunction) \times p(Target_{Value})$. This property claims that any quality measure M should test whether X and Y are statistically independent. In probability theory, two events are statistically independent if the occurrence of one does not affect the probability of occurrence of the other.

  Property 2: $M(Conjunction \rightarrow Target_{Value}) = 0$ increases with $Support(R)$ when $p(Conjunction)$ and $p(Target_{Value})$ remain the same.

  Property 3: $M(Conjunction \rightarrow Target_{Value}) = 0$ decreases with $p(Conjunction)$ or with $p(Target_{Value})$ when other parameters remain the same, i.e. $Support(R)$ and $p(Conjunction)$ or $p(Target_{Value})$ remain unchanged.

Hybrid Measures

These measures attempts to obtain a tradeoff between generality, interest and precision in the results obtained. [Herrera et al., 2011]. The different quality measures used can be found below:

- *Unusualness* - This measure derives from novelty (Eq. 2.9). It is defined as the weighted relative accuracy of a rule. This measure can be formulated as:

$$WRAcc(R) = \frac{n(Conjunction)}{n_{Total}} \times (\frac{n(Target_{Value} \times Conjunction)}{n(Conjunction)} - \frac{n(Target_{Value})}{n_{Total}}) \Leftrightarrow \tag{2.11}$$

$$\Leftrightarrow WRAcc(R) = p(Conjunction) \times (p(Target_{Value} \times Conjunction) - p(Target_{Value}))$$

  From this equation, the unusualness of a rule can be defined as the balance between the coverage of the rule, $p(Conjunction)$ (Eq. 2.10) and its accuracy gain, $p(Target_{Value} \times Conjunction) - p(Target_{Value})$.

2.4.3 Existing Algorithms

Since the 1990s several approaches have been developed in the area of Subgroup Discovery. The algorithms that contributed to the advancement of this area can be grouped into three main groups: extensions of association algorithms, extensions of classification algorithms and evolutionary fuzzy systems [Herrera et al., 2011].

In the following sections the algorithms belonging to each group are detailed. The SD algorithms are briefly described and then characterised according to their description language, type of target value, quality measures and search strategy. Subsequently, a comparative analysis is also made taking into account the mentioned characteristics.

2.4.3.1 Extensions of Classification Algorithms

Several algorithms resulting from the adaptation of classification rules have been developed for the discovery of subgroups. Classification rule learning algorithms have the objective of generating models consisting of a set of rules inducing properties of all the classes of the target variable, while in subgroup discovery the objective is to discover individual rules of interest. With this in mind, in order to use a classification rule learning algorithm for subgroup discovery, some modifications must be implemented [de Almeida, 2012].

Within the SD algorithms developed as extensions of classification algorithms it is still possible to distinguish between those that are based on classification algorithms and the pioneering algorithms of the SD area, EXPLORE and MIDOS. This distinction is due to the fact that EXPLORE and MIDOS were the first to be developed and use a different search strategy. The pioneers algorithms were the first to be created and use a different search strategy.

- EXPLORA [Klösgen, 1996] - This algorithm was developed in 1996 and it was the first one proposed in the SD area. In this algorithm the interest of a rule is based on some statistical measures, such as generality (size of the subgroup) and redundancy. The EXPLORA algorithm relates to an aspect of interestingness called *non-redundancy*. A hypothesis (subgroup) *H1* is redundant with respect to another hypothesis *H2*, if *H1* can be derived from *H2*. In terms of quantification of redundancy, this can be expressed by the conditional probability of *H1* given *H2*. This algorithm uses exhaustive and heuristic search strategies. These strategies are performed without pruning.

- MIDOS [Wrobel, 1997] - The MIDOS algorithm uses EXLORA in multi-relational databases, more precisely in the search for subgroups with rare statistical distributions. It searches the space of rules in an exhaustive way. In this algorithm the interest of a subgroup takes into account the size of the subgroups and the distributional unusualness. It uses as quality measure the novelty (Eq. 2.9). As a form of optimisation, this algorithm uses sampling in the data space. This allows the reduction of the search space and the acceleration of the search process.

Table 2.6: Features of the pioneering algorithms for SD [Herrera et al., 2011].

Algorithm	**Description Language**	**Type of Target Value**	**Quality Measures**	**Search Strategy**
EXPLORA	Conjunctions of pairs attribute-value. Operators $=$ and $\neq$	Categorical	Redundancy, generality, among others	Exhaustive and heuristic without pruning
MIDOS	Conjunctions of pairs attribute-value. Operators $=$, $<$, $>$ and $\neq$	Binary	Novelty or distributional unusualness, among others	Exhaustive and minimum support pruning

Although both algorithms use an exhaustive or heuristic search strategy, the type of target value that they support is different. While EXPLORA support categorical variables, MIDOS support binary variables. The quality measures used by both algorithms are also different. EXPLORA uses as quality measures the generality, redundancy, among others. MIDOS uses novelty or distributional unusualness [Herrera et al., 2011]. Both measures use conjunctions of pairs and the operators $=$ and $\neq$ to represent subgroups. In addition, MIDOS can also use the operators $<$ and $>$. As an example, the rules 2.2 and 2.3 use conjunctions of pairs and the operator $=$ to represent the subgroups. Table 2.6 summarises the features of both algorithms.

In addition to the pioneer algorithms, there are other algorithms developed by means of adaptations of algorithms used for classification. The algorithms presented below are algorithms based on classification algorithms and are not part of the pioneering algorithms of SD.

- SubgroupMiner Klösgen and May [2002] - SubgroupMiner is an extension of the pioneer algorithms mentioned above. It uses decision rules and interactive search in the space of the solutions. This allows the use of large databases by means of an efficient integration of databases, visualisation based on interaction options, among others. Although this algorithm allows the use of several measures of quality, the most common is the binomial test [Klösgen, 1996]. In the context of a rule, a binomial test is used when a $Conjunction$ has two possible $Target_{C}lasses$ and it is expected that the probability of a $Target_{C}lasses$ taking into account the antecedent of the rule is a given value. A binomial test is run to see if the observed test results differ from what was expected. In this way, it measures the deviations from the expected distribution of observations when they are in the form of two categories. SubgroupMiner only supports categorical target values. Regarding the description language, this algorithm uses conjunctions of pairs attribute-value and the operator $=$ to represent the subgroups found.

- Gamberg and Lavrac's SD (Subgroup Discovery) - In 2002, Gamberger and Lavrac [2002] presented an approach to expert-guided subgroup discovery. This algorithm is a rule induction system based on beam search algorithms. Beam search algorithms use breadth-first search to build its search tree. At each level of the tree, it generates all successors of the states at the current level, sorting them in increasing order of an utility function. This algorithm is guided by expert knowledge: instead of defining an optimal measure to discover and automatically select the subgroups, the objective is to help the expert in performing flexible and effective searches on a wide range of optimal solutions. Discovered subgroups must satisfy the minimal support criteria and must also be relevant. The algorithm keeps the best subgroups descriptions in a fixed width beam and in each iteration a conjunction (antecedent of the rule) is added to every subgroup description in the beam, replacing the worst subgroup in the beam by the new subgroup if it is better. It supports target values of the categorical type, such as SubgroupMiner. The main quality measure used by this algorithm is a precision measure called Qg (Eq. 2.7). To describe the subgroups found, this algorithm uses conjunctions of pairs attribute-value and the operators $=$, $<$ and $>$.

- CN2-SD [Lavrač et al., 2004c] - The algorithm CN2-SD was obtained by adapting the CN2 classification rule learner algorithm to Subgroup Discovery. This algorithm represents subgroups in the form of rules and uses unusualness (Eq. 2.11) as quality measure for the selection of rules. It uses target variables of the categorical type and performs a beam search strategy. Regarding the description language, this algorithm uses conjunctions of pairs attribute-value and the operators $=$, $\neq$, $<$ and $>$.

- RSD (Relational Subgroup Discovery) [Železnỳ and Lavrač, 2006] - This algorithm derives from the CN2-SD algorithm and allows the discovery of relational subgroups. The quality measures used by this algorithm are the unusualness (Eq. 2.11), significance (Eq. 2.10) or coverage (Eq. 2.4). As in the algorithms presented above, the RSD algorithm supports target values of the categorical type. This algorithm has the same description language as the SD algorithm.

Table 2.7 summarises the features of algorithms that extend classification algorithms. These are characterised by the description language, type of the target variable, the quality measures they support and the search strategy.

2.4.3.2 Extensions of Association Algorithms

The purpose of association rule algorithms is to find relationships between data variables. In this type of algorithms, several variables can appear both in the antecedent and consequent of the rule. In contrast, in SD the consequent of the rule, consisting of the property of interest is prefixed. The characteristics of association rule algorithms make it possible to extend them to SD tasks [Herrera et al., 2011].

Below, some SD algorithms based on association algorithms are briefly described. All algorithms presented below use decision trees for representation.

Table 2.7: Features of algorithms of SD based on classification [Herrera et al., 2011].

Algorithm	**Description Language**	**Type of Target Value**	**Quality Measures**	**Search Strategy**
SubgroupMiner	Conjunctions of pairs attribute-value. Operators $=$	Categorical	Binomial test	Beam search
SD	Conjunctions of pairs attribute-value. Operators $=$, $<$ and $>$	Categorical	Qg	Beam search
CN2-SD	Conjunctions of pairs attribute-value. Operators $=$, $<$, $>$ and $\neq$	Categorical	Unusualness	Beam search
RSD	Conjunctions of first order features. Operators $=$, $<$ and $>$	Categorical	Unusualness, significance or coverage	Beam search

- APRIORI-SD [Kavšek et al., 2003] - The APRIORI-SD algorithm was obtained from the modification of the algorithm APRIORI-C. This algorithm is applied to target variables with categorical type and uses the unusualness (Eq. 2.11) as a quality measure for the induced rules and probabilistic classification of the examples. According to Kavšek et al. [2003], the algorithm APRIORI-SD produces results similar to the CN2-SD algorithm. Comparisons were also made with the RIPPER, CN2 and APRIORI-C algorithms and it was concluded that the subgroup discovery algorithm APRIORI-SD was able to produce smaller sets of rules with greater coverage (Eq. 2.4) and significance (Eq. 2.10). Regarding the description language, the APRIORI-SD algorithm uses conjunctions of pairs attribute-value and the operators $=$, $<$ and $>$ to represent the subgroups found.

- SD4TS (**S**ubgroup **D**iscovery **F**or **T**est **S**election) [Mueller et al., 2009] - This algorithm is based on the APRIORI-SD algorithm. SD4TS uses a beam search strategy. The search space in this algorithm is further reduced using the quality of the subgroup to prune the search space. This algorithm was created in the context of medical diagnosis. In this way, a more specific quality measure was proposed for this context, called Prediction Measure. The prediction quality expresses how close the assessment comes to the diagnosis found. Having defining the score for a single lesion, it is possible to obtain the prediction quality of a test for an example set by averaging over the prediction scores of that test and all lesions in the example set. In this algorithm, the type of target value is categorical. The SD4TS algorithm uses the same description language as APRIORI-SD.

- SD-Map [Atzmueller and Puppe, 2006] - SD-Map is an algorithm that has an exhaustive search strategy for binary variables. An exhaustive search or brute-force search (also known as generate and test) is a problem-solving technique that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement. Atzmueller and Lemmerich [2009] also proposed the use of this algorithm for continuous variables. This algorithm uses the FP-growth method [Han et al., 2000b] for mining association rules for the subgroup discovery task. The most commonly used quality measures with this algorithm are *Piatetsky-Shapiro* (Section 2.4.2), unusualness (Eq. 2.11) and the binomial test. This algorithm uses conjunctive languages with internal disjunctions and the operator $=$ to represent the subgroups found.

- DpSubgroup [Grosskreutz et al., 2008] - This algorithm uses an exhaustive search strategy to explore the search space and uses a frequent pattern tree to obtain the subgroups efficiently. The frequent pattern tree (FP-Tree) is an efficient data structure for association-rule mining without generation of candidate *itemsets* [Hong et al., 2008]. This algorithm works with target variables of the binary or categorical type. There are several possibilities of quality measures used by this algorithm, such as *Piatetsky-Shapiro* (Section 2.4.2). This algorithm represents the subgroups found from conjunctions of pairs attribute-value and using the operators $=$.

- Merge-SD [Grosskreutz and Rüping, 2009] - This algorithm uses an exhaustive search strategy with a new type of pruning scheme which exploits the constraints among the quality of subgroups ranging over overlapping intervals. The main quality measure used by this algorithm is the *Piatetsky-Shapiro* (Section 2.4.2). Merge-SD works with target variables of the continuous type. In addition to the representation of subgroups from conjunctions of pairs attribute-value and from the operators $=$, $<$, $>$ and $\neq$, this measure also allows representation of the subgroups from intervals. The age of a person or their blood pressure are examples of two continuous variables. In these cases, the features of the subgroup description can involve interval, like for example *blood_pressure* $\in]80,120]$ or *age* $\in]18,23]$.

- IMR [Boley and Grosskreutz, 2009] - This algorithm uses an exhaustive search with tight optimistic estimate pruning, like the DpSubgroup algorithm. It represents the subgroups from the conjunctions of pairs attribute-value and from the equality operator. This algorithm can use several quality measures, but the main measure used by this algorithm is the binomial test. From several experiments, Boley and Grosskreutz [2009] demonstrated that the search space and output are significantly reduced with the use of this algorithm. This algorithm represents the subgroups found in the same way as the algorithm DpSubgroup.

Of the algorithms presented above APRIODI-SD and SD4TS are extensions of the association rule learner algorithm APRIORI. The remaining are adaptations of the FP-Growth algorithm.

Table 2.8 summarises the features of algorithms that extend association algorithms.

2.4.3.3 Extensions of Evolutionary Algorithms

Evolutionary algorithms use search processes that mimic the natural principles of evolution [Bäck et al., 1997]. Genetic algorithms [Holland, 1975] are the most widely used evolutionary algorithms. For a complete understanding of this section it is necessary to define what is fuzzy logic

Table 2.8: Features of algorithms of SD based on association [Herrera et al., 2011].

Algorithm	**Description Language**	**Type of Target Value**	**Quality Measures**	**Search Strategy**
APRIORI-SD	Conjunctions of pairs attribute-value. Operators $=$, $<$ and $>$	Categorical	Unusualness	Beam search with minimum support pruning
SD4TS	Conjunctions of pairs attribute-value. Operators $=$, $<$ and $>$	Categorical	Prediction quality	Beam search with pruning
SD-MAP	Conjunctive languages with internal disjunctions. Operator $=$	Binary	*Piatetsky-Shapiro*, unusualness, binomial test, among others	Exhaustive search with minimum support pruning
SD-MAP*	Conjunctive languages with internal disjunctions. Operator $=$	Continous	*Piatetsky-Shapiro*, unusualness, lift	Exhaustive search with minimum support pruning
DpSubgroup	Conjunctions of pairs attribute-value. Operator $=$	Binary and categorical	*Piatetsky-Shapiro*, split, gini and pearson's X^2, among others	Exhaustive search with tight optimistic estimate pruning
MergeSD	Conjunctions of pairs attribute-value. Operators $=, <, >, \neq$ and intervals	Continuous	*Piatetsky-Shapiro*, among others	Exhaustive search with pruning based on constraints among the quality of subgroups
IMR	Conjunctions of pairs attribute-value. Operator $=$	Categorical	Binomial test	Heuristic search with optimistic estimate pruning

and fuzzy rules. Fuzzy logic is a form of many-valued logic (propositional logic with more than two truth values) in which the truth values of variables may be any real number between 0 and 1. It is employed to handle the concept of partial truth, where the truth value may range between completely true and completely false. Fuzzy rules are used within fuzzy logic systems to infer an output based on input variables [Cord et al., 2001]. Crisp logic is essentially boolean logic, in which the statement is set to 0 or 1. In crisp logic, intermediate values are not allowed as in fuzzy logic. Just as there are fuzzy rules for fuzzy logic, there are also crisp rules for crisp logic.

Below, some SD algorithms based on evolutionary algorithms are briefly described.

- SDIGA [Del Jesus et al., 2007] - This algorithm uses a search strategy based on genetic algorithms. SDIGA uses fuzzy rules as a descriptive language in the subgroup specification and works with target variables of the nominal type. It supports several types of quality measures, like confidence (Eq. 2.6) and support (Eq. 2.5) and can also use other measures such as interest (Eq. 2.8), significance (Eq. 2.10) or unusualness (Eq. 2.11). The algorithm

Table 2.9: Features of algorithms of SD based on evolutionary algorithms [Herrera et al., 2011].

Algorithm	Description Language	Type of Target Value	Quality Measures	Search Strategy
SDIGA	Conjunctive or disjunctive fuzzy rules. Operators =	Nominal	Confidence, support, sensitivity, interest, significance or unusualness, among others	Genetic algorithm
MESDIF	Conjunctive or disjunctive fuzzy rules. Operators =	Nominal	Confidence, support, sensitivity, significance or unusualness, among others	Multi-objective genetic algorithm
NMEEF-SD	Conjunctive or disjunctive fuzzy and/or crisp rules. Operators =	Nominal	Confidence, support, sensitivity, significance or unusualness, among others	Multi-objective genetic algorithm

evaluates the quality of the rules by means of a weighted average of the measures selected. This algorithm support target values of nominal type.

- MESDIF [Berlanga et al., 2006] - MESDIF uses a search strategy based on multi-objective genetic algorithms for the extraction of subgroups. This approach applies the concepts of the SPEA2 multi-objective approach [Zitzler et al., 2001]. The subgroups discovered are described by fuzzy rules. This algorithm use several quality measures at the same time. Confidence (Eq. 2.6), support (Eq. 2.5) and significance (Eq. 2.10) are some examples. As the previous algorithm, this algorithm uses nominal target values.

- NMEEF-SD [Carmona et al., 2009] - This algorithm has very similar characteristics to the MESDIF algorithm. However, this algorithm uses a multi-objective approach based on NSGA-II [Deb et al., 2002]. NMEEF-SD intends to extract fuzzy and/or crisp rules for the description of subgroups.

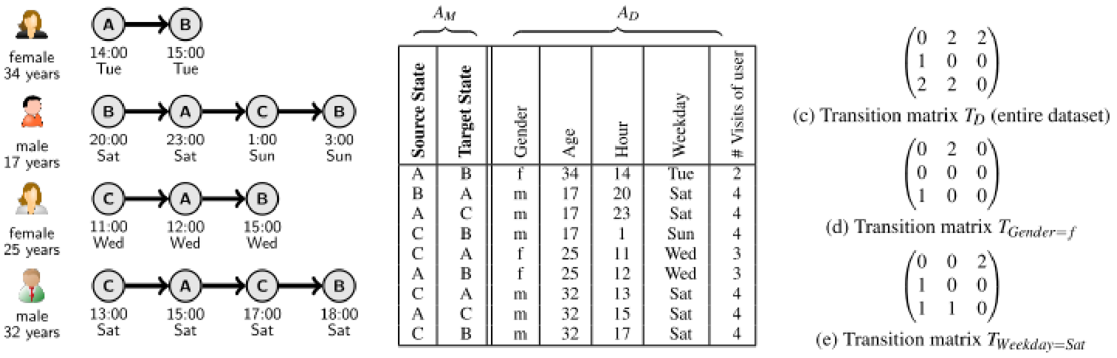
From the analysis of these three algorithms we can see that the evolutionary algorithms applicable to the discovery of subgroups are based on a hybrid model between evolutionary genetic algorithms and fuzzy logic. All these algorithms make use of some quality measures applicable in association-based algorithms.

Table 2.9 summarises the features of algorithms that extend evolutionary algorithms.

Table A.2 lists all Subgroup Discovery algorithms mentioned in this dissertation.

2.5 Related Work

Lemmerich et al. [2016] presented an approach to find subgroups with exceptional transition behaviour in sequential datasets. This article is part of the Exceptional Model Mining area [Leman et al., 2008]. This area is a generalisation of Subgroup Discovery. Exceptional Model Mining is concerned with finding patterns that reveal unusual interactions among multiple target attributes. While in SD a subgroup is considered interesting considering the distribution of a single target variable, in EMM a subgroup is considered interesting if its model parameters deviate significantly from the parameters of the model that is derived from all dataset instances. When this happens, the subgroup is considered exceptional. EMM allows the discovery of subgroups with more complicated target concepts.



A_M		A_D				
Source State	Target State	Gender	Age	Hour	Weekday	# Visits of user
A	B	f	34	14	Tue	2
B	A	m	17	20	Sat	4
A	C	m	17	23	Sat	4
C	B	m	17	1	Sun	4
C	A	f	25	11	Wed	3
A	B	f	25	12	Wed	3
C	A	m	32	13	Sat	4
A	C	m	32	15	Sat	4
C	B	m	32	17	Sat	4

$$\begin{pmatrix} 0 & 2 & 2 \\ 1 & 0 & 0 \\ 2 & 2 & 0 \end{pmatrix}$$

(c) Transition matrix T_D (entire dataset)

$$\begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

(d) Transition matrix $T_{Gender=f}$

$$\begin{pmatrix} 0 & 0 & 2 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

(e) Transition matrix $T_{Weekday=Sat}$

Figure 2.1: Illustrative example of the approach described [Lemmerich et al., 2016].

One of the EMM problems, as well as SD, is the inability to capture transitional behaviour in the data. In this way it is impossible to find unusual patterns in sequences. As a way of approaching this problem, this article proposes the introduction of first-order Markov chains as a new model class for EMM. Markov chains are stochastic systems modelling transitions between states $s_1, \ldots, s_m$. Each observed sequence of states corresponds to a sequence of assignments of random variables $X_1, \ldots, X_z, X_i \rightarrow s_1, \ldots, s_m$. In addition, a new quality measure is also proposed. This measure is based on the difference between the Markov transition matrices of a given subgroup and the Markov Transition Matrix of the total population with the distance of random dataset samples. A Markov Transition Matrix is a square matrix used to describe the transitions of a Markov Chain. A Markov Chain describe a sequence of possible events in which the probability of each event depends only on the state attained in the previous event Gagniuc [2017].

In Fig. 2.1, sequential data with background knowledge (representative attributes of the examples who have gone through the sequence) is represented in the leftmost image. First, these data are transformed to a transition dataset with transition model attributes, AM, and descriptive attributes, AD (central image). The transition model attributes consist only of the current state and the target state. Subsequently, to discover interesting subgroups, transition matrices for the total dataset (c) and for each candidate subgroup are compiled and then compared to each other (rightmost image). In the figure the subgroups *Gender=f* (d) and *Weekday=Sat* (e) are used as examples.

In order to quantify the interest of subgroups, the authors of this paper employ an interestingness measure that assigns a score to each candidate subgroup. The score is based on a comparison between the transition matrix of the subgroup ($T_{Gender=f}$ or $T_{Weekday=Sat}$ in Fig. 2.1) and a reference transition matrix (T_D in Fig. 2.1) that is derived from the overall dataset. With this in mind, the interestingness measure that is proposed expresses how unusual the distance between the transition matrix of a subgroup and the reference matrix is in comparison to transition matrices of random samples from the overall dataset.

Chapter 3

Subgroup Discovery for Sequences

This chapter starts by describing the business problem that led to this dissertation. Subsequently, a formalisation of the problem is done in a domain-independent way and the different scientific approaches chosen to address it are presented. The different types of quality measures and their formulation are also described. In addition, a preliminary exploratory analysis of the results obtained is carried out.

3.1 Business Problem

This study came about as part of a project carried out by INESC TEC with a company that sells chatbots, Smarkio [Smarkio, 2018]. This project aims to develop a technology to support the marketing teams and at the same time the chatbot development teams. The main objective of this project is the discovery of what makes users behave differently than usual in a context of user-bot interaction. Therefore, it is intended to discover and understand patterns followed by users who behave differently than all users. These patterns are called unusual patterns. This discovery is based on the flow of user interactions.

Chatbots represented one-third to one-half of all online interactions between the years 2007 and 2015 [Tsvetkova et al., 2017] and the rate of development of chatbots has been increasing since then [Radziwill and Benton, 2017]. Currently there are chatbots on the market for everything from forecasting the weather, keeping people up to date on news, scheduling meetings, helping people manage their money, among others things [CrowdFlower, 2017]. However, creating and maintaining a successful chatbot according to a company's business goals is challenging. This analysis, when carried out by humans, can be time-consuming. The performance of a chat regarding the user's waiting time (i.e. time the bot takes to respond to the user) is an example of this type of tasks. The semantics of the interactions regarding the way user messages are interpreted by the bot, the satisfaction of the client or even the business goals are some examples of measures that can define the quality of a chatbot. In this way, the analysis of the quality of a chatbot can be related with several aspects.

Databases of this type of communication often store high amounts of interactions between bots and humans. For a human, analysing all these data, in addition to taking a lot of time, can lead to many errors. Many companies do this process manually or by using simple automatic statistical-based analysis methods.

As mentioned in Section 1.1, the conversation flow in a chatbot is not homogeneous. Depending on the chatbot, there may be more or less subjectivity on the user side. If we are dealing with a chat where the user only answers multiple choice questions, there is less subjectivity than if we were dealing with an open-ended chat. In the chabots under study the interactions made by the bot are made in natural language. However, the responses given by users are most often restricted by a selection of one of the available options. When that does not happen the answers are open. However, despite the limitations of possible answers to a question, the uncertainty of the user choices, as well as the reasons for those choices remain unknown.

A user behaviour can be represented by a set of interactions between bot and user. In this way, a behaviour can be interpreted as a pattern of interactions. As mentioned in the first paragraph of this section, the main goal of the Smarkio project is the discovery of patterns that deviate from what is considered normal. These unexpected behaviours can be either positive (i.e. interactions that exceeds expectations) or negative (i.e. interactions below expectations) in terms of business goals. Within this dissertation, a behavioural pattern is evaluated according to the deviation from mean reference values. For example, a pattern that has a higher than usual proportion of provided email address can be considered a positive pattern. In the same context, a pattern that has a proportion of provided email address lower than the average of the population can be considered a negative pattern. Therefore, understanding the improvements that must be made, or find the best practices that should be replicated is very important.

The main objective of this problem is then from behavioural patterns of the users, help to foster the progress of the companies' sales and satisfaction of their clients, which can be translated into business success. While, on the one hand it can be used as a decision support for corrections of system failures, on the other hand it can lead to the extension of best practices to other components or areas.

3.2 Problem Formalisation

As already mentioned, the problem described in this dissertation can be approached as a combination of Subgroup Discovery and Sequential Pattern Mining techniques. In this way, the purpose of this dissertation is the creation of an algorithm that allows to discover unusual patterns in sequential data.

Taking into account the fundamental concepts of a Sequential Pattern Mining (SPM) problem presented in 2.3, a sequence S with length l can be defined as $< s_1, s_2, ..., s_l >$, where each element of the sequence S (s_i) is an *intemset*. Taking into account the fundamental concepts of a Subgroup Discovery (SD) problem presented in 2.4, a subgroup is represented by a rule. The antecedent of the rule is a conjunction of features that represents pairs of attributes and values and the consequent of the rule is the class that represents the property or variable of interest for a Subgroup Discovery task.

From the combination of both topics and their fundamental concepts, it is possible to define the conjunction of features that represents pairs of attributes and values (antecedent of the rule) in a SD problem as a sequence of *itemsets* in an SPM problem. A sequence is an ordered set of *itemsets*. Thus, a sequence is characterised by its *itemsets* and their transitions. Taking into account the representation of subgroups presented in Rule 2.1, it is possible to alter this representation in order to support the discovery of subgroups in sequential data. This adaptation can be done by replacing the antecedent of the rule by a sequence of *itemsets*. The consequence of the rule, $Target_{Class}$, continues to have the same meaning as in an SD algorithm.

Considering the concepts presented, the rule R_3 represents the formalisation of a subgroup in a SD problem applied to sequences.

$$R_3 :< s_1, s_2, ..., s_l > \rightarrow Target_{Class} \tag{3.1}$$

Following are the approaches developed within this dissertation to find unusual patterns in sequential data.

3.3 Approaches

As already mentioned, this dissertation addresses the business problem described in Section 3.1 as a combination of Subgroup Discovery and Sequential Pattern Mining techniques In this section, two different approaches to solving the problem are described. In this section a domain-specific terminology for chatbots will be used. However, the terms presented are easy to generalise.

A user session is interpreted as a sequence of interactions. Taking into account the fundamental concepts of a Sequential Pattern Mining problem (Section 2.3), it is possible to define each interaction between human and bot as an element of a session. An interaction is a set of m distinct items, each representing an attribute of the interaction. For example, if we define as attributes of an interaction the text shown to the user ($Text$), the time it occurred ($Time$) and the response given by the user ($Answer$) we can define an interaction as $I = \{Text, Time, Answer\}$. A session S can be represented as a sequence consisting of an ordered set of interactions. Since each element of a sequence is an *itemset* representing an interaction, then a sequence is an ordered list of *itemsets*.

In case we define an interaction as I and a session as a sequence of l ordered interactions, we can define a session as $< I_1, I_2, ..., I_l >$. If we have a session where only 2 interactions occurred, the sequence length will be 2. In this way, we can designate this session as a *2-sequence*.

The number of user sessions that contain a certain pattern corresponds to the absolute support of that pattern. The relative support of a pattern is obtained from dividing the number of sessions where the pattern appears (i.e. sessions that are *supersequences* of the pattern) by the total number of sessions.

As mentioned, a pattern is represented by an ordered sequence of *itemsets* where each *itemset* corresponds to an interaction. Thus, in the context of chatbots sessions, the set of features that identify a pattern (antecedent of the rule) is the sequence of interactions of the same. The consequent represents the property or variable of interest. Within this dissertation, the variable of interest is represented by an indicator. An indicator can be of two types. It can be an indicator of interest or a dropout indicator. The first indicator is defined by the company and points to specific interaction (i.e. it is associated with an interaction). This indicator is only relative to some chatbot interactions. These interactions represent interesting interactions for the organisation. An interaction is considered interesting by a company if the company intends to understand what types of behavioural deviations exist with respect to it. For example, considering that we are dealing with a chatbot that has an interaction that aims to collect the user's contact. If the company wants to discover the behavioural deviations that occur in relation to this interaction, it can create an indicator associated to this interaction. In addition to the indicators of interest, there is also the dropout indicator. This indicator, contrary to the indicators of interest, is not defined by the company. The dropout indicator measures the average dropout of the reference population. Both types of indicators are explained in more detail in Section 3.4.

As an example, let $< I_1, I_2, I_3 >$ be a pattern represented by the sequence of interactions I_1, I_2 and I_3, and I_4 an interaction that has an indicator of interest associated. Rule R_4 represents the subgroup of users who traversed the sequence of interactions $< I_1, I_2, I_3 >$ and reached the indicator of interest associated with interaction I_4.

$$R_4 :< I_1, I_2, I_3 >\rightarrow Indicator_{Interest} = I_4 \tag{3.2}$$

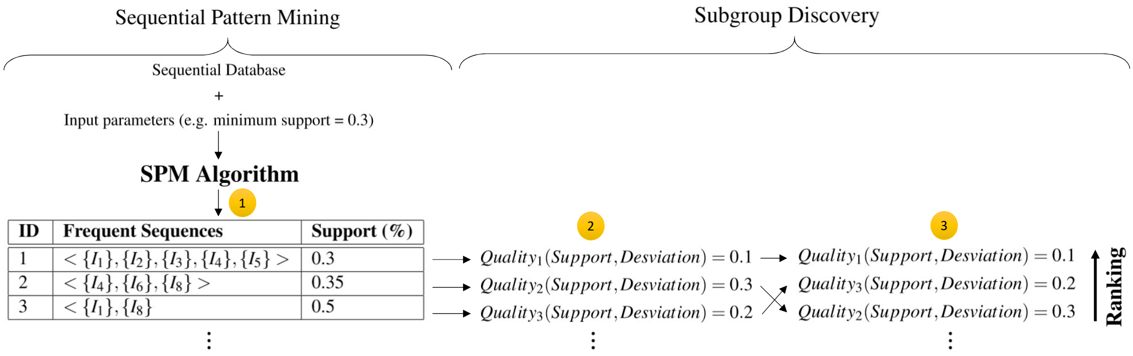
Two different approaches are proposed in this dissertation (Section 3.3.1 and Section 3.3.2). The first approach uses an off-the-shelf Sequential Pattern Mining solution to discover frequent patterns and then a post-processing to find out the interesting patterns is performed. In the second approach the discovery of frequent and interesting patterns is performed alternately.

3.3.1 Sequential Pattern Mining with Post Processing of Subgroups Discovery

This approach uses an off-the-shelf Sequential Pattern Mining (SPM) solution to discover frequent subsequences. Then, these subsequences are evaluated according to the quality measures mentioned in Section 3.5. In this way, this approach is divided into two phases.

Figure 3.1 illustrates the flow of the current approach. As an example, a minimum support of 0.3 was considered. In the first phase all the sequential patterns with a support greater than or equal to the minimum support are obtained. The output of this phase is all the frequent sequential patterns and the support associated with each pattern (Number **1** in Fig. 3.1). In a second phase, the patterns found previously are evaluated taking into account a quality measure regarding a $Target_{Class}$. Within this dissertation, a $Target_{Class}$ corresponds to an average reference value (e.g. average email delivery of the total population). The quality measure evaluates a pattern taking into account its support and its deviation from the average reference value (Number **2** in Fig. 3.1).

Finally, all patterns were ranked according to the values of the quality measure (Number **3** in Fig. 3.1).



ID	Frequent Sequences	Support (%)
1	$< \{I_1\}, \{I_2\}, \{I_3\}, \{I_4\}, \{I_5\} >$	0.3
2	$< \{I_4\}, \{I_6\}, \{I_8\} >$	0.35
3	$< \{I_1\}, \{I_8\}$	0.5

Sequential Pattern Mining
Subgroup Discovery
Sequential Database
+
Input parameters (e.g. minimum support = 0.3)
SPM Algorithm
1
2
3
$Quality_1(Support, Desviation) = 0.1$
$Quality_2(Support, Desviation) = 0.3$
$Quality_3(Support, Desviation) = 0.2$
$Quality_1(Support, Desviation) = 0.1$
$Quality_3(Support, Desviation) = 0.2$
$Quality_2(Support, Desviation) = 0.3$
Ranking

Figure 3.1: Flow of the first approach implemented.

During the first phase of the current approach it was necessary to choose an SPM algorithm to obtain the frequent sequential patterns. For this, a study of the existing SPM algorithms was made and the advantages and disadvantages of each were evaluated. Several aspects were taken into account to select the most suitable approach.

Regarding the search strategies, as mentioned in Section 2.3.1.2, DFS algorithms allow longer patterns to be generated earlier. This leads to discarding more search paths. In this way, the number of candidate patterns generated that appear to be non-frequent is smaller compared to BFS algorithms [Chand et al., 2012]. Taking this into account, algorithms with a BFS search strategy were excluded. In addition to these two types of search strategies there are also the Pattern-growth algorithms. These algorithms have a depth-first search strategy. The Pattern-growth algorithms aim to solve the problem of generating candidate sequences that may not exist in the database, which happens in DFS algorithms. However, one disadvantage of this type of algorithm is the cost in terms of time and space, since many scans are being made to the database. With this in mind, as well as the size of datasets available and some initial tests made to compare the metrics mentioned, Pattern-growth algorithms were excluded. These tests consisted of running the datasets with algorithms of both categories in the SPMF[1] (Sequential Pattern Mining Framework) library and comparing the execution times and the memory used.

Then, considering the variety of possible constraints applicable to the SPM algorithms presented in Table 2.4 and the performance in terms of execution time and spent memory, the algorithm chosen was the SPAM algorithm. It was possible to verify that the SPAM and CM-SPAM algorithms have a greater variety of possible constraints when compared to the other algorithms. The performance evaluation was done using the SPMF library. From this analysis, it was possible to verify that the SPAM algorithm has a slightly higher execution time, but it spends less memory. Considering the resources available, the memory spent by the algorithm was more valued. In this

[1] http://www.philippe-fournier-viger.com/SPMF/

way the SPAM algorithm was chosen as the base algorithm for the implementation of the first approach.

The SPAM algorithm was implemented with the following input parameters:

- Names of files with input data (all files relating to the same data set have the same name. They only differ in the file extensions)
- Minimum frequency with which a pattern must occur to be considered frequent (minimum equals to 1%)
- Minimum pattern length (by default this is 2 and this is the minimum value of this parameter)
- Maximum pattern length (by default there is no maximum length)
- Maximum interval between sequence items (by default there is no maximum interval)

In the current approach, for a pattern to be considered unusual, it must be considered frequent first. In the following approach, the discovery of frequent and interesting patterns is performed differently.

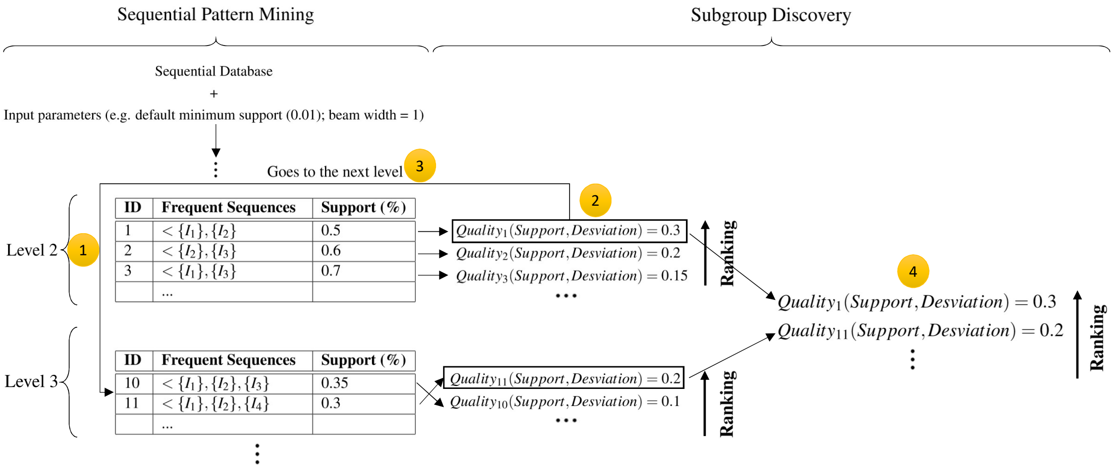
3.3.2 Sequential Pattern Mining with Subgroups Discovery On the Fly

In this approach the discovery of frequent and interesting patterns is performed alternatively. For this approach the search strategy was modified. While the first approach uses a depth-first search strategy, this approach uses a beam search strategy. This search strategy not only explores possible patterns in a different way, but also selects only some of the patterns to be expanded at the next level. This approach is described in more detail below.

As already mentioned in Chapter 2, Breadth-first search is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root and explores all of the neighbour nodes at the present depth prior to moving on to the nodes at the next depth level. Beam search uses breadth-first search to build its search tree. At each level of the tree, it generates all successors of the states at the current level, sorting them in decreasing order of an utility function. The utility function used in this context corresponds to the quality measures explained in Section 3.4. However, it only stores a predetermined number of best states at each level (called the beam width). Only those best states pass to the next level in the search tree. In this way, only the best states from the previous level are expanded at each level, except for the first level that has no previous state. In this case, all states are expanded. The greater the beam width, the fewer states are pruned. With an infinite beam width, no states are pruned and beam search is identical to breadth-first search [Fournier-Viger et al., 2017].

Figure 3.2 illustrates the flow of the current approach. At each level, from all the patterns generated in it, the ones with a lower support than the set minimum are discarded (Number **1** in Fig. 3.2). This prior selection is intended to prevent patterns that only occur a very small number of times from being selected. With this in mind, by default the minimum support is 1%. It should be noted that this support can be changed, since it is one of the input parameters of the algorithm.

After discarding the patterns with a frequency of occurrence less than the minimum support, the interest of each pattern is calculated based on the utility function. Taking into account the results obtained, a ranking of the patterns according to their interest is made (Number **2** in Fig. 3.2). Then the patterns with the best scores to be expanded to the next level are chosen (Number **3** in Fig. 3.2). The number of expanded patterns corresponds to the value of the beam width (In Fig. 3.2 the beam width is set to 1). The remaining patterns are disregarded. After discovering the best patterns at each level, a global ranking of all saved patterns is made (Number **4** in Fig. 3.2).


Sequential Pattern Mining
Subgroup Discovery
Sequential Database
+
Input parameters (e.g. default minimum support (0.01); beam width = 1)
Goes to the next level
3
2
1
4
Level 2
Level 3
$Quality_1(Support, Desviation) = 0.3$
$Quality_2(Support, Desviation) = 0.2$
$Quality_3(Support, Desviation) = 0.15$
$Quality_{11}(Support, Desviation) = 0.2$
$Quality_{10}(Support, Desviation) = 0.1$
$Quality_1(Support, Desviation) = 0.3$
$Quality_{11}(Support, Desviation) = 0.2$
Ranking
Ranking
Ranking

ID	**Frequent Sequences**	**Support (%)**
1	$< \{I_1\}, \{I_2\}$	0.5
2	$< \{I_2\}, \{I_3\}$	0.6
3	$< \{I_1\}, \{I_3\}$	0.7
	...	

ID	**Frequent Sequences**	**Support (%)**
10	$< \{I_1\}, \{I_2\}, \{I_3\}$	0.35
11	$< \{I_1\}, \{I_2\}, \{I_4\}$	0.3
	...	

Figure 3.2: Flow of the second approach implemented.

In the following section, the quality measures designed within this dissertation are described.

3.4 Quality Measures

In the current section are presented the different quality measures that evaluate the interest of a certain pattern. In this section a domain-specific terminology will be used. However, the terms presented are easy to generalise.

As discussed in Chapter 2.4.2, a quality measure of a Subgroup Discovery algorithm is an evaluation measure of some particular aspect of a subset of individuals in relation to the total population. These measure is a function that assigns a numeric value to a subgroup taking into account specific parameters [Duivesteijn and Knobbe, 2011]. These measures aim to select and rank the patterns found [Herrera et al., 2011].

As it is possible to verify from Chapter 2.4, a subgroup is represented by a rule. The antecedent of the rule is the conjunction of features that represents pairs of attributes and values. The consequent represents the property or variable of interest. Within this dissertation, the variable of interest is represented by an indicator. Thus, the quality measures developed define which patterns are interesting in relation to an indicator. In this dissertation, an indicator can be of two types. It can be an indicator of interest or a dropout indicator. The first indicator is defined by the company and is associated with a specific interaction. This indicator is only relative to some chatbot interactions. These are interactions that the company has an interest in knowing what kinds of behavioural deviations exist regarding them. Taking as an example a chatbot that has an interaction that aims to collect the email of a user. If the company wants to understand which behavioral patterns deviate from the average email delivery, then it creates an indicator associated with that interaction. In addition to the indicators of interest, there is also the dropout indicator. This indicator, contrary to the indicators of interest, is not defined by the company. The dropout indicator refers to the average dropout of the reference population. If the reference population is the total population, this indicator is obtained from the division between the sum of the average dropouts of all the interactions (Average dropout of an interaction $= \frac{\text{number of times a user left the chat in the interaction}}{\text{total number of sessions that went through the interaction}}$) by the total number of interactions.

In a Subgroup Discovery problem, a pattern is considered exceptional when compared to some reference. In most cases, the reference used for comparison is the total population [Atzmueller, 2015, Lavrač et al., 2004c, Herrera et al., 2011]. In this dissertation, two types of references were taken into account. One of the references is the total population. For a more intuitive and clear interpretation of the results, this reference was designated as **global reference**, since it represents the whole population. With this in mind, in this reference, the behaviour presented by the users/sessions that contains a pattern is compared to all existing users/sessions. A pattern is contained in a session if the pattern is a *subsequence* of the session. In this case, it is also possible to state that the session is a *supersequence* of the pattern (Section 2.3).

In addition to the comparison between the *supersequence* sessions of a pattern and the total population of individuals, a comparison was also made between *supersequence* sessions of a pattern and the sessions that contain the input the pattern (i.e. the first interaction of the pattern). As an example, let $< I_1, I_2, I_3 >$ be a pattern represented by the sequence of interactions/*itemsets* I_1, I_2 and I_3. The pattern input is the first *itemset* of the pattern, in this case I_1. This reference was designated as **local reference**. In this reference the deviation within the pattern is measured. In this case, the deviation of a pattern is calculated from the mean value of the local reference. In this way, the influence of the pattern is measured in relation to all the users who initialised the pattern. If we are analysing the pattern $P =< 3, 4, 5 >$, interaction 3 represents the beginning of the pattern. Taking this into account, the local reference population for this pattern are all the sessions that reached the interaction 3.

For each type of indicators presented, interest and dropout, quality measures were designed for both references, local and global. For each combination <*type of indicator* − *reference type*> different quality measures were formulated. Below, these measures are presented.

3.4.1 Indicators of Interest

As previously mentioned, an indicator of interest is associated with an important interaction for the company. An interaction in which the purpose is to gather information from the user or to know their satisfaction are two examples of possible interactions that can be important or interesting for a company. An indicator of interest happens in a session if the user reaches the interaction associated with the indicator during the session. In this way, an indicator I belongs to a session S, in case I belongs to the set of *itemsets* of S.

The quality measures of Subgroup Discovery methods in the literature are not suitable for sequences. Thus, within this dissertation, new quality measures were designed. These are one of the contributions of this project.

Subgroup Discovery can be formalised based on probabilities. In this way, the developed formulations were based on the probabilities presented below.

- $p(P)$ - This probability corresponds to the probability of a pattern P happening. This probability represents the support of a pattern P. The support of a pattern is calculated by dividing the number of sessions that are *supersequences* of the pattern and the total number of sessions. Taking $n(P)$ as the number of sessions where the pattern P occurs and n_{Total} the total number of chatbot sessions, we can formulate the support of a pattern as follows:

$$p(P) = \frac{n(P)}{n_{Total}} \tag{3.3}$$

- $p(I)$ - Since I represents an indicator of interest, $p(I)$ is the probability of the indicator of interest I happening in a session. As mentioned earlier, an indicator points to an interaction. In this way, $p(I)$ translates into the probability of a user going through the I interaction during their session. If $n(I)$ corresponds to the number of sessions that contain the interaction I and n_{Total} the total number of chatbot sessions, as in Eq. 3.3, we can define $p(I)$ as:

$$p(I) = \frac{n(I)}{n_{Total}} \tag{3.4}$$

- $p(I \cap P)$ - The probability $p(I \cap P)$ represents the probability of a user, during a session, reach the interaction I and the pattern P. $n(I \cap P)$ represents the number of sessions in which the user goes through the interaction I and the pattern P in the same session. With this in mind, it is possible to formalise this probability as follows:

$$p(I \cap P) = \frac{n(I \cap P)}{n_{Total}} \tag{3.5}$$

- $p(I \mid P)$ - This probability represents the conditional probability of I given P. More specifically, in the context of chatbots sessions, it translates into the probability of the interaction I being contained in a session of a user, knowing that the pattern P is a *subsequence* of the user session. This pattern may occur before, during, or after interaction I. Assuming that we have a session formed by the sequence of interactions $< A,B,C,D,E,F,G >$ and D is an interaction associated with an indicator of interest for the company. In this way, not considering any *gap constraints*, the patterns $< A,B >$, $< B,C >$, $< A,C >$ and $< A,B,C >$ are patterns that occur before interaction D. On the other hand, the patterns $< E,F >$, $< F,G >$,

$< E, G >$ and $< E, F, G >$ are patterns that occur after D. In addition to these possibilities, there may also exist patterns that occur during the interaction, such as $< A, F >$, $< B, E, G$, $< C, D, E >$, among others.

In this way, considering the formalisation of the conditional probability, we can define the probability of I given P as:

$$p(I \mid P) = \frac{p(I \cap P)}{p(P)} \tag{3.6}$$

In this formalisation, $p(I \cap P)$ is defined in Equation 3.5 and $p(P)$ in Equation 3.3.

As mentioned previously, two types of references were considered for comparison: global and local. Below is the formulation developed to assess the interest of a pattern P regarding an indicator I relative to the global reference:

$$Global_{Interest}(P) = \mid p(I \mid P) - p(I) \mid \times p(P) \tag{3.7}$$

Formula $Global_{Interest}$ (Eq. 3.7) measures the deviation in the indicator in the sessions associated with the pattern when compared to all the sessions in the data. Subgroup Discovery combines the component of interest of a pattern with the component related to its support. The development of the measure $Global_{Interest}$ (Eq. 3.7) was based on this combination. It is possible to notice that this new formula is constituted by the product of two parts.

The first part represents the deviation of the pattern. This deviation can be measured in multiple ways, such as the difference or the ratio. The ratio between probabilities would inform us about how much greater or lesser a probability is in comparison with the other. The difference between both probabilities is an absolute measure that allows us to measure how much one group differs from another. In this case, the deviation between probabilities was formulated using the difference between them. The difference between both probabilities measures how distinct the distribution of the occurrence of an indicator is in the sessions where the pattern P occurs when compared to all sessions.

From the results obtained by the first part of the formulas, it is possible to understand the influence that the pattern has on an indicator. This influence can be positive or negative. A negative influence translates into a negative result of the first part. Such happens when the probability of I happening is greater than the probability of I occur, given that P happens. If, on the other hand, the difference of the probabilities is a positive result, then the pattern has a positive influence on the indicator. In other words, if the pattern occurs the probability that the interaction of interest will happen increases.

In Formula $Global_{Interest}$ (Eq. 3.7) is considered the absolute value of this difference due to the fact that at the end of the algorithm a ranking is made according to the interest values of the patterns found, as referred to in section 3.3. It is intended that a negative value of the deviation in absolute value has the same importance as a positive value. However, when viewing the results, it is possible to see the value of both parts of the formula and check the positive or negative deviations.

Finally, the second part of the formula represents the frequency of occurrence of the pattern. Although the main goal of SD is to find patterns that are unusual, for a pattern to be considered as such, it must occur a minimum number of times. The more frequent a pattern, the more it is supported.

Regarding the local reference (sessions that contain the first pattern interaction), the following formulation was developed to find interesting patterns:

$$Local_{Interest}(P) = \mid p(I \mid P) - p(I \mid P[1]) \mid \times P(P) \tag{3.8}$$

Comparing both measures (Eq. 3.7 and Eq. 3.8), it is possible to notice that these are only different in the probability relative to the reference population. In the formula relative to the global reference, the difference between $p(I \mid P)$ and $p(I)$ was made. However, in the formula concerning the local population, $p(I)$ is replaced by $p(I \mid P[1])$. Since $P[1]$ represents the first interaction of the pattern P (i.e. first *itemset* of the sequence P), $p(I \mid P[1])$ translates into the probability of I occur knowing that the user reaches the beginning of the pattern P. As an example, if we are faced with the pattern $P =< A,B,C >$ and the indicator of interest under study is I, $p(I \mid P[1])$ can be translated into the probability of the interaction I occurring knowing that the interaction A also happens.

3.4.2 Indicators of Dropout

In this section, the measures used to evaluate the dropout interest of a pattern are presented. As previously mentioned, the dropout is a measure relative to the average dropout of the reference population. In this case, the total population. For the global reference this indicator is obtained from the division between the sum of the average dropouts of all the interactions (Average dropout of an interaction $= \frac{\text{number of times a user left the chat in this interaction}}{\text{total number of sessions that went through this interaction}}$) by the number of interactions.

As in the quality measures mentioned in the previous chapter, the formulation of the dropout interest of a pattern depends on two parts. The first part represents the deviation from the reference population. The second part represents the support of the pattern. For the global reference two quality measures were designed. For the local reference a quality measure was developed. Therefore, with respect to the global reference, the following measure have been created for the discovery of interesting patterns:

$$Global_{AVG\_Dropout}(P) = \left| \frac{\sum_{j=1}^{n_{Total}} Dropout(j)}{n_{Total}} - \frac{\sum_{i=1}^{length(P)} Dropout(i)}{length(P)} \right| \times p(P) \tag{3.9}$$

$$Global_{MIN\_Dropout}(P) = \min_{\forall i \in P}\left(\left| \frac{\sum_{j=0}^{n_{Total}} Dropout(j)}{n_{Total}} - Dropout(i) \right| \right) \times p(P) \tag{3.10}$$

As it may be noted, both formulas have similarities. In both, the support of the pattern, represented by $p(P)$, is taken into account.

In the first part of formula $Global_{AVG\_Dropout}$ (Eq. 3.9) a subtraction of two values is made. The first value corresponds to the average chat dropout. This value takes into account the average dropout of all chat interactions. For the calculation of the average chat dropout, interactions that do not lead to other interactions are not taken into account. These interactions are called final chat interactions. In this type of interactions, the user can not continue for any other interaction. The user only has the option to leave/close the chat. In this way, on a scale of 0 to 1, the average dropout of these interactions will always be 1. This would incorrectly influence the calculation of the dropout.

The second value in the first part of the Formula $Global_{AVG\_Dropout}$ refers to the average dropout of the pattern P. The average dropout of a pattern is obtained from the average of the dropouts of all its interactions. Final chat interactions were also not taken into account for the average pattern dropout. Taking into account final chat interactions could have a high impact on the average dropout of a pattern. Patterns that ended in a final chat interaction would always have an average dropout higher than they should. Interactions that have never been reached were also not taken into account. In this case, the dropout would be impossible. Since the average dropout of an interaction is calculated from the ratio between the number of times the user left the chat in that interaction and the total number of sessions that went through the interaction, the numerator and denominator of this fraction will be 0. Since the denominator of the fraction is 0, then the result will be impossible. The consideration of this type of interaction would result in incorrect values and would eventually mislead the results.

It is possible to notice that, in the measures described in the current chapter, the order of the subtraction operands is the opposite of the measures related to the indicators of interest. As mentioned earlier, a positive deviation represents a positive consequence for the chat. In the case of the indicators of interest, a deviation from a pattern is positive if $p(I|P)$ is greater than $p(P)$. However, in the case of the dropout, the average dropout of a pattern being higher than the average dropout of the chat translates into a negative consequence for the chat. A pattern with an average dropout greater than the average dropout of the reference population is negative for the company. It should be noted that, regardless of the order of the operands, the absolute value of the subtraction is the same.

Formula $Global_{MIN\_Dropout}$ (Eq. 3.10) addresses the calculation of the interest of a pattern in a slightly different way. In Formula $Global_{AVG\_Dropout}$ (Eq. 3.9) the deviation of the pattern is the difference between the chat and pattern average dropouts. However, in Formula $Global_{MIN\_Dropout}$, the deviation of a pattern is calculated from the minimisation of the absolute value of the difference between the dropout of all pattern interactions and the average chat dropout. In this way, if $DROP$ represents the average dropout of the chat and the pattern under study is $P =< A,B,C >$, the value relative to the deviation of pattern P will be $min(|DROP-A|,|DROP-B|,|DROP-C|)$. Due to the fact that the same importance should be given to a positive or negative deviation, the absolute value of the difference and not the actual value is taken into account. From this minimum value it is guaranteed that, for a pattern to be considered interesting, all its nodes have a significant deviation from the average chat dropout.

Both formulas benefit smaller patterns. In Formula $Global_{AVG\_Dropout}$ (Eq. 3.9), the larger a pattern is, the more likely the average dropout of the pattern is to resemble the average chat dropout. In Formula $Global_{MIN\_Dropout}$ (Eq. 3.10), for a pattern to stop being considered interesting, it is enough to add an interaction that has a dropout similar to the chat average.

In addition to these measures, it was also considered the calculation of pattern deviation from the maximum difference instead of the minimum. However, this measure was excluded. From the maximum deviation, it was only necessary that a pattern had a node with a dropout significantly different from the average chat dropout to be considered interesting. In this way, having a pattern with an interesting node and adding multiple nodes with a dropout closer to the dropout of the reference population, makes no difference in the result of the deviation. Although the pattern became less interesting, its deviation was not changed.

Regarding the discovery of patterns with interesting dropouts relative to the local reference, the following quality measure was formulated:

$$Local_{AVG\_Dropout}(P) = |Dropout(P[1]) - \frac{\sum_{i=1}^{length(P)} Dropout(i)}{length(P)}| \times p(P) \tag{3.11}$$

This measure are very similar to ther measure $Global_{AVG\_Dropout}$ (Eq. 3.9). The reference population is the only different point. Since the local reference of a pattern is its initial interaction, the average chat dropout in formula $Global_{AVG\_Dropout}$ was replaced by the average dropout of the pattern input interaction in measure $Local_{AVG\_Dropout}$ (Eq. 3.11).

It should be noted that, contrary to the global reference, for the local reference, a formula based on minimising the dropout difference was not considered. If this formulation were taken into account, the first interaction of the pattern could not be considered for calculating the difference. This decision was due to the fact that, if the first interaction were considered, the value of the first part of the formula would always be 0. Since we would be minimising the absolute value of a difference where the same value exists on both sides of the operator, the result of the difference would always be 0. With this in mind, the interest of a pattern would always be 0 as well. If we were faced with a pattern with 2 interactions, one of them being a final chat interaction, this pattern could not be evaluated. In this way, this pattern would benefit longer patterns. This formulation would not work well for all kinds of patterns. As a result, this measure was not added to the local dropout.

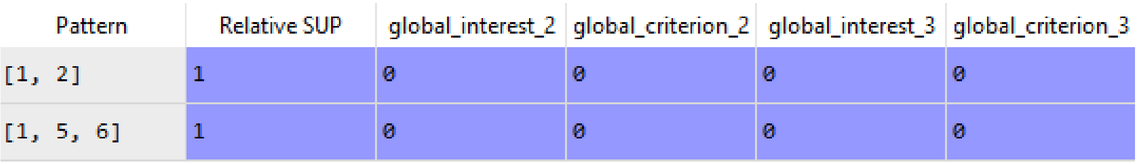
As a way of verifying the results returned by the described measures, a preliminary analysis of the results was made. This analysis is presented below.

3.5 Preliminary Analysis with Artificial Data

For the validation of the results obtained by the measures presented in section 3.4, four datasets were created. For a systematic analysis, all datasets were run with the minimum constraints of the input parameters listed in Section 3.3. All the experiments described in this section were run with a minimum support of 0.1 (10%), without maximum interval between interactions, minimum pattern length equal to 2 and without maximum pattern length.

The first test dataset was generated with the purpose of not including any interesting pattern, both regarding indicators of interest and dropout. In this dataset all users went through the same sequence of interactions. There are 6 interactions in total. The designations of the interactions are '1a', '2a', '3a', '4a', '5a' and '6a'. Four sessions were created in which the sequence of interactions covered by all users was <'1a', '2a', '3a', '4a', '5a', '6a'>. There are two indicators of interest that point to interactions '2a' and '3a'.

In total, 57 patterns were found. As expected, all patterns found were not considered interesting. In this way, all presented an interest value equal to 0. Figure 3.3 shows part of the results obtained by the quality measure 3.7. This measure aims to discover interesting patterns in relation to an indicator of interest taking into account the global reference.



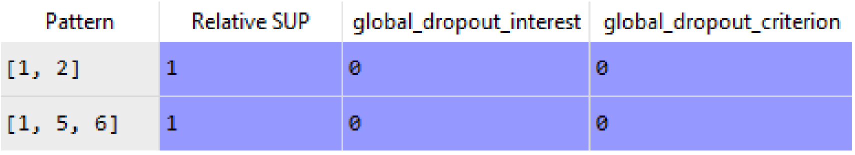
Pattern	Relative SUP	global_interest_2	global_criterion_2	global_interest_3	global_criterion_3
[1, 2]	1	0	0	0	0
[1, 5, 6]	1	0	0	0	0

Figure 3.3: Results obtained by the quality measure 3.7 with the test dataset 'test1'.

The 'Pattern' column shows the illustrative sequence of each pattern found. For a more intuitive interpretation of the results, each interaction is represented by its identifier and not by its name. The 'Relative SUP' column displays the support of the pattern. In this case, all patterns have the same support, 1, which is the maximum support. As previously mentioned, two indicators of interest were considered. This indicators were represented by their identifiers, 2 and 3. In this way, the columns 'global_interest_2' and 'global_criterion_2' represent the value of the deviation present in measure $Global_{Interest}$ (Eq. 3.7) (first part of the formula) and the final result of the formula, respectively. In this case, the indicator of interest was the indicator with identifier equal to 2. The same explanation fits the indicator with the identifier 3. It is possible to verify that all patterns have a deviation equal to 0. Because of this, the interest of all patterns is 0. In other words, there is no interesting pattern.

The results obtained with the same dataset and the measure $Local_{Interest}$ (Eq. 3.8) showed that there is also no interesting pattern with respect to the local reference. This was also expected, since users who start the pattern and those who cross the whole pattern always have the same distribution.

No pattern was also found with an interesting dropout, either relative to the global or local reference. Figure 3.4 shows the result of two patterns found. The same result were obtained for all other patterns. This measure aims to discover patterns with an interesting dropout as far as the global reference is concerned.

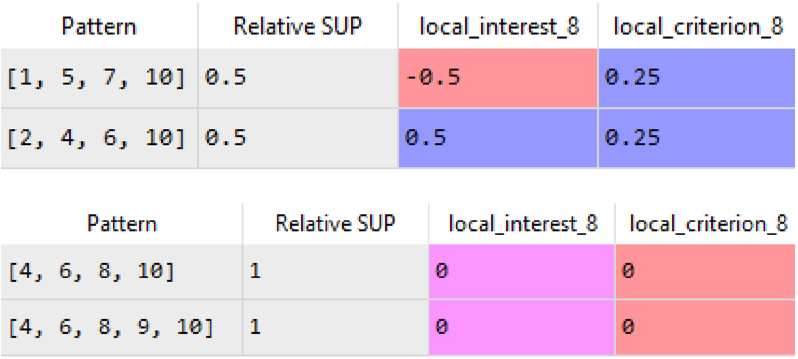


Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[1, 2]	1	0	0
[1, 5, 6]	1	0	0

Figure 3.4: Results obtained by the quality measure 3.9 with the test dataset 'test1'.

Later, another dataset was created with the purpose of testing the quality measures related to an indicator of interest. These data are intended to show that a positive or negative deviation from the reference population with the same absolute value has the same influence in the interest of a pattern. For this dataset, ten interactions and ten sessions were created. This chat has only one indicator of interest. This indicator is associated with the interaction with the identifier 8. Of the 10 existing sessions, 50% go through the sequence of interactions $< 1,2,3,5,7,9,10 >$ and 50% the sequence of interactions $< 1,2,3,4,6,8,9,10 >$.

In this way, in terms of support, it is expected that all the patterns found that contain the interactions 4,5,6,7 or 8 have a support equal to 0.5. In addition to having a support equal to 0.5 it is likely that its deviation from the reference population is equal to 0.5 or -0.5. The deviation should be equal to 0.5 in the patterns that contain the interactions 4, 6 and 8. Since whenever these interactions appear the interaction 8 is reached, the probability of the interaction 8 happens, given that the pattern under study contains one of the interactions 4,6 and 8, is equal to 1. Given that the probability of indicator 8 is 0.5, it is expected that these patterns will have a positive influence on the indicator equal to 0.5 (1-0.5). Likewise, if we are faced with a pattern that contains the interactions 3 and 7, the user will not reach the indicator 8. Therefore, these patterns will have a negative influence on this indicator. More precisely, an influence equal to -0.5 (0-0.5). This corresponds to a negative deviation. As noted in Section 3.4.1, a pattern containing the indicator of interest is not interesting. Hence, all patterns containing the interaction with the identifier 8 have an interest equal to 0. The results described are shown in Fig. 3.5.



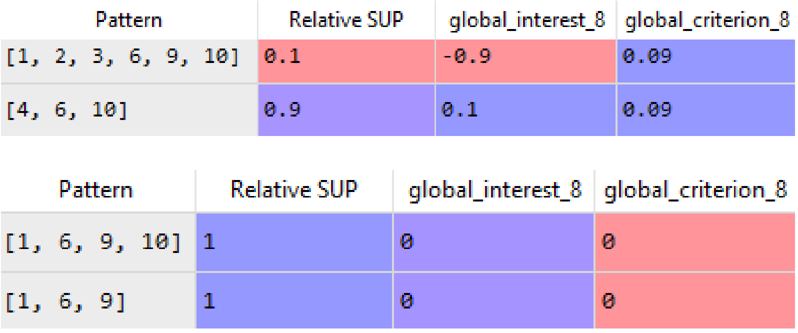
Pattern	Relative SUP	local_interest_8	local_criterion_8
[1, 5, 7, 10]	0.5	-0.5	0.25
[2, 4, 6, 10]	0.5	0.5	0.25

Pattern	Relative SUP	local_interest_8	local_criterion_8
[4, 6, 8, 10]	1	0	0
[4, 6, 8, 9, 10]	1	0	0

Figure 3.5: Results obtained by the quality measure 3.8 with the test dataset 'test2'.

In order to verify that the support and the deviation of the pattern have the same influence in its final interest,, another dataset was created. As in the previous dataset, this test dataset has ten interactions and ten sessions. In this dataset nine sessions are the same. These sessions are represented by the sequence $< 1,2,4,6,8,9,10 >$. One session is different from the rest. This session is represented by the sequence $< 1,2,3,6,7,9,10 >$. Patterns contained only in the first sequence are expected to have a support of 0.9, while patterns contained only in the second sequence have a support of 0.1. In patterns consisting only of interactions present in both sequences, the support will be 1. In this way, it is also expected that the reverse happens in terms of the deviation. Thus, it is expected that the patterns contained only in the first sequence have an absolute deviation of 0.1. Similarly, it is also expected that the patterns contained only in the second sequence have an absolute deviation equal to 0.9. In patterns consisting only of interactions present in both sequences, the deviation should be 0. In this way, only interests equal to 0.09 (0.9*0.1 or 0.1*0.9) or 0 (0*1) should be obtained.

In Fig. 3.6 it is possible to visualise both cases. Note that patterns containing interactions that are present only in the first sequence (interaction 4) will have a positive deviation, since only this sequence passes through the indicator of interest. Likewise, patterns containing interactions that are present only in the second sequence (interaction 3) will have a negative deviation, since they never reach the indicator. This can be seen in the first two patterns of Fig. 3.6. This dataset also shows the trade-off between the support and the deviation of a pattern. The higher the support the lower the deviation and vice versa.



Pattern	Relative SUP	global_interest_8	global_criterion_8
[1, 2, 3, 6, 9, 10]	0.1	-0.9	0.09
[4, 6, 10]	0.9	0.1	0.09

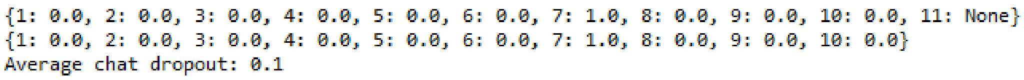
Pattern	Relative SUP	global_interest_8	global_criterion_8
[1, 6, 9, 10]	1	0	0
[1, 6, 9]	1	0	0

Figure 3.6: Results obtained by the quality measure 3.7 with the test dataset ’test3’.

Finally, in order to evaluate the results obtained according to the measures related to the dropout, a fourth dataset was created. To make calculations easier, this dataset has 11 interactions. The designations of the interactions are ’1a’, ’2a’, ’3a’, ’4a’, ’5a’, ’6a’, ’7a’, ’8a’, ’9a’, ’10a’ and ’11a’. One is a final chat interaction. In this way, this interaction (’11a’) is not taken into account for the calculation of the average dropouts, as explained in Chapter 3.4.2. This test dataset has 10 sessions. Of which, 50% are represented by the sequence $< 1,2,3,4,6,8,9,10,11 >$ and the other 50% by the sequence $< 1,2,3,5,7 >$.

From Fig. 3.7 it is possible to visualise the average dropout of each node. It is also possible to verify that node 11 has a dropout set to ’None’. This is due to the fact that node 11 is a final chat node.

Given the test data, it is expected that a pattern containing node 7 will be a negative pattern. In addition, it is also expected that the most interesting dropout pattern should contain it. In this

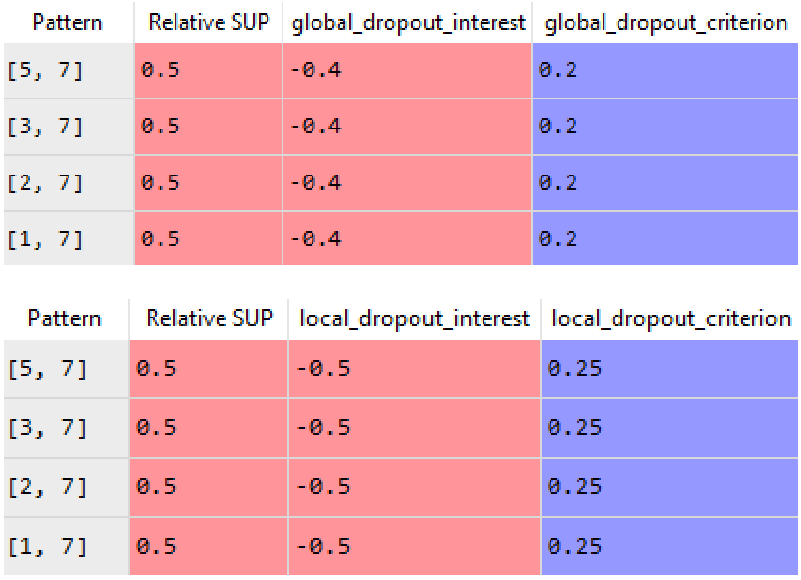


```
{1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0, 5: 0.0, 6: 0.0, 7: 1.0, 8: 0.0, 9: 0.0, 10: 0.0, 11: None}
{1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0, 5: 0.0, 6: 0.0, 7: 1.0, 8: 0.0, 9: 0.0, 10: 0.0}
Average chat dropout: 0.1
```

Figure 3.7: Average dropout of chat and all nodes belonging to it.

way, a pattern that contains node 7 should have a negative interest. Since node 7 is not a final chat node and has a dropout of 1, a pattern containing it has a dropout that is higher than the average chat dropout. The same is expected for the average input dropout of the pattern. A dropout higher than the reference population should have a negative interest.

In Fig. 3.8 it is possible to visualise the patterns with the greatest interest in terms of average dropout. The results for the global and local references are presented. We can verify that the expected results were obtained.



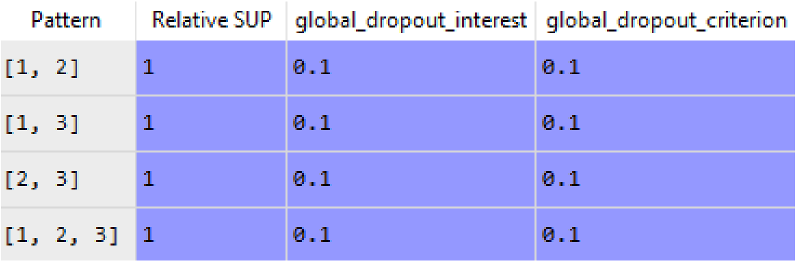
Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[5, 7]	0.5	-0.4	0.2
[3, 7]	0.5	-0.4	0.2
[2, 7]	0.5	-0.4	0.2
[1, 7]	0.5	-0.4	0.2

Pattern	Relative SUP	local_dropout_interest	local_dropout_criterion
[5, 7]	0.5	-0.5	0.25
[3, 7]	0.5	-0.5	0.25
[2, 7]	0.5	-0.5	0.25
[1, 7]	0.5	-0.5	0.25

Figure 3.8: Results obtained by the quality measures 3.9 (above) and 3.11 (bellow) with the test dataset ’test4’.

However, in formula $Global_{MIN\_Dropout}$ (Eq. 3.10), the expected results were not obtained. This can be visualised from Fig. 3.9. Node 7 was expected to appear in the most interesting patterns. However, this did not happen. Looking back at the formula, it is known that the average dropout of the chat is a constant for each chat. In this case, the average chat dropout is 0.1. What is changing in the formula is the dropout of the pattern nodes. From the Fig., it is possible to perceive that the patterns that were considered more interesting were the initial patterns. In these, the dropout of each node is 0, so the difference value will be 0.1. In fact, all patterns have a deviation equal to 0.1. All nodes have an average dropout of 0, with the exception of node 7 that has an average dropout of 1. It should be noted that the minimum size of a sequence to be considered a pattern is equal to 2. In this way, minimising the difference between the average dropouts will always be 0.1. This is due to the fact that in all patterns there is at least one node with an average dropout of 0. As | 0.1-0.9 | > | 0.1-0 | then the presence of node 7 in the patterns makes no difference. We may also notice that the initial patterns appear first because they have the maximum support.

Formula $Global_{MIN\_Dropout}$ was designed with the goal of being more restricted. For a pattern to be considered interesting all of its nodes had to be interesting as well. However, in cases where there is only one dropout node quite distinct from the reference population, this formula will not lead to satisfactory results.



Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[1, 2]	1	0.1	0.1
[1, 3]	1	0.1	0.1
[2, 3]	1	0.1	0.1
[1, 2, 3]	1	0.1	0.1

Figure 3.9: Results obtained by the quality measure 3.10 with the test dataset 'test4'.

3.6 Summary

This problem came about as part of a project carried out by INESC TEC and it was approached by a company that implement chatbots, SMARKIO. Chatbots are increasingly used in business contexts. With this increase also comes the need for a better and automated analysis of the flow of this type of interactions. This analysis can be performed from the analysis of the behavioural patterns of the clients. The perception of the improvements that must be made and the best practices that should be expanded in these types of systems can be translated into an opportunity to improve the quality of services.

The approach chosen to solve this problem was the combination of two areas of Data Mining, Sequential Pattern Mining and Subgroup Discovery. This approach allows the discovery of unusual behavioural patterns in chatbots users. Two variants of this approach were developed. In the first approach, an off-the-shelf SPM solution to discover frequent subsequences (SPAM algorithm) was extended. After the discovery of frequent subsequences, these were evaluated according to quality measures designed within the scope of this dissertation. One limitation with the previous approach is that for a pattern to be considered unusual, it must be considered frequent first. In this way, a second approach was developed. The algorithm used in the first approach to obtain the frequent patterns was modified. A beam search strategy was used instead of the depth-first search strategy used in the first approach. From the new search strategy, only the patterns with the highest values for the quality measure are considered and continue to be extended. In this way, the search for unusual patterns in the second approach is done alternately.

In addition to the approaches designed and implemented to find unusual patterns in sequential data, five different quality measures were developed. These measures are intended to verify which patterns are interesting in relation to an indicator. An indicator can be of two types. It can be an indicator of interest or a dropout indicator. For each type of indicator, two types of reference populations were taken into account for what is considered normal, the global reference (total population) and the local reference (users who initiated a pattern).

From the preliminary analysis of the results obtained with artificial data, it was possible to verify that all the results obtained by the different quality measures were as expected, with the exception of measure $Global_{MIN\_Dropout}$ (Eq. 3.10). In cases where there is only one dropout node quite distinct from the reference population, this formula will not lead to satisfactory results.

Chapter 4

Experimental Setup

This chapter presents the case study used for the implementation and evaluation of the results of this dissertation. The data provided and the experiences produced are also described. In addition, the algorithm hyperparameters and the different combinations tested during the experiments are specified. Finally, the resources and technologies used for the implementation are presented.

4.1 Data

The data used for the analysis and evaluation of the results of this dissertation was provided by a company which develops chatbots. As already mentioned in Section 3.1, this company is called Smarkio. Created in 2015, Smarkio (Sales, Marketing, Integration, and Optimisation) aims to create cloud-based marketing solutions by combining a marketing automation platform (MAP) with chatbots [Smarkio, 2018]. This project came about with the goal of helping teams belonging to Smarkio in terms of decision support. In order to understand what should be improved, a chatbot needs constant monitoring and analysis. With this is mind, the company wanted to find a way to monitor the performance of their chatbots, according to certain business metrics. For Smarkio, there are two types of important business metrics: metrics relating to an interesting interaction for the company and dropout metrics.

In addition to the data created for testing and for the preliminary analysis of the results presented in Section 3.5, data provided by Smarkio were also used. These data were used to generate the results presented in Chapter 5. Each dataset provided by Smarkio includes information about the indicators of interest, the chat structure and the session logs. While Smarkio made several datasets available, only three met the necessary conditions for the analysis. Datasets where the chat structure and respective logs were not compatible were disregarded. This may be due to the incompatibility of versions, logs that include interactions that do not exist in the file related to the chat structure, transitions existing in the logs that are not allowed by the chat structure, among others. In addition, datasets which did not have any interest indicators defined by the company were also not taken into account. In this way, three datasets will be presented during the current chapter.

The indicators of interest related to the datasets under analysis, defined by Smarkio, are focused mainly on the collection of information. Acceptance of terms, provision of emails, addresses, and other personal data are some examples of indicators of interest that have been studied.

The structure of a chat encompasses all the nodes of the system as well as all the possible transitions between them. Note that in the provided datasets, a node may not be an interaction. However, an interaction is always represented by a node. For example, if there is a node where the user is asked about their address, this node is considered an interaction. However, if, for example, the subsequent nodes to this interaction are checks of the provided address (i.e. having more than 5 letters, having numbers, among others), these nodes are not considered interactions. A node corresponds to a system stage. An interaction corresponds to a node where there is an interaction with the user. In order to not lose information about user behaviour, all nodes were considered, regardless of whether they were interactions or not.

The structure of a chatbot is used for checking the sessions. In case a session does not respect the chat structure, it is discarded. Some inconsistencies and redundant data were detected in the chatbots structure. From the session logs it was possible to obtain the sequence of node transitions, made by each user. Some inconsistencies were also detected in these data. These found inconsistencies were addressed in the data preparation phase.

The data provided by Smarkio, as well as the structure of the files regarding the indicators, the chat structure and the session logs, are more detailed in Section B.1.

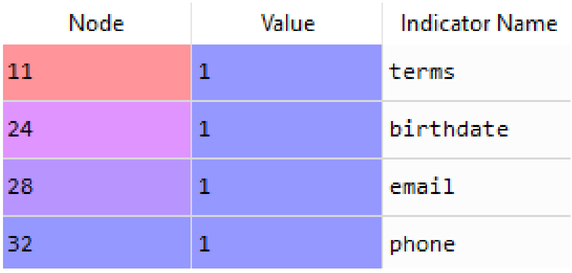
As mentioned previously, three sets of data were evaluated in this dissertation. In order to protect the data provided by the company, the original names of each chat were replaced by fictitious names representing the chat business domain. The attribution of fictitious names was based on the purpose of each chat. This way, the names of the chats under study are "Credit", "Christmas" and "Employment". In the next section, the different chats are presented according to their characteristics.

4.1.1 Chatbot "Credit"

The chatbot "Credit" has the purpose of allowing users to ask for credit For this reason, the bot questions the user about some personal information.

In total, this chat has 38 nodes, of which 4 interactions are considered interesting by the company. Therefore, there are 4 indicators of interest defined for this chat. The information about the indicators of interest is shown in figure 4.1. The indicators point to a terms acceptance node and to data collection nodes. In this case, the personal data that are intended to be collected are the user's date of birth, email address and mobile phone number.

The structure of this chat is illustrated in Figure 4.2. Each circle represents a chat node. Each connection between two nodes is represented by the line connecting these nodes. In addition, each node has an associated colour, which represents its type. In this dissertation, however, the type of the nodes was not taken into account. For the sake of better understanding the chat structure, the red coloured node represents a bifurcation. Nodes associated with indicators of interest (i.e. nodes with the identifiers 11, 24, 28 and 32 - Fig. 4.1) are highlighted in yellow and with their identifier.



Node	Value	Indicator Name
11	1	terms
24	1	birthdate
28	1	email
32	1	phone

Figure 4.1: Interest indicators for chatbot "Credit".

From the structure of the chat, we can recognise that it only contains one bifurcation. Thus, of the 38 existing nodes, only one has more than one possibility for following nodes. This chat has 615 sessions, which means this number of users used this chat.

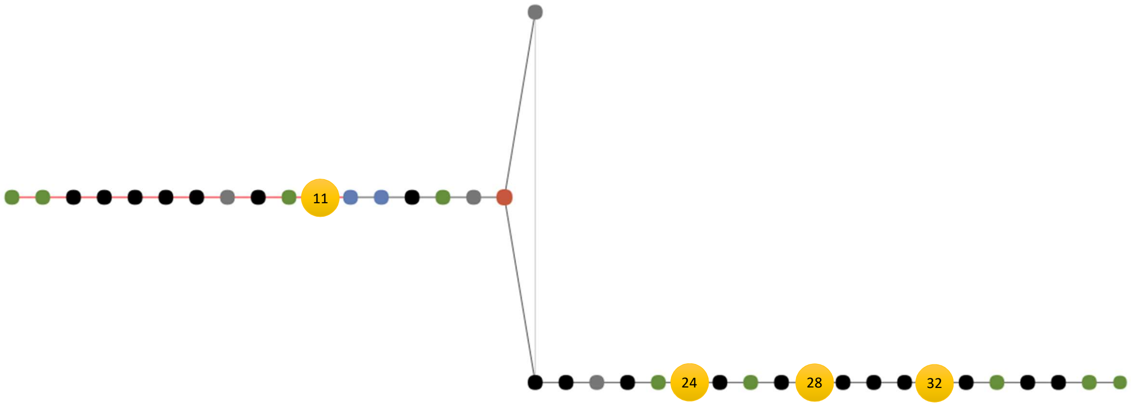


Figure 4.2: Structure of the chatbot "Credit".

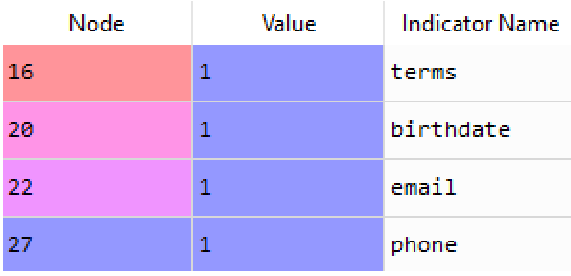
4.1.2 Chatbot "Christmas"

The chat "Christmas" is a chat for gifts suggestions. In this chat, the user intends to get suggestions for gifts to give for Christmas. As in the previous chat, the bot questions the user about some personal information.

The types of interactions considered interesting by the company in the "Credit" chat are repeated in the "Christmas" chat. Figure 4.3 shows the interest indicators for this chat.

The structure of this chat is shown in Fig. 4.4. Nodes associated with indicators of interest (i.e. nodes with the identifiers 16, 20, 22 and 27 - Fig. 4.3) are highlighted in yellow and with their identifier. In total, this chat has 33 nodes, of which two interactions are bifurcations.

There were 705 users who started this chat. Of those 705, 308 did not went beyond the first three interactions. This means that 44% of all users have only reached at most the third chat interaction.



Node	Value	Indicator Name
16	1	terms
20	1	birthdate
22	1	email
27	1	phone

Figure 4.3: Interest indicators for chatbot "Christmas".

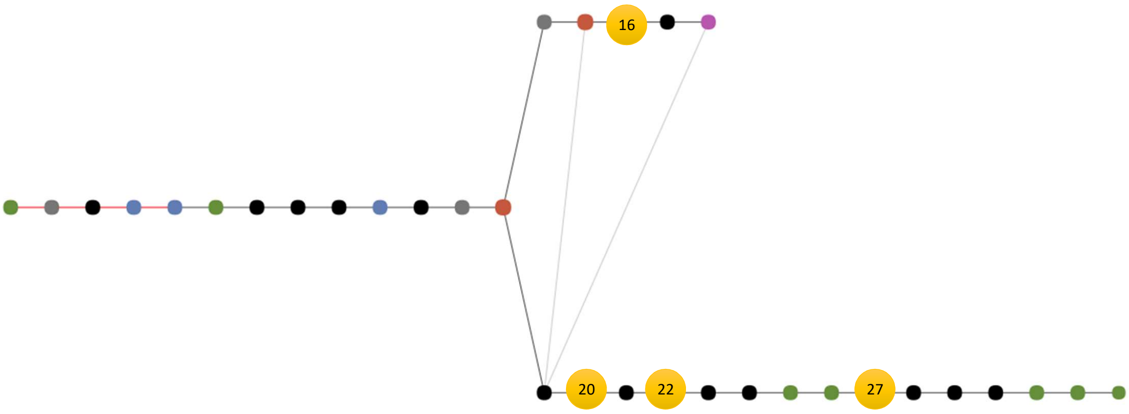
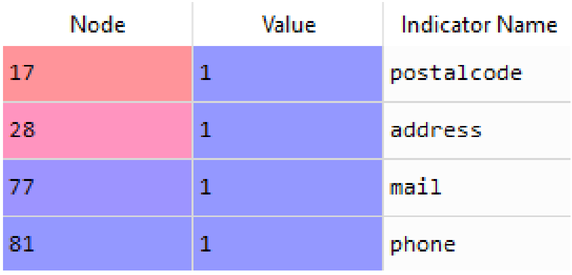


Figure 4.4: Structure of the chatbot "Christmas".

4.1.3 Chatbot "Employment"

Finally, the chat "Employment" is used by the user with the purpose of enrolling in a course or job. In total there are 87 interactions, of which 32 are bifurcations. Of these, 4 are considered interesting by the company. This way, Smarkio intended to understand the exceptional behavioural patterns associated with the indicators listed in Figure 4.5. As can be seen, all the indicators are related to data collection, namely, the postal code, address, email and mobile phone number of the user. It is also relevant to note that this chat contains 2980 sessions.



Node	Value	Indicator Name
17	1	postalcode
28	1	address
77	1	mail
81	1	phone

Figure 4.5: Interest indicators for chatbot "Employment".

From Figure 4.6, it is possible to verify that the structural complexity presented in this chat is superior to the previous ones. Nodes associated with indicators of interest (i.e. nodes with the identifiers 17, 28, 77 and 81 - Fig. 4.5) are highlighted in yellow and with their identifier.

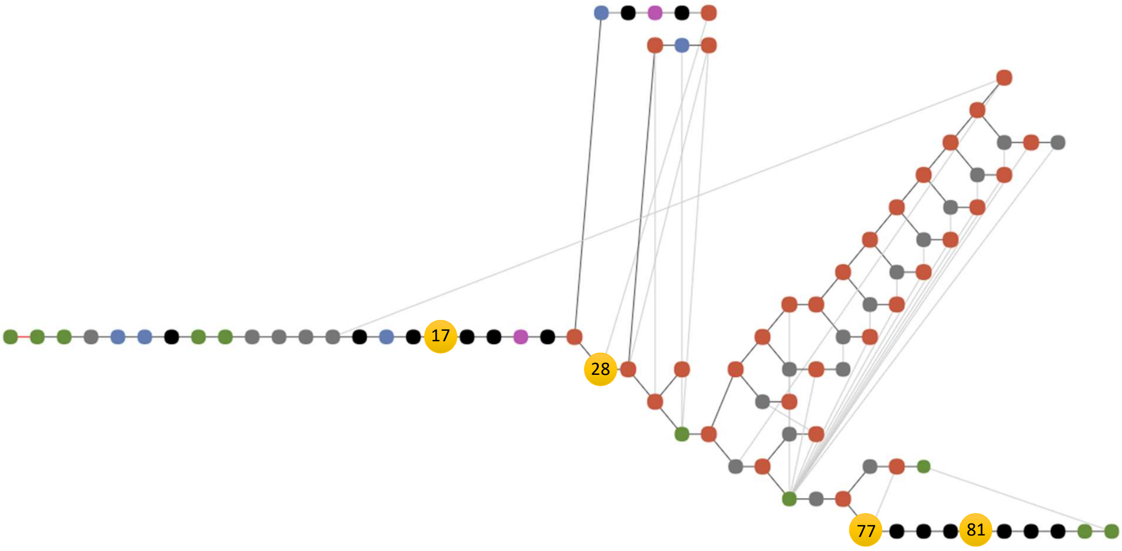


Figure 4.6: Structure of the chatbot "Employment".

The number of possible next nodes in all existing bifurcations is always two, for all the chats. Therefore, although the chats presented do not have a flat structure, they have the minimum number of possible possibilities at each bifurcation. A chat that has a flat structure is a chat with no bifurcations.

4.2 Hyperparameters

Both approaches developed within the scope of this dissertation have a set of input parameters, which are:

- **max_gap** - Maximum interval between sequence items (maximum gap). The minimum value is 1.
- **min_length** - Minimum pattern length. The minimum value of this parameter is 2, so that it is possible to consider a sequence as a pattern.
- **max_length** - Maximum pattern length. By default there is no maximum length.
- **min_sup** - Minimum frequency or minimum support with which a pattern must occur to be considered frequent.
- **remove_flat_patterns** - Possibility of removing, or not, flat patterns. Flat patterns are patterns that do not contain any bifurcation.
- **quality_measure** - Measure of interest to be used. As previously mentioned, the quality measures can be related to an indicator of interest, the global reference (Eq. 3.7) or the local reference (Eq. 3.8). They can also be relative to the average or minimum global dropout (Eq. 3.9 and Eq. 3.10) or relative to the average local dropout (Eq. 3.11).

In addition to the parameters described above, the second approach also includes the following ones:

- **interest_indicator** - Indicator of interest, if the measure of interest is relative to indicators of interest. In the first approach, the measure of interest for all the indicators of interest for all the patterns is calculated. However, in this approach it is necessary to define the indicator of interest, so that only the most interesting N patterns in relation to that indicator are passed to each level, where N is the beam width.
- **beam_width** - Beam width. This width corresponding to the number of patterns that go to the next phase of the search, as previously mentioned in chapter 3.3.2.

In order to analyse the impact of each hyperparameter and to analyse those that best fit each dataset, several experiences were made. In Table 4.1 it is possible to observe the different combinations of hyperparameters tested in both approaches in order to analyse the results. Note that the last column refers only to the second approach.

4.3 Implementation

The technology used to implement the approaches previously described in Section 3.3 was *Python*[1]. The most important *Python* libraries which were used in the development of both approaches were

[1] https://www.python.org/

Table 4.1: Variations in the hyperparameters of the approaches for the analysis of the results.

Maximum Gap	**Remove flat patterns**	**Minimum Pattern Length**	**Maximum Pattern Length**	**Minimum Frequency**	**Beam Width**
1 and None	Yes and No	2, 3, 4 and 6	3, 4, 6, 10 and None	0.1, 0.2, 0.3, 0.4 and 0.5	1, 5 and 10

NumPy[2] and *Pandas*[3]. The former is the main package for scientific computing of *Python* with the latter being a software library programming language for analysis and data manipulation. The IDE used for the development of the project was *Spyder*[4].

Another important used tool was SPMF[5]. *SPMF* is an open-source Data Mining Java library which contains Sequential Pattern Mining algorithms. It was used to check the results obtained by the implementation of the base algorithm, SPAM.

It is also relevant to mention that the source code, the datasets and the results of this project are hosted on *Github*[6].

4.4 Summary

The data used for the analysis and evaluation of the results presented in this dissertation came from Smarkio, a company that develops chatbots. The interactions between bot and user were evaluated according to multiple indicators defined by that same company. As mentioned in Chapter 3, there are two different types of indicators, of interest and of dropout. Dropout indicators refer to the percentage of people who leave the chat on a particular node. This indicator refers to all nodes in the chatbot. The indicators of interest are pointers to interactions that the company finds interesting.

Three different datasets were studied: chatbot "Credit", "Christmas" and "Employment". The chatbot "Credit" is a chat for users to get credit. In the chatbot "Christmas", the user can get suggestions for gifts to give for Christmas. Finally, the chat "Employment" is used by the user with the purpose of enrolling in a course or job.

The implemented algorithms have several input hyperparameters. In order to analyse the impact of each one and to analyse those that best fit each dataset, several experiences were made. For each, the following hyperparameters were varied: maximum gap between interactions, a pattern having ou not at least one bifurcation, minimum pattern length, maximum pattern length and minimum frequency. In addition to these hyperparametres, for the second approach the beam width was also varied.

The technology used in this dissertation was *Python* for the implementation of both approaches. The Data Mining *SPMF* library was also used to test the results obtained when implementing the

[2] http://www.numpy.org/

[3] https://pandas.pydata.org/

[4] https://pythonhosted.org/spyder/

[5] http://www.philippe-fournier-viger.com/spmf/

[6] https://github.com/CatarinaAmaral/Subgroup-Discovery-for-Sequences

SPAM algorithm. Lastly, the IDE used for the development of the project was *Spyder* and the source code of this project is hosted on *Github*.

Chapter 5

Results

In the current chapter, the results of the experiments carried out in the scope of this dissertation are described. The results are grouped by chat and, for each chat, according to the type of indicator and the reference population. It should be noted that, for each combination *Indicator - Measure of Interest* only the patterns with the highest score were analysed.

The results of chatbot "Employment" will be analysed first. This is due to the fact that this chatbot has led to the most interesting results, since it is also the chatbot with the most complex structure. Concerning chatbots "Credit" and "Christmas", only the most interesting aspects will be presented. These chatbots are described in more detail in Annex C and D.

5.1 Chatbot "Employment"

A user uses the chatbot "Employment" with the purpose of enrolling in a course or applying for a job. Compared with the previously presented chatbots, this one has a more complex structure, as it is possible to see from Fig. 4.6.

In order to have a clearer understanding of the indicators of interest, Table 5.1 lists the probabilities of each indicator/interaction happening.

Table 5.1: Probability of the indicators of interest for chat "Christmas"

Indicator	**Probability (%)**
postalcode (17)	47
address (28)	37
email (77)	26
phone (81)	23

Below are the most interesting patterns discovered for each measure of interest in the current chatbot.

5.1.1 Indicator of Interest & Global Reference

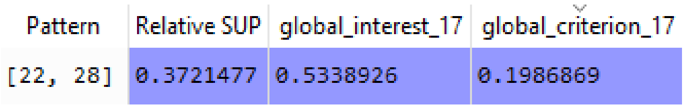
When running the program with this measure for the current dataset, it was possible to verify that, regardless of the indicators of interest, many patterns were found with the maximum score.

It was possible to verify that, for all the indicators, the length of the patterns with maximum score varied. However, it was also possible to verify that the larger patterns contained the smaller ones. This is due to the fact that a user who takes a particular decision tends to follow a certain path according to that response. Each bifurcation has only two possibilities of following nodes, which is something that also contributes to that conclusion. With this in mind, by decreasing the maximum length of the patterns to 2, many unnecessary patterns are discarded.

After restricting the length of the patterns, it has been found that removing flat patterns also decreases the number of unnecessary results. As previously stated, a flat pattern is a pattern that does not contain nodes that are bifurcations or nodes that result from bifurcations.

From the above restrictions it was possible to obtain the following results for each indicator of interest:

- **Indicator "postalcode" (17)** - The most interesting pattern found was the pattern $< 22, 28 >$. This pattern is a pattern that occurs after node 17. Therefore, this pattern does not show its influence on node 17, but rather the influence of node 17 on this pattern. Since the user has not reached any node of the pattern $< 22, 28 >$ before reaching the interaction with identifier 17, it is not possible that the pattern influences the behaviour of the user in the interaction. This pattern means that users who provide the postal code tend to provide their address as well. From Fig. 5.1 we can see more details about the influence of node 17 in this pattern. A user who supplies his postal code is 53 p.p. more likely to provide his address as well, compared to other users.

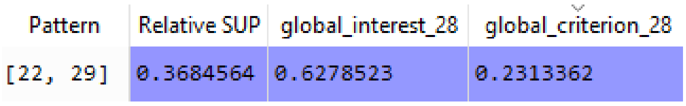


Pattern	Relative SUP	global_interest_17	global_criterion_17
[22, 28]	0.3721477	0.5338926	0.1986869

Figure 5.1: Most interesting pattern found for the indicators "postalcode" with the quality measure "Indicator of Interest & Global Reference" (Eq. 3.7), regarding the chatbot "Employment".

- **Indicator "address" (28)** - The most interesting pattern found was the pattern $< 22, 29 >$. By analysing the interactions of this pattern, it was possible to conclude the same as in the previous indicator (17). From Fig. 5.2, it is possible to conclude that who correctly supplies their postal code, is also more likely to provide their address with an additional 63 p.p. probability.

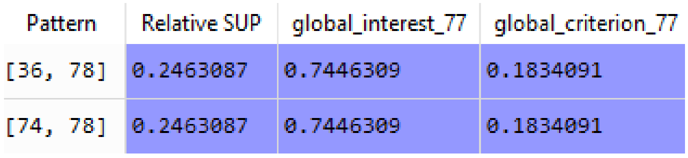
- **Indicator "mail" (77)** - Regarding this indicator of interest, two patterns with the highest value were obtained, $< 36, 78 >$ and $< 74, 78 >$. In this case, the highest value obtained was approximately 0.18. From the analysis of the interactions of the first pattern, it is possible to verify that, who intends to enroll courses in accounting, management or finance have a



Pattern	Relative SUP	global_interest_28	global_criterion_28
[22, 29]	0.3684564	0.6278523	0.2313362

Figure 5.2: Most interesting pattern found for the indicator "address" with the quality measure "Indicator of Interest & Global Reference" (Eq. 3.7), regarding the chatbot "Employment".

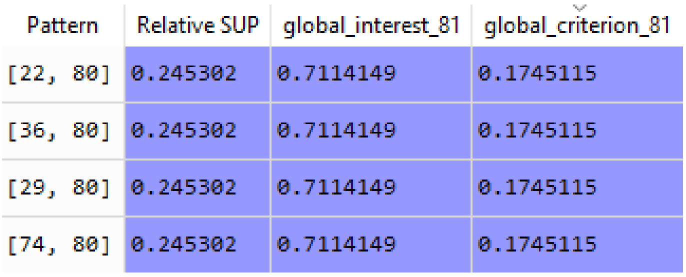
probability 74 p.p. higher of giving their email. The second pattern shows that someone who agrees to receive direct marketing communications is also 74 p.p. more likely to give their email compared to other users. These patterns can be seen from Fig. 5.3.



Pattern	Relative SUP	global_interest_77	global_criterion_77
[36, 78]	0.2463087	0.7446309	0.1834091
[74, 78]	0.2463087	0.7446309	0.1834091

Figure 5.3: Most interesting patterns found with indicator "mail" with the quality measure "Indicator of Interest & Global Reference" (Eq. 3.7), regarding the chatbot "Employment".

- **Indicator "phone" (81)** - This indicator is one of the last interactions of this chatbot. Several patterns were found with the highest value obtained for this measure of quality with this indicator. All the patterns found have the same support and the same deviation from the total population. The patterns found were $< 22, 80 >$, $< 29, 80 >$, $< 36, 80 >$ and $< 74, 80 >$. The pattern $< 22, 80 >$ shows that who provides the zip code is 71 p.p. more likely to also provide the mobile phone number. From the pattern $< 29, 80 >$ it is possible to conclude that who provides the address is also more 71 p.p. likely to provide the telephone contact. The pattern $< 36, 80 >$ shows that, just as in the "mail" indicator, who intends to enroll in accounting, management or finance have an additional 71 p.p. of giving their phone too. The pattern $< 29, 80 >$ shows that who supplies the address also tends to provide the phone number. The results of the quality measure for these patterns can be seen from Fig. 5.4.



Pattern	Relative SUP	global_interest_81	global_criterion_81
[22, 80]	0.245302	0.7114149	0.1745115
[36, 80]	0.245302	0.7114149	0.1745115
[29, 80]	0.245302	0.7114149	0.1745115
[74, 80]	0.245302	0.7114149	0.1745115

Figure 5.4: Most interesting patterns found with the indicator "phone" with the quality measure "Indicator of Interest & Global Reference" (Eq. 3.7), regarding the chatbot "Employment".

The results obtained by the second approach with beam width equal to 1 in the first two indicators, 2 in the second indicator and 4 in the last one, were the same as in the first approach.

Table 5.2 lists the constraints required to obtain the interesting patterns for all the indicators mentioned above.

Table 5.2: Constraints needed to obtain the most interesting pattern for all the indicators of interest of the chatbot "Employment" and quality measure "Indicator of Interest & Global Reference" (Eq. 3.7).

Restriction	**Value**
Maximum pattern length	2
Remove flat patterns	Activated
Minimum support	$\leq$0.2
Beam width	4

5.1.2 Indicator of Interest & Local Reference

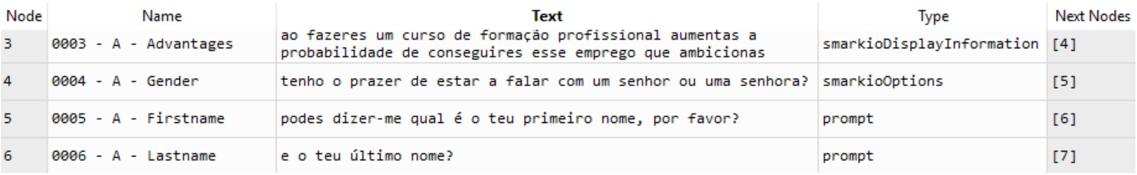
Initially, for a more comprehensive view of the results, the program was run with as few restrictions as possible. When analysing the results obtained for all the indicators defined by the company, it was possible to perceive that the most interesting patterns always ended in nodes subsequent to the indicator. There were several patterns with the highest value obtained for this measure of quality with this indicator.

In addition, all input nodes of the most interesting patterns were initial nodes of the chat (nodes 1,2,3 or 4). This is due to the fact that, in the beginning, there are more possible paths to where the user can go. This leads to the user having a lower probability of reaching a certain indicator if he/she is in an initial chat node.

Although these patterns make sense, they do not introduce new knowledge. In this way, the results with the restriction *max_gap=1* were evaluated. When analysing the results obtained for all the indicators defined by the company, it was possible to perceive that the most interesting pattern was always a very long pattern, with lengths around 15 nodes. Regarding Eq. 3.8, it is possible to understand why this happens. In this measure of interest, the reference population is the initial node of the pattern. With this in mind, the probability of reaching the indicator knowing that it passes through the pattern is compared to the probability of reaching the indicator knowing that the user starts the pattern. The longer the pattern, the more likely it is to reach the indicators. In order to try to decrease this trend, the maximum length of the pattern was limited to 4.

Following are the most interesting patterns found for each chat indicator.

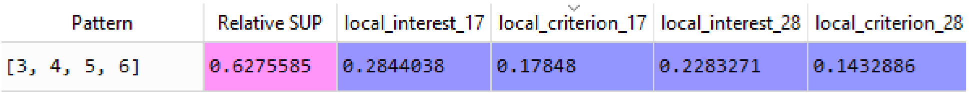
- **Indicators "postalcode" (17) and "address" (28)** - With the restrictions *max_gap=1* and maximum length equal to 4, the pattern with the highest interest for indicators 17 and 28 was the pattern $< 3,4,5,6 >$. From Fig. 5.5 it is possible observe the text shown to the user in the initial interactions of the current chat.



Node	Name	Text	Type	Next Nodes
3	0003 - A - Advantages	ao fazeres um curso de formação profissional aumentas a probabilidade de conseguires esse emprego que ambicionas	smarkioDisplayInformation	[4]
4	0004 - A - Gender	tenho o prazer de estar a falar com um senhor ou uma senhora?	smarkioOptions	[5]
5	0005 - A - Firstname	podes dizer-me qual é o teu primeiro nome, por favor?	prompt	[6]
6	0006 - A - Lastname	e o teu último nome?	prompt	[7]

Figure 5.5: Details about the first nodes of the chatbot "Employment".

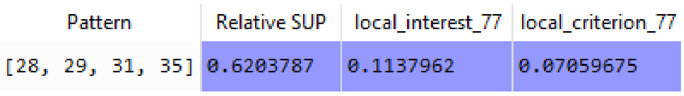
Using Fig. 5.6, we can conclude that a user passing through nodes 3, 4, 5 and 6 is 28 p.p. more likely to provide the postal code compared to all users who passed through node 2. From the text presented to the user in these interactions, it is possible to conclude that someone who is in node 3 has not yet provided any information, while those who have crossed the whole pattern have already provided at least their gender and first name. The same can be concluded for the indicator 28. The pattern $< 3,4,5,6 >$ also has a positive local influence on the "address" indicator. In this case, the deviation from users who start the pattern is approximately 23 p.p..



Pattern	Relative SUP	local_interest_17	local_criterion_17	local_interest_28	local_criterion_28
[3, 4, 5, 6]	0.6275585	0.2844038	0.17848	0.2283271	0.1432886

Figure 5.6: Most interesting pattern found for the indicators "postalcode" and "address" with the quality measure "Indicator of Interest & Local Reference" (Eq. 3.8), regarding the chatbot "Employment".

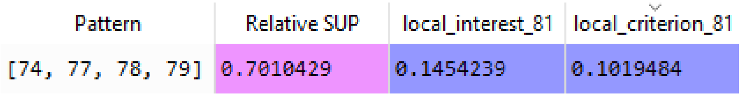
- **Indicator "mail" (77)** - For the indicator 77 the pattern with the highest interest was the pattern $< 28,29,31,35 >$. This pattern represents the users who entered an address correctly. Node 28 corresponds to the interaction related to the request and response of the address. Node 35 corresponds to the validation of the address. Nodes 29 and 31 are intermediate nodes. In this way, from Fig. 5.7 it is possible to verify that someone who enters a correct address on the first try is 11 p.p. more likely to provide the email as well.



Pattern	Relative SUP	local_interest_77	local_criterion_77
[28, 29, 31, 35]	0.6203787	0.1137962	0.07059675

Figure 5.7: Most interesting pattern found for the indicator "mail" with the quality measure "Indicator of Interest & Local Reference" (Eq. 3.8), regarding the chatbot "Employment".

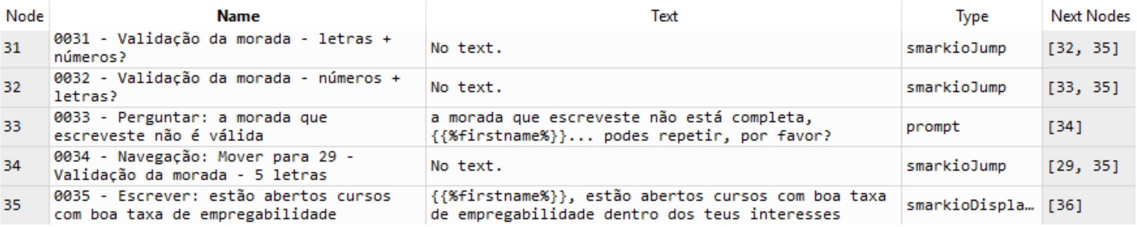
- **Indicator "phone" (81)** - For the indicator 81 the pattern with the highest interest was the pattern $< 74,77,78,79 >$. From this pattern it is possible to verify that the users who accept the terms the first time have a probability 15 p.p. higher of giving the phone number (Fig. 5.8). These 15 p.p. are relative to users who saw the message for acceptance of terms.



Pattern	Relative SUP	local_interest_81	local_criterion_81
[74, 77, 78, 79]	0.7010429	0.1454239	0.1019484

Figure 5.8: Most interesting pattern found for the indicator "phone" with the quality measure "Indicator of Interest & Local Reference" (Eq. 3.8), regarding the chatbot "Employment".

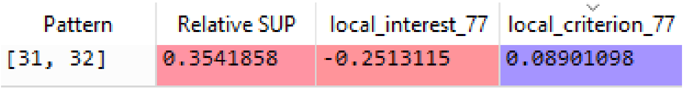
Regarding the second approach, with the parameter beam width equal to 1, the patterns originated by the second approach were not always the same as those of the first approach. For the indicators 17, 28 and 81 the most interesting pattern was the same. However, in the case of the indicator 77, the most interesting pattern with length 2 was $< 31, 32 >$. Nodes 31 and 32 correspond to two types of validations in the user's address. Information about these interactions can be seen in Fig. 5.9.



Node	Name	Text	Type	Next Nodes
31	0031 - Validação da morada - letras + números?	No text.	smarkioJump	[32, 35]
32	0032 - Validação da morada - números + letras?	No text.	smarkioJump	[33, 35]
33	0033 - Perguntar: a morada que escreveste não é válida	a morada que escreveste não está completa, {{%firstname%}}... podes repetir, por favor?	prompt	[34]
34	0034 - Navegação: Mover para 29 - Validação da morada - 5 letras	No text.	smarkioJump	[29, 35]
35	0035 - Escrever: estão abertos cursos com boa taxa de empregabilidade	{{%firstname%}}, estão abertos cursos com boa taxa de empregabilidade dentro dos teus interesses	smarkioDispla…	[36]

Figure 5.9: Details about nodes 31 and 32 of the chatbot "Employment".

From Fig. 5.10 it is possible to notice that this pattern has a negative local influence with respect to the "email" indicator (77). A user who does not check the validation presented at the node 31 and that passes to the validation presented in the node 32 has less 25 p.p. probability of giving the email. However, this pattern did not lead to the pattern with the best score. Changing the beam width to 5 has already made it possible to achieve the same results as in the first approach.



Pattern	Relative SUP	local_interest_77	local_criterion_77
[31, 32]	0.3541858	-0.2513115	0.08901098

Figure 5.10: Most interesting pattern found for the indicator "phone" with the quality measure "Indicator of Interest & Local Reference" (Eq. 3.8), regarding the chatbot "Employment".

Table 5.3 lists the constraints required to obtain the interesting patterns for all the indicators mentioned above.

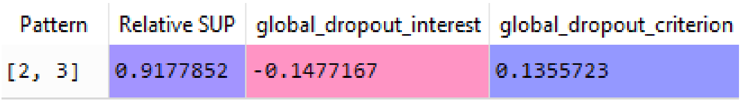
5.1.3 Average Dropout & Global Reference

As far as this measure (Eq. 3.9) is concerned, the most interesting pattern found for the current dataset was $< 2, 3 >$. This pattern was obtained by both the first and second approaches. In addition, this pattern was obtained with the minimum constraints for each approach (minimum support equal to 0.1 and minimal length of the pattern equal to 2).

Table 5.3: Constraints needed to obtain the most interesting pattern for all the indicators of interest of the chatbot "Employment" and quality measure "Indicator of Interest & Local Reference" (Eq. 3.8).

Restriction	Value
Maximum pattern length	4
Minimum support	$\leq$0.2
Maximum gap	1
Beam width	5

As it can be seen from Fig. 5.11, the probability of a user leaving chat in this pattern $< 2,3 >$ is approximately 15 p.p. higher than the average probability of a user leaving the chat in any interaction. The average dropout for the "Employment" chat is approximately 2%.



Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[2, 3]	0.9177852	-0.1477167	0.1355723

Figure 5.11: Pattern with the highest score of the quality measure "Average Dropout & Global Reference" (Eq. 3.9), regarding the chatbot "Employment".

Table 5.4 lists the constraints that could be introduced to obtain the pattern mentioned above.

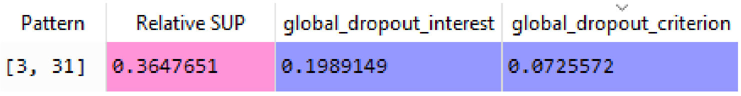
Table 5.4: Constraints required to obtain the most interesting pattern for the dropout indicator of the chatbot "Employment" and quality measure "Average Dropout & Global Reference" (Eq. 3.9).

Restriction	Value
Maximum pattern length	2
Minimum support	$\leq$0.9
Beam width	1
Maximum gap	1

5.1.4 Minimum Dropout & Global Reference

The most interesting pattern found for this measure was the pattern $< 3,31 >$. Node 3 is an informative interaction and is one of the initial interactions of the chatbot. Node 31 refers to users who provided an address with more than 5 letters. This node is not an interaction, it is a system node. In this case, it is a node related to the validation of the address entered by the user. This requirement is the first requirement for an address to be considered valid.

Based on Fig. 5.12, a user is at least 20 p.p. less likely to leave chat on the nodes of this pattern than on all other nodes in the chat. Node 3 is an initial node and any user who does not leave the chat at the beginning goes through this node. With this in mind, the pattern $< 3,31 >$ represents the users that respect the first validation of the address (address entered with at least 5 letters), since only node 31 differentiates in some way a user. Therefore, it is possible to conclude that, who places a valid address is less likely to leave the chat.



Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[3, 31]	0.3647651	0.1989149	0.0725572

Figure 5.12: Pattern with the highest score of the quality measure "Minimum Dropout & Global Reference" (Eq. 3.10), regarding the chatbot "Employment".

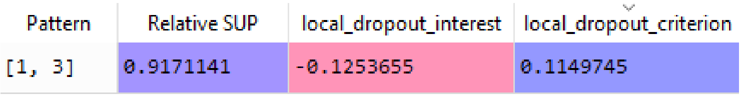
The second approach also led to the same result with beam width equal to 1. Table 5.5 lists the constraints that could be introduced to obtain the pattern mentioned above.

Table 5.5: Constraints needed to obtain the most interesting pattern for the dropout indicator of the chatbot "Employment" and quality measure "Minimum Dropout & Global Reference" (Eq. 3.10).

Restriction	**Value**
Maximum pattern length	2
Minimum support	$\leq$0.3
Beam width	1

5.1.5 Average Dropout & Local Reference

The pattern with the highest score obtained from this quality measure (Eq. 3.11) was $< 1,3 >$. As can be seen from Fig. 5.13, this pattern has a support of approximately 92% and a negative deviation from the input of the pattern of approximately 13 p.p.. Node 1 is the initial node of the chat. This node has an average dropout of 0%. This means that no one has left the chat at this node. Node 3 has the highest average dropout of all chat. A user is 25% likely to leave the chat on node 3. The same result was obtained from the second approach. This node corresponds to the third successive information interaction.



Pattern	Relative SUP	local_dropout_interest	local_dropout_criterion
[1, 3]	0.9171141	-0.1253655	0.1149745

Figure 5.13: Pattern with the highest score of the quality measure "Average Dropout & Local Reference" (Eq. 3.11), regarding the chatbot "Employment".

The dropout value of this node suggests that it is a bit tedious to have three successive information nodes. This pattern suggests that the chatbot design team should decrease the number of initial informational interactions. With this change, the most impatient users might not leave the chat early on.

Table 5.6 shows the restrictions that were be applied to the "Employment" chat so that this pattern would be obtained more quickly.

Table 5.6: Constraints needed to obtain the most interesting pattern for the dropout indicator of the chatbot "Employment" and quality measure "Average Dropout & Local Reference" (Eq. 3.9).

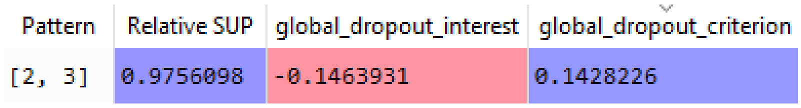
Restriction	**Value**
Maximum pattern length	2
Minimum support	≤0.9
Beam width	1

5.2 Chatbot "Credit"

As previously mentioned, this chat is used by users for credit requests. With this in mind, the main objective of the company that owns this chat is to gather information from its users. This chatbot has only one bifurcation.

The most interesting user behaviours found in this chatbot were obtained by the quality measures "Average Dropout & Global Reference" (Eq. 3.9) and "Average Dropout & Local Reference" (Eq. 3.11) and correspond to a design error.

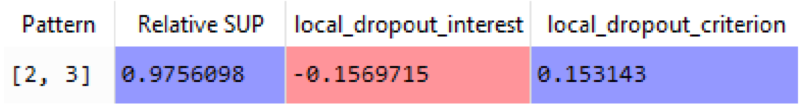
When analysing the results obtained with the minimum constraints for both measures, the most interesting pattern found was $< 2,3 >$. Regarding the measure "Average Dropout & Global Reference", this pattern has a deviation of -0.15 and a support of 0.98. Resulting in an interest of 0.14. From this result, we can deduce that, in this pattern, the probability of the user leaving the chat increases 15 p.p. (Fig. 5.14). This means that there is a significant percentage of users who leave the chat at the beginning.



Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[2, 3]	0.9756098	-0.1463931	0.1428226

Figure 5.14: Pattern with the highest value obtained with the quality measure "Average Dropout & Global Reference" (Eq. 3.9), regarding the chatbot "Credit".

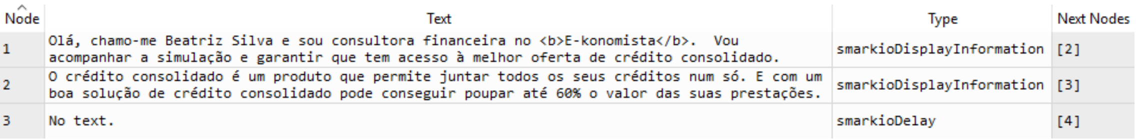
Regarding the local reference (users who start the pattern), the deviation from the average dropout of this pattern is relative to the average dropout of the input node. In this case, the input node is the node with the identifier 2. The results show that the introduction of the node 3 significantly increased the probability of the user leaving the chat. In this way, a user who passes to node 3 is 15 p.p. more likely to leave the chat than a user who is in the interaction 2 (Fig. 5.15).



Pattern	Relative SUP	local_dropout_interest	local_dropout_criterion
[2, 3]	0.9756098	-0.1569715	0.153143

Figure 5.15: Pattern with the highest value obtained with the quality measure "Average Dropout & Local Reference" (Eq. 3.11), regarding the chatbot "Credit".

By analysing the results and the text shown to the user on both nodes (Fig. 5.16), it is possible to realise that node with the identifier 3 is a delay node. The result values suggest that the delay that is occurring may be too high, which causes impatient users to leave the chat in that node.



Node	Text	Type	Next Nodes
1	Olá, chamo-me Beatriz Silva e sou consultora financeira no <b>E-konomista</b>. Vou acompanhar a simulação e garantir que tem acesso à melhor oferta de crédito consolidado.	smarkioDisplayInformation	[2]
2	O crédito consolidado é um produto que permite juntar todos os seus créditos num só. E com um boa solução de crédito consolidado pode conseguir poupar até 60% o valor das suas prestações.	smarkioDisplayInformation	[3]
3	No text.	smarkioDelay	[4]

Figure 5.16: Information about the nodes with the identifiers 2 and 3, regarding the chatbot "Credit".

The chatbot "Credit" is described in more detail in Annex C.

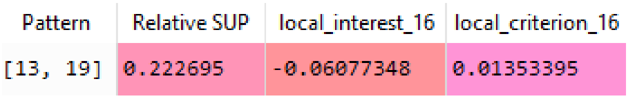
5.3 Chatbot "Christhmas"

The chatbot "Christmas" aims to provide Christmas gift ideas to its users. Most of the interactions in this chat are information gathering interactions. The indicators of interest defined by the company for this chatbot are similar to the previous chatbot. Smarkio intends to understand the interesting behavioural patterns regarding the collection of information of date of birth (indicator 20), email (indicator 22) and mobile phone number (indicator 27). In addition, it also intended to understand the interesting patterns regarding the interaction with the identifier 16. This interaction represents users who have not accepted the proposed terms and conditions.

From the results obtained, several design errors were found in the current chatbot. In addition, a weakness was also discovered in the quality measures designed within this dissertation.

The most interesting patterns found in this chatbot were as follows:

- **Indicator "terms" and Quality Measure "Indicator of Interest & Local Reference"** - Regarding the "terms" indicator and the quality measure "Indicator of Interest & Local Reference" (Eq. 3.8), the most interesting pattern found was the pattern $< 13, 19 >$. This pattern was the pattern found with the highest score, both with the constraint *max_gap=1* and without the constraint. It was possible to verify that this pattern presents a deviation of the input node equal to -0.06 (Fig. 5.17).



Pattern	Relative SUP	local_interest_16	local_criterion_16
[13, 19]	0.222695	-0.06077348	0.01353395

Figure 5.17: Most interesting pattern found for the indicator "terms" with the measure "Indicator of Interest & Local Reference" (Eq. 3.8), regarding the chatbot "Christmas".

This means that, compared to users who passed through node 13, those who went to node 19 are 6 p.p. less likely to not go through node 16. More detailed information about these nodes is shown in Fig. B.3. According to this pattern, in comparison with people who are questioned about accepting terms (node 13), those who accept immediately (node 19) are 6 p.p. less likely to not accept the terms. As previously mentioned, the value of this deviation comes from the following formula:

$$p(16| < 13,19 >) - p(16|13) = 0 - p(16|13) = 0 - 0.06 = -0.06$$

Which can be read as:

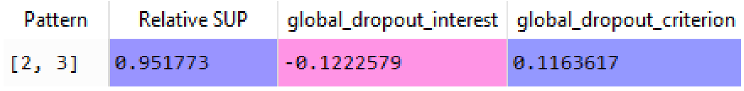
$$p(Do\ not\ accept\ terms) - p(Accept\ the\ terms\ immediately) =$$

$$= 0 - p(Accept\ the\ terms\ immediately) = 0 - 0.06 = -0.06$$

In fact, this deviation should be 0 or impossible, since someone who accepted the terms, can not not have accepted them. Thus, with this indicator, a weakness of the proposed measure of interest was found.

From the $< 13,19 >$ pattern it is not possible to reach node 16, and because of this, the first part of the formula is 0. In this way, the deviation of the pattern is equal to the negative value of the probability of the indicator occurring knowing that the user started the pattern. Thus, the deviation of a pattern that will never reach the indicator of interest will always be $-p(reach\ the\ interest\ indicator$ | beginning of the pattern$)$. If we were faced with a high value of this probability, i.e. 0.75, the pattern $< 13,19 >$ would be considered an extremely interesting pattern relative to the indicator 16. The deviation value would be -0.75. This would lead to erroneous conclusions about the influence of the pattern in the indicator. The pattern $< 13,19 >$, although considered the most interesting, has a very low final score (close to 0). This means that no interesting pattern was found for this indicator.

- **Quality Measure "Average Dropout & Global Reference"** - From Fig. 5.18, it is possible to notice that the most interesting pattern regarding the average dropout of the total population is $< 2,3 >$. The chatbot "Christmas" has an average dropout of 1%. This means that at each node, there is a 1% probability that the user will leave the chat on that node. According to this pattern, a user who is on node 2 or 3 is 12 p.p. more likely to leave the chat than a user on another node in the chat. The same result was obtained with the second approach and with beam width equal to 1.

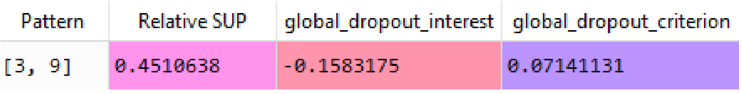


Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[2, 3]	0.951773	-0.1222579	0.1163617

Figure 5.18: Pattern with the highest score of the quality measure "Average Dropout & Global Reference" (Eq. 3.9), regarding the chatbot "Christmas".

Since 44% of the users leave the chat in the first three nodes, this result was expected. From the text shown to the users in interactions 2 and 3 it is possible to realise that the first question made to the user is their gender (Fig. B.3). This result indicates that possibly someone who uses a gift suggestions chat does not feel it necessary to provide their gender. Maybe because of this question the user leaves the chat early on.

- **Quality Measure "Minimum Dropout & Global Reference"** - The pattern found with the highest score for this quality measure (Eq. 3.10) is $< 3,9 >$. Node 3 has an average dropout of approximately 41% and node 9 of 26%. Both nodes have a high average dropout. It was possible to conclude that, in these nodes, a user has at least an additional 16 p.p. probability of leaving the chat (Fig. 5.19).



Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[3, 9]	0.4510638	-0.1583175	0.07141131

Figure 5.19: Pattern with the highest score of the quality measure "Minimum Dropout & Global Reference" (Eq. 3.10), regarding the chatbot "Christmas".

From the text shown to the user in nodes 3 and 9 (Fig. B.3), we can conclude that the users are not willing to provide their gender and do not want to choose one of the proposed categories. The hypothetical basis for the high dropout of node 3 is mentioned in the previous quality measure. The high dropout at node 9 can be due to the fact that the user does not fit what he looks for in the categories available. These categories are "Homem" (men), "Jovem(Rapariga)" (girl), "Jovem (Rapaz)" (boy), "Criança(Menina)" (child - girl) and "Criança(Menino)" (child - boy). There is no information about user data. However, it is possible to verify that there is no category "Mulher" (Woman). This category may be highly sought after, and because this interaction does not have that option, users end up abandoning the chat on this node.

The same result was obtained with the second approach and with beam width equal to 1.

The chatbot "Christmas" is described in more detail in Annex D.

5.4 Summary

For all data sets presented in Section 4.1, several experiments were performed with the combinations of the hyperparameters shown in 4.2. For each dataset and for each combination *Indicator - Measure of Interest* only the patterns with the highest score were analysed.

Most of the patterns with the highest scores for each measure of interest did not prove to be very interesting. The structure of the analyzed chatbots tend to be flat is something that contributes to such. All bifurcations have two possibilities of following nodes. Moreover, although the structure diverges at the bifurcations, it always converges after the user response. This leads to users tending to behave in a similar way over time. It was also possible to verify that, in all the chatbots, a great part of the users leaves the chat in the beginning. This led to the discovery of several flaws in chatbots design.

From the experiments performed, a weakness was discovered in the proposed pattern deviation formulation regarding the quality measures related to the indicators of interest. When the formulation developed is faced with a pattern that does not reach the indicator of interest, the value of the deviation leads to a misinterpretation of the pattern and consequently of the behavior of the user. In this case, the deviation of the pattern is equal to the negative value of the probability of the indicator occurring knowing that the user started the pattern. If we were faced with a high value of this probability, i.e. 0.75, the pattern would be considered an extremely interesting pattern relative to the indicator. This deviation should be 0 or impossible, since someone who follows a path that can not reach the indicator will never reach the indicator.

It could be verified that, in all chatbots, the patterns with the highest score obtained for all indicators and measures were the same for both approaches. The second approach, in all cases, led to the generation of fewer patterns and, consequently, to faster results. Sometimes, in order to obtain the highest scoring pattern in the second approach, it was necessary to increase the beam width size to 10. However, depending on chatbot structure and sessions, this beam width may need to be further increased.

Based on the input parameters required to obtain the best results presented during this chapter, it was possible to obtain a set of general hyperparameters. It should be noted that the adjustment of the parameters for both approaches was a trial error process. The values assigned to each hyperparameter in Table 5.7 are based on the maximum range of input values for the parameter that encompass all the standards mentioned in the current chapter.

Table 5.7: Set of general hyperparameters that allow the discovery of all the patterns mentioned in Chapter 5.

Restriction	**Value**
Maximum gap	None
Maximum pattern length	5
Minimum pattern length	2
Minimum support	≤0.1
Remove flat patterns	Activated/Deactivated
Beam width	10

It is possible to verify that all the hyperparametros have a value or a range of values advisable, except the removal of flat patterns.

The removal or not of flat patterns depends on the chatbot structure. If the chat under study has no bifurcation, removing the flat patterns will lead to no pattern being found. The flat pattern removal option should be enabled when the chat structure is complex enough for this to be justified. Additionally, if the indicator under analysis is a dropout indicator, it is advisable not to enable this option. Most of the interesting patterns found regarding the dropout indicator are flat patterns.

It should also be noted that, in relation to the maximum gap, most of the patterns found were possible to obtain without maximum gap. However, the **max_gap=1** constraint causes the program execution time to decrease significantly. Thus, if the most interesting pattern can be obtained with maximum gap equal to 1, this restriction is advisable to obtain the results in a faster way.

Chapter 6

Conclusions & Future Work

In the current chapter the conclusions reached throughout the development of this dissertation are exposed. The contributions of the work developed, both at the scientific and business levels, are also mentioned. In addition, possible improvements and experiments to be carried out in the future are discussed.

The aim of this dissertation was the development of a way that would allow the discovery of unusual patterns in chatbots users. Two different approaches have been developed to address this problem. The most significant difference between both approaches is the search strategy. While the first approach uses a depth-first search strategy, the second uses a beam search strategy.

From the experiments carried out, it was possible to understand the main differences in the results obtained by both approaches. Due to the limit imposed by the beam width, the second approach always generates a significantly lower number of patterns. This lower number of generated patterns leads to the algorithm being able to more quickly obtain results. The smaller the beam width, the smaller the number of expanded patterns in each level and the total number of patterns generated. One of the problems that was detected in this approach was the sensitivity of the results obtained with respect to the beam width. When the pattern that got the highest score does not have a length equal to 2, this pattern could not be found. In order to find the pattern with the highest score, it is necessary to expand the patterns that originate it. When this does not happen, the pattern with the highest score is not found. In the experiments conducted, sometimes, in order to obtain the highest scoring pattern in the second approach, it was necessary to increase the beam width size to 10. However, depending on chatbot structure and sessions, this beam width may need to be further increased.

Regarding the patterns found, it can be concluded that some patterns did not bring new knowledge. Such patterns emphasise the knowledge already taken and support the veracity of the patterns obtained. It was also possible to verify that, in all the chatbots, a great part of the users leaves the chat in the beginning. This led to the discovery of several flaws in chatbots design. From the patterns discovered, chatbots design and development teams can use the results obtained to correct failures. It was also expected to find patterns that corresponded to best practices. From the data sets received, no best practices were discovered. However, from the preliminary analysis with artificial data present in section 3.5, it was possible to verify that the quality measures can discover patterns that stand out quite positively. With this in mind, it is possible to conclude that the chatbots provided as case studies are not contributing as they should for the company's prosperity.

With the work developed throughout this dissertation it was possible to create several ways to discover unusual patterns in sequential data. The sequential data used in the context of this dissertation were chatbots sessions. However, the algorithms developed can be applied to all sequential data types.

In addition to the two approaches developed, five quality measures have been created taking into account different types of indicators and reference populations. An indicator can be of two types. It can be an indicator of interest or a dropout indicator. An indicator of interest is associated with an interaction that the company has an interest in knowing what kinds of behavioural deviations exist regarding it. The dropout indicator is related to the number of users who leave the chat and corresponds to the average dropout of the reference population. Two types of reference populations were taken into account: the global reference and the local reference. The global reference refers to the total population. In the context of this dissertation, the total population of a chatbot are all sessions of the same. The local reference refers to individuals who have reached the first *itemset* of the pattern. In a context of chatbots users, the local reference of a pattern is all users who initiated the pattern (i.e. reached the first node of the pattern). For each combination *indicator-reference* a different quality measure was designed, except for the combination *dropout indicator-global reference*, for which two different metrics were developed.

In addition to the scientific contributions mentioned, a paper was also submitted to the International Conference on Discovery Science (DS 2018)[1], where the work developed under this dissertation was presented.

Regarding the business contribution, the work developed can also benefit organisations which use chatbots to communicate with their partners, such as customers. Chatbot design and marketing teams can use the results obtained to correct failures and implement the best practices found in other areas or components, both at the system level and at the business level.

6.1 Future Work

Starting with the work carried out within this dissertation, there are many possibilities for future work. Below are some possible extensions of the work developed:

- As discussed in the previous section, the algorithms and approaches developed are adaptable to other sequential data. One possible future experience would be the application of the methods developed to other domains. An example could be Web Mining. Web Mining is the area of Data Mining that aims to extract information from web documents and services [Kosala and Blockeel, 2000]. In this context, the sequences of pages visited by the user could be analysed.

[1] http://www.cyprusconferences.org/ds2018/

- In the context of chatbots, the type of each node could be another information to be taken into account in the future. In case we want to consider only the nodes that are interactions for the analysed data, we could use the node type information in the preprocessing phase of the data provided.

- In this dissertation, the parameter setting phase was a trial error process. In the future, the method for tuning the hyperparameters could be improved.

- From the analysis of the obtained results, it was possible to verify that the value of the beam width in the second approach can influence the most interesting pattern found. The introduction of a low beam width may lead to the pattern with the highest score not being obtained. In this dissertation, the beam width corresponds to the number of patterns with the highest scores that go to the next level. If the meaning of the beam width were changed to the number of best scores that go to the next level, the problem mentioned above would be reduced. In this way, all the patterns that had a score equal to the best *X* scores would be considered, where *X* is the *beam width*. This change would lead to an increase in the number of patterns generated. However, it would be more likely to find the pattern with the highest score, since the number of patterns that would be considered for the next level would be at least equal to those considered with the current approach. With this change it would be expected that the second approach would continue to be faster than the first.

- From the obtained results, it was possible to find patterns that, despite having a deviation considerably low in relation to the reference population, obtained high scores. This was due to the influence of the support of the pattern on its score. This leads to an incorrect interpretation of the results. As a way of controlling the influence of the support of a pattern in its interest score, a possible work for the future would be the introduction of a maximum support. This would minimise the number of patterns that are considered interesting because of their high support.

- During the analysis of the results a weakness regarding the calculation of the deviation of a pattern was discovered. When the formulation developed is faced with a pattern that does not reach the indicator of interest, the value of the deviation induces flaws in the interpretation of the results. In this case, the deviation of the pattern is equal to the negative value of the probability of the indicator occurring knowing the reference population. Calculating the deviation of a pattern from the ratio between the probabilities and not from the difference between them could be a hypothesis for solving the problem. If one of the probabilities were 0, the result of the ratio would be 0 or impossible. This result would be the intended result.

References

Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE, 1995.

Eman Saad AlHagbani and Muhammad Badruddin Khan. Challenges facing the development of the arabic chatbot. In *First International Workshop on Pattern Recognition*, volume 10011, page 100110Y. International Society for Optics and Photonics, 2016.

Tarique Anwar and Muhammad Abulaish. A social graph based text mining framework for chat log investigation. *Digital Investigation*, 11(4):349–362, dec 2014a. ISSN 17422876. doi: 10.1016/j.diin.2014.10.001.

Tarique Anwar and Muhammad Abulaish. A social graph based text mining framework for chat log investigation. *Digital Investigation*, 11(4):349–362, 2014b.

Martin Atzmueller. Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49, 2015.

Martin Atzmueller and Florian Lemmerich. Fast subgroup discovery for continuous target concepts. In *International Symposium on Methodologies for Intelligent Systems*, pages 35–44. Springer, 2009.

Martin Atzmueller and Frank Puppe. Sd-map–a fast algorithm for exhaustive subgroup discovery. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 6–17. Springer, 2006.

Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential pattern mining using a bitmap representation. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 429–435. ACM, 2002.

Thomas Bäck, DB Fogel, and Z Michalewicz. Handbook of evolutionary computation. *Release*, 97(1):B1, 1997.

Francisco Berlanga, María José Del Jesus, Pedro González, Francisco Herrera, and Mikel Mesonero. Multiobjective evolutionary induction of subgroup discovery fuzzy rules: a case study in marketing. In *Industrial Conference on Data Mining*, pages 337–349. Springer, 2006.

Rajesh Boghey and Shailendra Singh. Sequential pattern mining: A survey on approaches. *Proceedings - 2013 International Conference on Communication Systems and Network Technologies, CSNT 2013*, pages 670–674, 2013. doi: 10.1109/CSNT.2013.142.

Mario Boley and Henrik Grosskreutz. Non-redundant subgroup discovery using a closure system. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 179–194. Springer, 2009.

Cristóbal J Carmona, Pedro González, María José del Jesús, and Francisco Herrera. Non-dominated multi-objective evolutionary algorithm based on fuzzy rules extraction for subgroup discovery. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 573–580. Springer, 2009.

Chetna Chand, Amit Thakkar, and Amit Ganatra. Sequential Pattern Mining : Survey and Current Research Challenges. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(1): 185–193, 2012.

CMSWire. Google chatbase ushers in the rise of chatbot analytics, 2018. URL https://www.cmswire.com/digital-experience/google-chatbase-ushers-in-the-rise-of-chatbot-analytics/.

Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Web mining: Information and pattern discovery on the world wide web. In *Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on*, pages 558–567. IEEE, 1997.

Oscar Cord et al. *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*, volume 19. World Scientific, 2001.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

CrowdFlower. *Chatbots Gone Wild*. CrowdFlower, 2017.

Sérgio Luís Neves de Almeida. Extração de conhecimento com data mining na indústria têxtil. *Faculty of Engineering of University of Oporto*, 2012.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2): 182–197, 2002.

María José Del Jesus, Pedro González, Francisco Herrera, and Mikel Mesonero. Evolutionary fuzzy rule induction process for subgroup discovery: a case study in marketing. *IEEE Transactions on Fuzzy Systems*, 15(4):578–592, 2007.

Wouter Duivesteijn and Arno Knobbe. Exploiting false discoveries - Statistical validation of patterns and quality measures in subgroup discovery. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2011. ISBN 9780769544083. doi: 10.1109/ICDM.2011.65.

Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa. *Game analytics*. Springer, 2016.

Philippe Fournier-viger and Jerry Chun-wei Lin. Survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1(1):0–4, 2017.

Philippe Fournier-Viger, Antonio Gomariz, Manuel Campos, and Rincy Thomas. Fast vertical mining of sequential patterns using co-occurrence information. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 40–52. Springer, 2014.

Philippe Fournier-Viger, Jerry Chun-Wei Lin, Rage Uday Kiran, Yun Sing Koh, and Rincy Thomas. A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1 (1):54–77, 2017.

LK Fryer and Rollo Carpenter. Bots as language learning tools. *Language Learning & Technology*, 2006.

Paul A Gagniuc. *Markov Chains: From Theory to Implementation and Experimentation*. John Wiley & Sons, 2017.

Dragan Gamberger and Nada Lavrac. Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research*, 17:501–527, 2002.

Google. Chatbase, 2017. URL https://chatbase.com/welcome.

Henrik Grosskreutz and Stefan Rüping. On subgroup discovery in numerical domains. *Data mining and knowledge discovery*, 19(2):210–226, 2009.

Henrik Grosskreutz, Stefan Rüping, and Stefan Wrobel. Tight optimistic estimates for fast subgroup discovery. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 440–456. Springer, 2008.

Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Freespan: frequent pattern-projected sequential pattern mining. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 355–359. ACM, 2000a.

Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *ACM sigmod record*, volume 29, pages 1–12. ACM, 2000b.

Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data mining and knowledge discovery*, 8(1): 53–87, 2004.

Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.

Francisco Herrera, Cristóbal José Carmona, Pedro González, and María José del Jesus. An overview on subgroup discovery: Foundations and applications. *Knowledge and Information Systems*, 29(3):495–525, 2011. ISSN 02191377. doi: 10.1007/s10115-010-0356-2.

John H Holland. Adaptation in natural and artificial systems. an introductory analysis with application to biology, control, and artificial intelligence. *Ann Arbor, MI: University of Michigan Press*, pages 439–444, 1975.

Tzung-Pei Hong, Chun-Wei Lin, and Yu-Lung Wu. Incrementally fast updated frequent pattern trees. *Expert Systems with Applications*, 34(4):2424–2435, 2008.

Jizhou Huang, Ming Zhou, and Dan Yang. Extracting chatbot knowledge from online discussion forums. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 423–428, 2007.

Jiyou Jia. Csiec (computer simulator in educational communication): an intelligent web-based teaching system for foreign language learning. *arXiv preprint cs/0312030*, 2003.

Harleen Kaur and Siri Krishan Wasan. Empirical Study on Applications of Data Mining Techniques in Healthcare. *Journal of Computer Science*, 2(2):194–200, 2006. ISSN 1549-3636.

Branko Kavšek, Nada Lavrač, and Viktor Jovanoski. Apriori-sd: Adapting association rule learning to subgroup discovery. In *International Symposium on Intelligent Data Analysis*, pages 230–241. Springer, 2003.

Alice Kerly, Phil Hall, and Susan Bull. Bringing chatbots into education: Towards natural language negotiation of open learner models. *Knowledge-Based Systems*, 20(2):177–185, 2007.

Willi Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in knowledge discovery and data mining*, pages 249–271. American Association for Artificial Intelligence, 1996.

Willi Klösgen and Michael May. Spatial subgroup mining integrated in an object-relational spatial database. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 275–286. Springer, 2002.

Raymond Kosala and Hendrik Blockeel. Web mining research: A survey. *ACM Sigkdd Explorations Newsletter*, 2(1):1–15, 2000.

Stewart Kowalski, Katarina Pavlovska, and Mikael Goldstein. Two case studies in using chatbots for security training. In *IFIP World Conference on Information Security Education*, pages 265–272. Springer, 2009.

Nada Lavrač, Bojan Cestnik, Dragan Gamberger, and Peter Flach. Decision support through subgroup discovery: three case studies and the lessons learned. *Machine Learning*, 57(1):115–143, 2004a.

Nada Lavrač, Branko Kavšek, Peter Flach, and Ljupčo Todorovski. Subgroup Discovery with CN2-SD. *The Journal of Machine Learning Research*, 5:153–188, 2004b. ISSN 1532-4435.

Nada Lavrač, Branko Kavšek, Peter Flach, and Ljupčo Todorovski. Subgroup discovery with cn2-sd. *Journal of Machine Learning Research*, 5(Feb):153–188, 2004c.

Dennis Leman, Ad Feelders, and Arno Knobbe. Exceptional model mining. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 1–16. Springer, 2008.

Florian Lemmerich, Martin Becker, Philipp Singer, Denis Helic, Andreas Hotho, and Markus Strohmaier. Mining subgroups with exceptional transition behavior. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 965–974. ACM, 2016.

Michele L McNeal and David Newyear. Introducing chatbots in libraries. *Library technology reports*, 49(8):5–10, 2013.

Katherine Moreland and Klaus Truemper. Discretization of target attributes for subgroup discovery. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 44–52. Springer, 2009.

Marianne Mueller, Rómer Rosales, Harald Steck, Sriram Krishnan, Bharat Rao, and Stefan Kramer. Subgroup discovery for test selection: a novel approach and its application to breast cancer diagnosis. In *International Symposium on Intelligent Data Analysis*, pages 119–130. Springer, 2009.

Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on knowledge and data engineering*, 16(11):1424–1440, 2004.

Gregory Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. *Knowledge discovery in databases*, pages 229–238, 1991.

Nicole M Radziwill and Morgan C Benton. Evaluating quality of chatbots and intelligent conversational agents. *arXiv preprint arXiv:1704.04579*, 2017.

Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, 2004.

Eliana Salvemini, Fabio Fumarola, Donato Malerba, and Jiawei Han. Fast sequence mining based on sparse id-lists. In *International Symposium on Methodologies for Intelligent Systems*, pages 316–325. Springer, 2011.

Bayan Abu Shawar and Eric Atwell. Using dialogue corpora to train a chatbot. In *Proceedings of the Corpus Linguistics 2003 conference*, pages 681–690, 2003.

Bayan Abu Shawar and Eric Atwell. Chatbots: are they really useful? In *LDV Forum*, volume 22, pages 29–49, 2007a.

Bayan Abu Shawar and Eric Atwell. Different measurements metrics to evaluate a chatbot system. In *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, pages 89–96. Association for Computational Linguistics, 2007b.

Smarkio. Smarkio - sales, marketing, integration, optimization, 2018. URL https://smark.io/.

Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *International Conference on Extending Database Technology*, pages 1–17. Springer, 1996.

Milena Tsvetkova, Ruth García-Gavilanes, Luciano Floridi, and Taha Yasseri. Even good bots fight: The case of wikipedia. *PloS one*, 12(2):e0171774, 2017.

Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

Ian H. Witten, Eibe Frank, Mark A. Hall, and Christoper J. Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2017. ISBN 9780128042915. doi: 10.1016/C2009-0-19715-5.

Stefan Wrobel. An algorithm for multi-relational discovery of subgroups. *Principles of Data Mining and Knowledge Discovery*, pages 78–87, 1997.

Mohammed J Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine learning*, 42(1-2):31–60, 2001.

Mohammed Javeed Zaki. Scalable algorithms for association mining. *IEEE transactions on knowledge and data engineering*, 12(3):372–390, 2000.

Filip Železnỳ and Nada Lavrač. Propositionalization-based relational subgroup discovery with rsd. *Machine Learning*, 62(1-2):33–63, 2006.

Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.

REFERENCES

Appendix A

Literature Review and Background

Table A.1: Features of Sequential Pattern Mining algorithms.

Algorithm	**Search Strategy**	**Database Representation**	**Support Calculation**	**Generation of Candidate Sequences**	**Constraints**
AprioriAll	Breadth-first search	Horizontal database	Multiple scans to the database	Apriori-Based (*generate-candidate-and-test* approach)	Minimum support
GSP	Breadth-first search	Horizontal database	Multiple scans to the database	Apriori-Based (*generate-candidate-and-test* approach)	Minimum support; *gap constraints*
SPADE	Depth-first search	Vertical database	*IDList* allows direct calculation of the support of a pattern (only one scan to the database)	Apriori-Based (*generate-candidate-and-test* approach)	Minimum support
bitSPADE	Depth-first search	Vertical database	It uses bit vectors to represent IDLists (Bitmap)	Apriori-Based (*generate-candidate-and-test* approach)	Minimum support
SPAM	Depth-first search	Vertical database	It uses bit vectors to represent IDLists (Bitmap)	Apriori-Based (*generate-candidate-and-test* approach)	Minimum support; minimum and maximum pattern lengths (*length constraints*); *gap constraints*
Fast	Depth-first search	Vertical database	It uses indexed sparse *IDLists* in order to reduce the time and the storage memory required to calculate the support of a pattern	Apriori-Based (*generate-candidate-and-test* approach)	Minimum support
CM-SPADE	Depth-first search with Co-occurrence Map	Vertical database	*IDList* like the SPADE algorithm	Apriori-Based (*generate-candidate-and-test* approach) with a new structure to store co-occurrence information, the Co-occurrence Map (CMAP)	Minimum support
CM-SPAM	Depth-first search with Co-occurrence Map	Vertical database	Bitmap like the SPAM algorithm	Apriori-Based (*generate-candidate-and-test* approach) with a new structure to store co-occurrence information, the Co-occurrence Map (CMAP)	Minimum support; minimum and maximum pattern lengths (*length constraints*); *gap constraints*
FreeSpan	Depth-first search	Database projections	Elaborates several projections of the database	Pattern-Growth-Based (*divide-and-conquer* approach)	Minimum support
PrefixSpan	Depth-first search	*Pseudo-projections*	Elaborates several *pseudo-projections* of the database	Pattern-Growth-Based (*divide-and-conquer* approach)	Minimum support; maximum pattern length

Table A.2: Features of Subgroup Discovery algorithms [Herrera et al., 2011].

Algorithm	**Description Language**	**Type of Target Value**	**Quality Measures**	**Search Strategy**
EXPLORA	Conjunctions of pairs attribute-value. Operators $=$ and $\neq$	Categorical	Redundancy, generality, among others	Exhaustive and heuristic without pruning
MIDOS	Conjunctions of pairs attribute-value. Operators $=, <, >$ and $\neq$	Binary	Novelty or distributional unusualness, among others	Exhaustive and minimum support pruning
Subgroup-Miner	Conjunctions of pairs attribute-value. Operators $=$	Categorical	Binomial test	Beam search
SD	Conjunctions of pairs attribute-value. Operators $=, <$ and $>$	Categorical	Qg	Beam search
CN2-SD	Conjunctions of pairs attribute-value. Operators $=, <, >$ and $\neq$	Categorical	Unusualness	Beam search
RSD	Conjunctions of first order features. Operators $=, <$ and $>$	Categorical	Unusualness, significance or coverage	Beam search
APRIORI-SD	Conjunctions of pairs attribute-value. Operators $=, <$ and $>$	Categorical	Unusualness	Beam search with minimum support pruning
SD4TS	Conjunctions of pairs attribute-value. Operators $=, <$ and $>$	Categorical	Prediction quality	Beam search with pruning
SD-MAP	Conjunctive languages with internal disjunctions. Operator $=$	Binary	*Piatetsky-Shapiro*, unusualness, binomial test, among others	Exhaustive search with minimum support pruning
SD-MAP*	Conjunctive languages with internal disjunctions. Operator $=$	Continous	*Piatetsky-Shapiro*, unusualness, lift	Exhaustive search with minimum support pruning
DpSubgroup	Conjunctions of pairs attribute-value. Operator $=$	Binary and categorical	*Piatetsky-Shapiro*, split, gini and pearson's X^2, among others	Exhaustive search with tight optimistic estimate pruning
MergeSD	Conjunctions of pairs attribute-value. Operators $=, <, >, \neq$ and intervals	Continuous	*Piatetsky-Shapiro*, among others	Exhaustive search with pruning based on constraints among the quality of subgroups
IMR	Conjunctions of pairs attribute-value. Operator $=$	Categorical	Binomial test	Heuristic search with optimistic estimate pruning
SDIGA	Conjunctive or disjunctive fuzzy rules. Operators $=$	Nominal	Confidence, support, sensitivity, interest, significance or unusualness, among others	Genetic algorithm
MESDIF	Conjunctive or disjunctive fuzzy rules. Operators $=$	Nominal	Confidence, support, sensitivity, significance or unusualness, among others	Multi-objective genetic algorithm
NMEEF-SD	Conjunctive or disjunctive fuzzy and/or crisp rules. Operators $=$	Nominal	Confidence, support, sensitivity, significance or unusualness, among others	Multi-objective genetic algorithm

Appendix B

Data

B.1 Data Structure

Each dataset provided by Smarkio includes three files: a *.txt* file containing the indicators of interest, a *JSON* file with the chat structure and a *.csv* file with the session logs.

The indicators file contains all the nodes, which represent interactions, that contribute to some indicator. This file also contains score values for given indicators associated with the transition through specific nodes. The data format for this file is **Node_ID Value Indicator**, where **Node_ID** is the node id and **Value** is the value that this node contributes to the indicator **Indicator**. An example would be "1 0.5 email" which means that if a given path of a user transitions through node 1, the value of 0.5 will be added to the email indicator of that path. All nodes that do not appear related to an Indicator mean that they do not contribute to that indicator, being therefore 0.

The file containing the chat structure represents the chat rules. This file contains all nodes in a chat. From this file we can also see the structure of all possible paths within a given chat. Each node has a set of next nodes. These next nodes represent the possible choices from that node. It is from each node and its next nodes that it is possible to perceive the structure of the chat. Below, it is possible to view the structure of the *JSON* file and some auxiliary comments. This consists of the chat id ("id"), the chat data type ("type") and the set of all existing nodes in that chat ("steps"). It is possible that within each node there are still more possibilities of attributes.

```
{"id": "Chat ID",
  "type": "Type of data. So far always ’sequence’",
  "steps": [
  {
    "id": "Node ID",
    "name": "Text to show to user",
    "next": "Next node ID ",
    "type": "Node Type", // For example: smarkioJump, prompt,
        smarkioDisplayInformation, smarkioSubmit, among others
    "varname": "Variable name", // For example, 'email'
    "additionalVarnames": [ // Only when entering some variable, such as email
      "lead[email]"
    ]
    "data": {
        "text": "Text to show to the user", // Only when the type is equal to 'smarkioDisplayInformation'
        "nextStep": "Next node ID",
        "validation": [ // Only when twxt input is required
           {
            "type": "Validation type, for example regex",
            "setup": {
              "pattern": "Validation, for example a regex expression",
              "flags": "i",
              "invalid_msg": "Message in case of error"
             }}],
        "valueParser": [ // Only when entering some variable, such as email
        "email",
        "trim"
        ]
        "targetRules": [
            { "step": "Next node ID",
              "condition": "Condition to move to the node above"},
            {"step": "Another possible next node",
             "condition": "Condition to move to the node above" }]}}
    ...
]}
```

From the above file, the node identifiers as well as the next node identifiers are used to create the chat structure. The structure is used for checking the sessions. In case a session does not respect the chat structure, it is discarded. In addition, information about the name, text and node type are also used for a better understanding the obtained results. From the previously presented data, some which were redundant were detected. In this file it was already possible to verify some inconsistencies. For example, when from a node it is possible to go to several other nodes, the ID of the first of these other nodes is repeated in the "next", "nextStep" and the first "step" fields within "targetRules". These found inconsistencies were addressed in the data preparation phase.

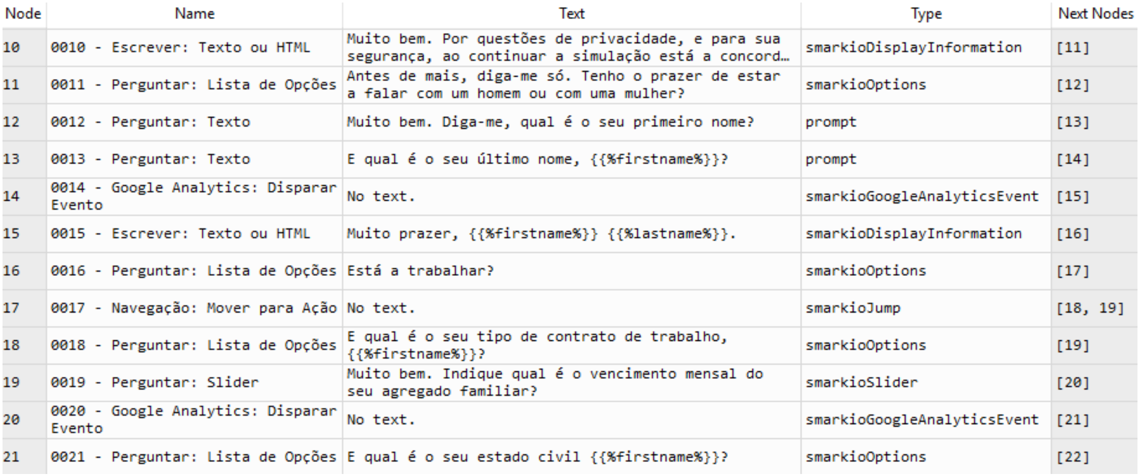
The third file consists of a *CSV* file which contains multiple usage sessions. Each line in this file represents a node from a session. In Table B.1 it is possible to observe the designations of the different columns and the meaning of each.

Table B.1: Format of the data present in the supplied *CSV* files.

Column Name	**Description**
user_id	User ID
server_time	Server time
user_time	Time on user's computer
action_id	Node ID
chat_version	Chat version
category	Chat name
action	Title of operation performed
label	Next node ID
value	The meaning of this attribute is not yet known. This attribute appears empty on all lines in the CSV data file.

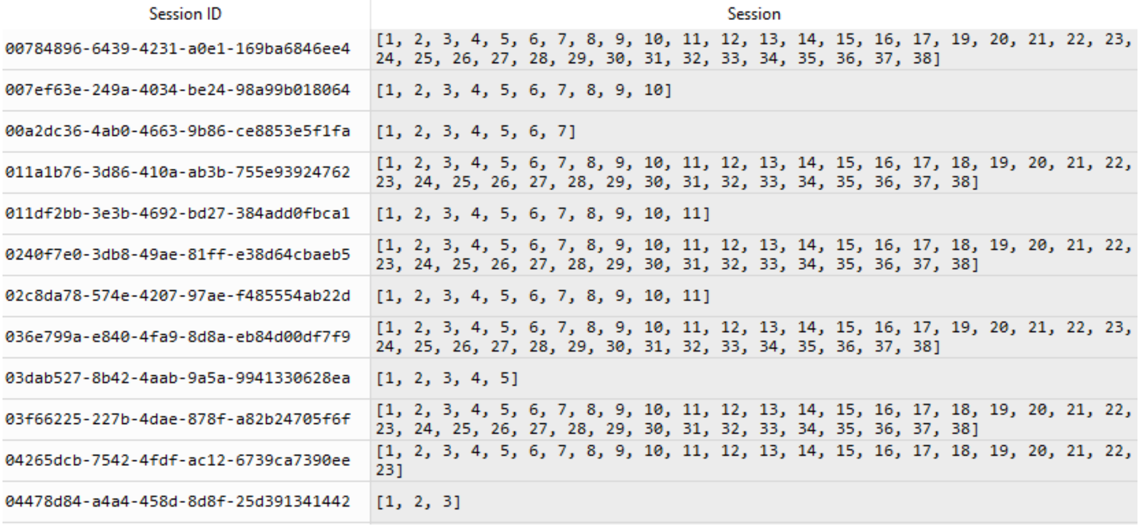
Data about sessions is stored asynchronously. This way, information about all nodes that exist during a session is sent to the database at the same time. For this reason, the order in which the nodes are organised may not be correct. The sorting of the nodes was done through the usage of the **server_time** and **user_time** columns. It has also been detected that each node usually has two rows. The first row is related to the purpose of the node. In case the node is an interaction, it is the question elaborated by the bot. It can also be a validation to the user's response. The second row is relative to the transition to the next node, taking into account the user's response in the previous node(s). However, sometimes it is possible to have only one row, which means that this node is an interaction. In addition, it also means that the user left the chat on that interaction. Since the user left the chat on this node, there is no response associated with it. With this in mind, the session ends at this node. It is also possible that sometimes there are more than two lines, which usually happens when the user refreshes the page. All these inconsistencies were taken into account in the data preparation phase.

B.2 Information about the Provided Datasets



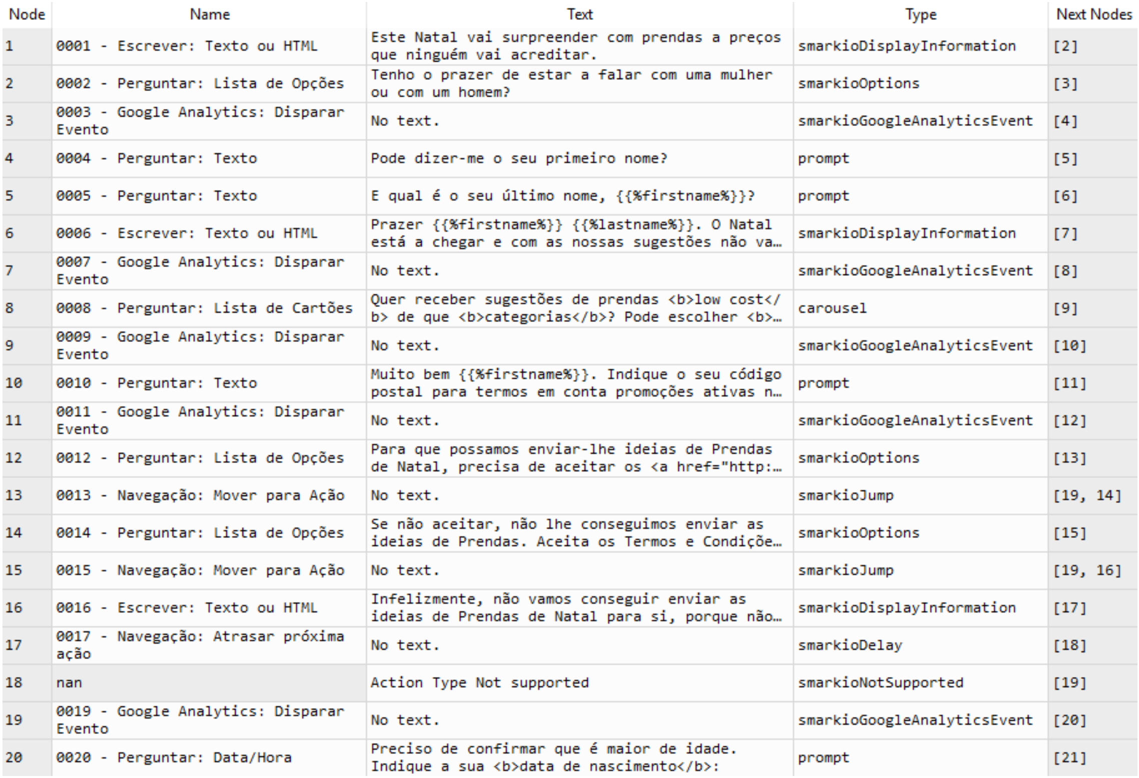
Node	Name	Text	Type	Next Nodes
10	0010 - Escrever: Texto ou HTML	Muito bem. Por questões de privacidade, e para sua segurança, ao continuar a simulação está a concord…	smarkioDisplayInformation	[11]
11	0011 - Perguntar: Lista de Opções	Antes de mais, diga-me só. Tenho o prazer de estar a falar com um homem ou com uma mulher?	smarkioOptions	[12]
12	0012 - Perguntar: Texto	Muito bem. Diga-me, qual é o seu primeiro nome?	prompt	[13]
13	0013 - Perguntar: Texto	E qual é o seu último nome, {{%firstname%}}?	prompt	[14]
14	0014 - Google Analytics: Disparar Evento	No text.	smarkioGoogleAnalyticsEvent	[15]
15	0015 - Escrever: Texto ou HTML	Muito prazer, {{%firstname%}} {{%lastname%}}.	smarkioDisplayInformation	[16]
16	0016 - Perguntar: Lista de Opções	Está a trabalhar?	smarkioOptions	[17]
17	0017 - Navegação: Mover para Ação	No text.	smarkioJump	[18, 19]
18	0018 - Perguntar: Lista de Opções	E qual é o seu tipo de contrato de trabalho, {{%firstname%}}?	smarkioOptions	[19]
19	0019 - Perguntar: Slider	Muito bem. Indique qual é o vencimento mensal do seu agregado familiar?	smarkioSlider	[20]
20	0020 - Google Analytics: Disparar Evento	No text.	smarkioGoogleAnalyticsEvent	[21]
21	0021 - Perguntar: Lista de Opções	E qual é o seu estado civil {{%firstname%}}?	smarkioOptions	[22]

Figure B.1: Information about the nodes of the chat "Credit".



Session ID	Session
00784896-6439-4231-a0e1-169ba6846ee4	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]
007ef63e-249a-4034-be24-98a99b018064	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
00a2dc36-4ab0-4663-9b86-ce8853e5f1fa	[1, 2, 3, 4, 5, 6, 7]
011a1b76-3d86-410a-ab3b-755e93924762	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]
011df2bb-3e3b-4692-bd27-384add0fbca1	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
0240f7e0-3db8-49ae-81ff-e38d64cbaeb5	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]
02c8da78-574e-4207-97ae-f485554ab22d	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
036e799a-e840-4fa9-8d8a-eb84d00df7f9	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]
03dab527-8b42-4aab-9a5a-9941330628ea	[1, 2, 3, 4, 5]
03f66225-227b-4dae-878f-a82b24705f6f	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]
04265dcb-7542-4fdf-ac12-6739ca7390ee	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]
04478d84-a4a4-458d-8d8f-25d391341442	[1, 2, 3]

Figure B.2: Some sessions of the chat "Credit".



Node	Name	Text	Type	Next Nodes
1	0001 - Escrever: Texto ou HTML	Este Natal vai surpreender com prendas a preços que ninguém vai acreditar.	smarkioDisplayInformation	[2]
2	0002 - Perguntar: Lista de Opções	Tenho o prazer de estar a falar com uma mulher ou com um homem?	smarkioOptions	[3]
3	0003 - Google Analytics: Disparar Evento	No text.	smarkioGoogleAnalyticsEvent	[4]
4	0004 - Perguntar: Texto	Pode dizer-me o seu primeiro nome?	prompt	[5]
5	0005 - Perguntar: Texto	E qual é o seu último nome, {{%firstname%}}?	prompt	[6]
6	0006 - Escrever: Texto ou HTML	Prazer {{%firstname%}} {{%lastname%}}. O Natal está a chegar e com as nossas sugestões não va…	smarkioDisplayInformation	[7]
7	0007 - Google Analytics: Disparar Evento	No text.	smarkioGoogleAnalyticsEvent	[8]
8	0008 - Perguntar: Lista de Cartões	Quer receber sugestões de prendas <b>low cost</b> de que <b>categorias</b>? Pode escolher <b>…	carousel	[9]
9	0009 - Google Analytics: Disparar Evento	No text.	smarkioGoogleAnalyticsEvent	[10]
10	0010 - Perguntar: Texto	Muito bem {{%firstname%}}. Indique o seu código postal para termos em conta promoções ativas n…	prompt	[11]
11	0011 - Google Analytics: Disparar Evento	No text.	smarkioGoogleAnalyticsEvent	[12]
12	0012 - Perguntar: Lista de Opções	Para que possamos enviar-lhe ideias de Prendas de Natal, precisa de aceitar os <a href="http:…	smarkioOptions	[13]
13	0013 - Navegação: Mover para Ação	No text.	smarkioJump	[19, 14]
14	0014 - Perguntar: Lista de Opções	Se não aceitar, não lhe conseguimos enviar as ideias de Prendas. Aceita os Termos e Condiçõe…	smarkioOptions	[15]
15	0015 - Navegação: Mover para Ação	No text.	smarkioJump	[19, 16]
16	0016 - Escrever: Texto ou HTML	Infelizmente, não vamos conseguir enviar as ideias de Prendas de Natal para si, porque não…	smarkioDisplayInformation	[17]
17	0017 - Navegação: Atrasar próxima ação	No text.	smarkioDelay	[18]
18	nan	Action Type Not supported	smarkioNotSupported	[19]
19	0019 - Google Analytics: Disparar Evento	No text.	smarkioGoogleAnalyticsEvent	[20]
20	0020 - Perguntar: Data/Hora	Preciso de confirmar que é maior de idade. Indique a sua <b>data de nascimento</b>:	prompt	[21]

Figure B.3: Information about the bifurcations of the chat "Christmas".

Appendix C

Chatbot "Credit" Results

As previously mentioned, this chat is used by users for credit requests. With this in mind, the main objective of the company that owns this chat is to gather information from its users. This chatbot has only one bifurcation. In Fig. B.1 it is possible to understand the nature of the only bifurcation in this chat. The existing bifurcation (node with the identifier 17) refers to the user having or not having a job. If it does, it is redirected to node 18 and later to node 19. Otherwise, it goes directly to node 19.

In order to have a clearer understanding of the indicators of interest, Table C.1 lists the probabilities of each indicator/interaction happen.

Below, the results obtained for the different quality measures for this dataset are presented.

Table C.1: Probability of the indicators of interest of the chat "Credit".

Indicator	**Probability (%)**
terms (11)	47
birthdate (24)	30
email (28)	28
phone (32)	25

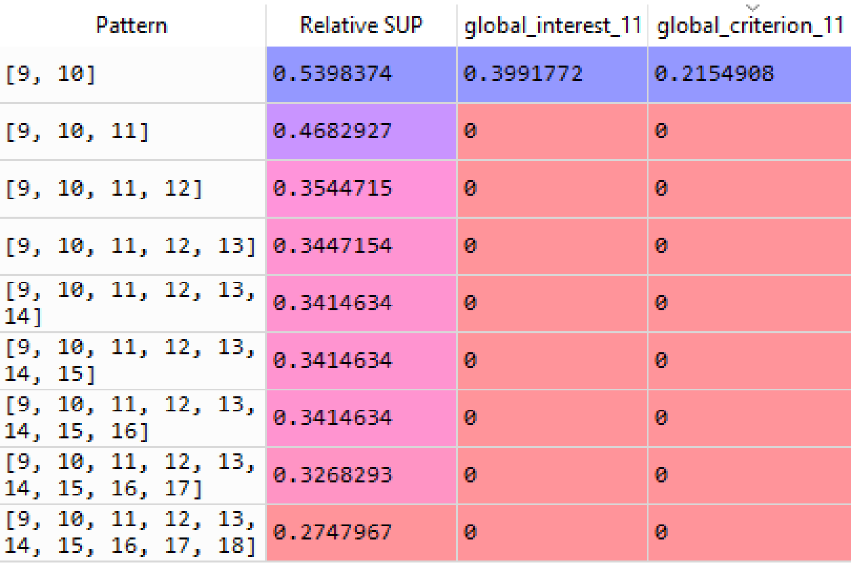
C.1 Indicator of Interest & Global Reference

The first measure of interest that will be discussed for this dataset is the one formulated in 3.7. This measure combines the interest of an indicator with the global reference. For each indicator of interest it was possible to obtain the following results:

- **Indicator "terms" (11)** - This indicator refers to the acceptance by the user of the terms presented. With the minimum restrictions (see section 4.2) the maximum value of this measure was approximately 0.21. There were several patterns that diverged from the total population in the same way. All patterns found with the maximum value have the same support (approximately 0.39) and the same deviation from the total population (approximately 0.22).

It should be noted that all nodes found contain interactions belonging to the interval $\{1,...,10\}$. As we can see from the structure of the current chat, there is no bifurcation up to that point. The flat chat structure up to this node leads to the fact that most of the patterns composed only of nodes that occur before node 11 have the same global deviation. The same goes for the support of the patterns. There is a tendency for all patterns to have the same support.

The second approach led to the same results as the first. As might be expected, the second approach led to the discovery of fewer patterns. The amount of patterns discovered depends on the width of the beam. A larger beam width leads to the discovery of more patterns. As previously mentioned, the beam width corresponds to the number of patterns with the best scores that pass to the next level. In this case, the patterns found with the highest value of the measure of interest were quite numerous. So, in the first level (pattern size equal to 2) only some of the patterns with the best measure score went to the next level. The choice of patterns is based on the order in which the patterns appear. In this way, patterns that go to the next level may not be the ones that lead to better results later. The pattern $< 9, 10 >$ is an example of this statement. From Fig. C.1, it is possible to visualise that the pattern $< 9, 10 >$ goes to the next level, since it has a score equal to the maximum score. However, the pattern with the highest score generated by the extension of this pattern, at the next level, was the pattern $< 9, 10, 11 >$ with a score equal to 0, which is not the best score for patterns with length equal to 3.

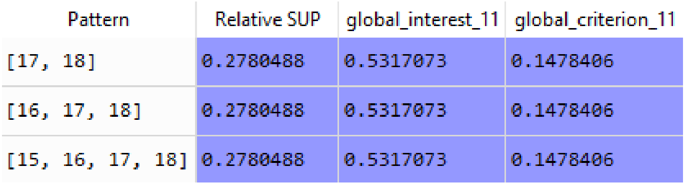


Pattern	Relative SUP	global_interest_11	global_criterion_11
[9, 10]	0.5398374	0.3991772	0.2154908
[9, 10, 11]	0.4682927	0	0
[9, 10, 11, 12]	0.3544715	0	0
[9, 10, 11, 12, 13]	0.3447154	0	0
[9, 10, 11, 12, 13, 14]	0.3414634	0	0
[9, 10, 11, 12, 13, 14, 15]	0.3414634	0	0
[9, 10, 11, 12, 13, 14, 15, 16]	0.3414634	0	0
[9, 10, 11, 12, 13, 14, 15, 16, 17]	0.3268293	0	0
[9, 10, 11, 12, 13, 14, 15, 16, 17, 18]	0.2747967	0	0

Figure C.1: Most interesting pattern found for the indicator "terms" with the quality measure "Indicator of Interest & Global Reference" (Eq. 3.7), regarding the chatbot "Credit". These results were obtained with the second approach and with a beam width of 1.

This chat has a nearly flat structure. It has a unique fork that either goes to node 18 and then 19, or goes directly to node 19. This means that, except for a small deviation, the chat has a linear structure. By activating the option to remove all patterns that do not contain at least one fork, it causes the number of patterns found to decrease significantly.

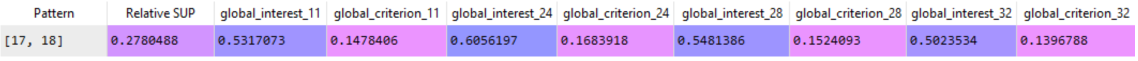
This option has led to more interesting patterns. With this option enabled and maximum gap equal to 1, the most interesting patterns found were $< 15, 16, 17, 18 >$, $< 16, 17, 18 >$ and $< 17, 18 >$. From Fig. C.2, it is possible to visualise that these patterns had a score of approximately 0.15 (support equal to 0.28 and deviation 0.53). These patterns mean that the probability of a user agreeing with the terms increases 53 percentage points (p.p.), knowing that the user indicated that he/she is working. It is possible to understand this analysis only from the pattern $< 17, 18 >$.



Pattern	Relative SUP	global_interest_11	global_criterion_11
[17, 18]	0.2780488	0.5317073	0.1478406
[16, 17, 18]	0.2780488	0.5317073	0.1478406
[15, 16, 17, 18]	0.2780488	0.5317073	0.1478406

Figure C.2: Most interesting pattern found for the indicator "terms" with the quality measure "Indicator of Interest & Global Reference" (Eq. 3.7), regarding the chatbot "Credit". These results were obtained with the second approach, with a beam width of 1 and remove flat patterns activated.

- **Indicators "birthdate", "email" and "phone"** - Regarding the other indicators of interest, it was possible to verify that the most interesting pattern found was the same for all, which was the pattern $< 17, 18 >$. This pattern can be obtained from the same constraints of the "terms" indicator. From Fig. C.3 it is possible to verify that all patterns found were positive patterns. In addition to an employee having an increase of 53 p.p. in relation to acceptance of terms, he/she also has an increase of 60 p.p. in supplying the date of birth, 54 p.p. in providing email, and 50 p.p. in giving the phone number.



Pattern	Relative SUP	global_interest_11	global_criterion_11	global_interest_24	global_criterion_24	global_interest_28	global_criterion_28	global_interest_32	global_criterion_32
[17, 18]	0.2780488	0.5317073	0.1478406	0.6056197	0.1683918	0.5481386	0.1524093	0.5023534	0.1396788

Figure C.3: Most interesting pattern found for the indicators "birthdate", "email" and "phone" with the quality measure "Indicator of Interest & Global Reference" (Eq. 3.7), regarding the chatbot "Credit".

In this chatbot, it is only possible to continue if the terms are accepted. It makes sense that someone who has a job is more interested in reaching the end of the chat, since he knows that he is more likely to receive a credit.

Based on the obtained results, it is possible to conclude that, in this case, the most interesting patterns can be found from the introduction of the restrictions listed in table C.2. These constraints apply to both approaches.

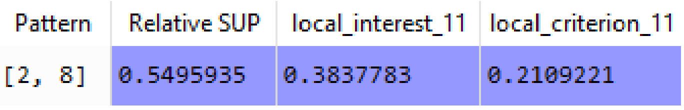
Table C.2: Constraints needed to obtain the most interesting pattern for the indicators of interest of the chatbot "Credit" and quality measure "Indicator of Interest & Global Reference" (Eq. 3.9).

Restriction	**Value**
Maximum gap	1
Maximum pattern length	2
Minimum support	≤0.3
Remove flat patterns	Activated

C.2 Indicator of Interest & Local Reference

As noted earlier, measure 3.8 allows the discovery of interesting patterns relative to local references. This formula calculates the influence of a pattern in its local reference, regarding an indicator of interest. From the obtained results, it is possible to realise the importance that following a certain path regarding a node has in reaching an indicator of interest. Following are the most interesting patterns found for each chat indicator.

- **Indicator "terms" (11)** - The most interesting pattern found was $< 2,8 >$. Node 2 is an initial information node. Node 8 represents the users who provided information about the total amount of their credits. This pattern shows that someone who gives the value of their credits has an increase of 38 p.p. in the probability of accepting the terms (Fig. C.4) compared to all users who reached node 2. Since there is no bifurcation before node 11, the difference of these probabilities corresponds to the dropout between these nodes.

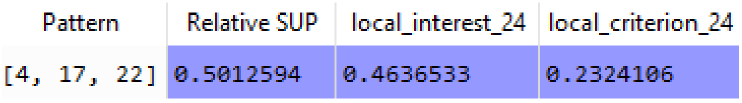


Pattern	Relative SUP	local_interest_11	local_criterion_11
[2, 8]	0.5495935	0.3837783	0.2109221

Figure C.4: Most interesting pattern found for the indicator "terms" with the the quality measure "Indicator of Interest & Local Reference" (Eq. 3.8), regarding the chatbot "Credit".

- **Indicator "birthdate" (24)** - The most interesting pattern relative to this node is $< 4,17,21 >$. Node 4 refers to an initial chat interaction. Until this interaction, no response was requested from the user. Node 17 corresponds to the users who indicated their professional situation. Finally, node 21 represents the users that provided their civil status. With this in mind, of all users who go through node 4, those who provide information about their professional situation and marital status have more 46 p.p. to provide their date of birth as well (Fig. C.5).

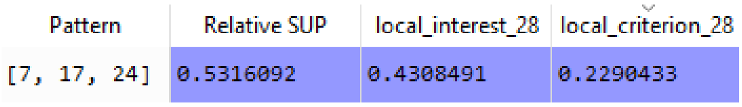
- **Indicator "email" (28)** - The most interesting pattern found was the pattern $< 7,17,24 >$. Node 7 corresponds to the users who provided information about the total amount of their credits. Up to node 17 there is no bifurcation. Node 24 refers to users who provide their date of birth. Therefore, this pattern suggests that, of the users who provide the amount of



Pattern	Relative SUP	local_interest_24	local_criterion_24
[4, 17, 22]	0.5012594	0.4636533	0.2324106

Figure C.5: Most interesting pattern found for the indicator "birthdate" with the the quality measure "Indicator of Interest & Local Reference" (Eq. 3.8), regarding the chatbot "Credit".

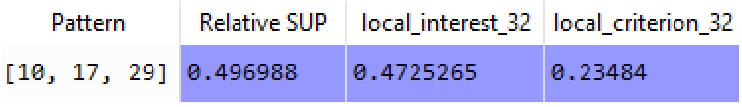
their credits, those who also provide their professional situation and their date of birth are 43 p.p. more likely to provide the email as well (Fig. C.6).



Pattern	Relative SUP	local_interest_28	local_criterion_28
[7, 17, 24]	0.5316092	0.4308491	0.2290433

Figure C.6: Most interesting pattern found for the indicator "email" with the the quality measure "Indicator of Interest & Local Reference" (Eq. 3.8), regarding the chatbot "Credit".

- **Indicator "phone" (32)** - The most interesting pattern was $< 10, 17, 29 >$. Node 10 corresponds to users who answered whether or not they had any credit overdue. Node 17 has already been explained previously. Lastly, node 29 corresponds to the users who provided the email. This pattern indicates that, of the users who answered if they had some credit overdue, those who indicated the professional situation and the email, are 47 p.p. more likely to also provide the mobile phone (Fig. C.7).



Pattern	Relative SUP	local_interest_32	local_criterion_32
[10, 17, 29]	0.496988	0.4725265	0.23484

Figure C.7: Most interesting pattern found for indicator "phone" with the the quality measure "Indicator of Interest & Local Reference" (Eq. 3.8), regarding the chatbot "Credit".

The results obtained with the second approach were the same with a beam width of 10. The length of the most interesting patterns found in this section is equal to 3.

Taking into account that these patterns are not extensions of the patterns generated in level 2 with the highest score, it was necessary to increase the beam width. In this case, the beam width had to be increased to 10, in order to find the patterns with the best scores. Therefore, in order to obtain the patterns with the best score, it is necessary to extend the beam width to 10.

All patterns found with this measure of interest and this dataset demonstrate that the more a user moves forward in the chat, the more likely it is to continue to respond to the bot questions. This is expected, since users who are more interested in obtaining credit tend to continue for longer in the chat.

It was possible to conclude that the most interesting pattern can be found from the introduction of the restrictions listed in table C.3. These constraints apply to both approaches.

Table C.3: Constraints needed to obtain the most interesting pattern for the indicators of interest of the chatbot "Credit" and quality measure "Indicator of Interest & Local Reference" (Eq. 3.9).

Restriction	Value
Maximum pattern length	3
Minimum support	≤0.4
Beam width	10

C.3 Average Dropout & Global Reference

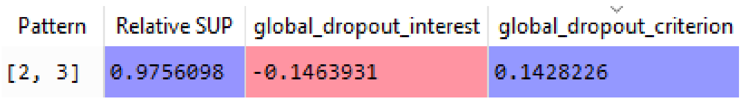
Regarding the measure of interest that combines the dropout indicator and the global reference, there are two possible measures. In this section, the results obtained with the Measure 3.9 will be evaluated. The dropout of a pattern is calculated based on the average dropout of all its nodes. In this measure, the deviation of a pattern is calculated from the difference between the average dropout of the pattern and the average dropout of the chat. In this chatbot, the average chat dropout is 0.035 (3.5%).

When analysing the results obtained from this measure with the minimum constraints (minimum support equal to 0.1 and minimal length of the pattern equal to 2), the most interesting pattern found was $< 2,3 >$. This pattern has a deviation of -0.15 and a support of 0.98. Resulting in an interest of 0.14. From this result, we can deduce that, in this pattern, the probability of the user leaving the chat increases 15 p.p. (Fig. C.10). This means that there is a significant percentage of users who leave the chat at the beginning. The average dropout for node 2 is approximately 0.024 (2.4%) and the average dropout of node 3 is about 0.34 (34%). By analysing the average dropout and the text shown to the user on both nodes (Fig. C.14), it is possible to realise that many users leave the chat on node 3, which is a delay node. Such values suggest that the delay that is occurring may be too high, which causes impatient users to leave that node.

The most interesting pattern found did not match the pattern with the highest deviation of the average dropout of the total population. The pattern with the most deviant average dropout was the pattern $< 10,11 >$. Node 10 has an average dropout of approximately 0.13 and node 11 of about 0.24. From Fig. C.10 it is possible to visualise the differences between the pattern with the greatest deviation and with the greatest interest. Although the pattern $< 2,3 >$ does not have the greatest deviation, it has a very similar deviation.

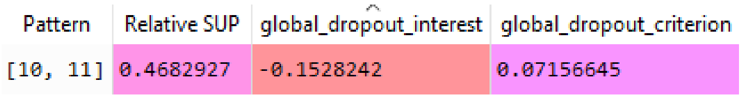
Regarding the second approach, the most interesting pattern was also the pattern $< 2,3 >$. Note that it is possible to obtain this value with beam width equal to 1. Therefore, considering this restriction, the second approach led to the solution more quickly. This speed was due to the fact that in each level only the pattern with the highest score was chosen to be expanded in the next level. If the restriction **max_gap = 1** is considered, both approaches also find the solution faster. Given this, it was possible to conclude that, in this case, the most interesting pattern can be found from the introduction of the restrictions listed in Table C.4. These constraints apply to both approaches, although the second approach is faster.

Chatbot "Credit" Results



Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[2, 3]	0.9756098	-0.1463931	0.1428226

Figure C.8: Pattern with the highest value of interest.



Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[10, 11]	0.4682927	-0.1528242	0.07156645

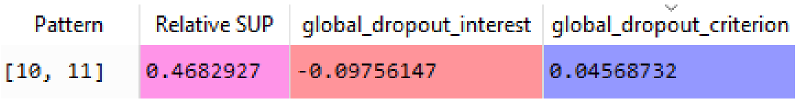
Figure C.9: Pattern with the highest deviation.

Figure C.10: Patterns with the highest deviation and final score of the quality measure "Average Dropout & Global Reference" (Eq. 3.9), regarding the chatbot "Credit".

C.4 Minimum Dropout & Global Reference

In the current section, the second measure related to the dropout indicator and the global reference is analysed. In this measure (Eq. 3.10), the deviation of a pattern is calculated from the minimum difference between the mean deviation of a pattern node and the average deviation of the chat.

From Fig. C.11 we can observe that the most interesting pattern found was $< 10, 11 >$. From the results of the previous measure, it is possible to verify that this pattern was also the pattern that obtained the highest deviation from the total population. Node 10 has an average dropout of 0.13 and node 11 has an average dropout of 0.24. As mentioned earlier, the average chat dropout is 3.5%. In this way, this pattern is the one with the highest minimum difference between the dropout of all nodes in a pattern and the average chat dropout. It was also possible to obtain the same result with both approaches.



Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[10, 11]	0.4682927	-0.09756147	0.04568732

Figure C.11: Patterns with the highest value of deviation and interest relative to quality measure "Minimum Dropout & Global Reference" (Eq. 3.10), regarding the chatbot "Credit".

The text relative to the nodes of this pattern can be seen from Fig. C.12. The transition from node 10 to node 11 corresponds to the acceptance of the terms. The transition from node 11 to node node 12 happens if the user provide his gender. This result shows that, at least, users who are in one of the interactions belonging to this pattern, are 1 p.p. more likely to leave the chat. With this measure of interest the deviation between the pattern and the global reference was significantly lower.

Table C.5 presents the constraints that lead to the most interesting pattern in a faster way. These constraints apply to both approaches, although the second approach is faster. The beam width parameter significantly decreases the number of patterns to be expanded and, in turn, the memory and the time required to obtain the results.

Table C.4: Constraints needed to obtain the most interesting pattern for the dropout indicator of the chatbot "Credit" and quality measures "Average Dropout & Global Reference" (Eq. 3.9) and "Average Dropout & Local Reference" (Eq. 3.11).

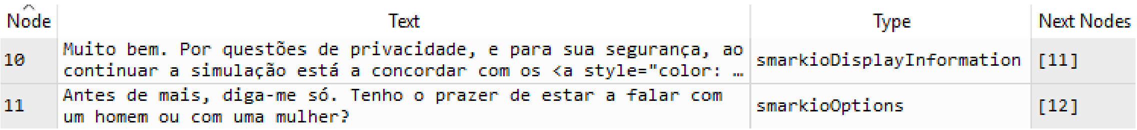
Restriction	**Value**
Maximum gap	1
Maximum pattern length	2
Maximum pattern length	3
Minimum support	$\leq$0.9
Beam width	1

C.5 Average Dropout & Local Reference

Regarding the local reference (users who start the pattern), the pattern with the most interesting dropout was $< 2,3 >$. The deviation from the average dropout of this pattern is relative to the average dropout of the input node. In this case, the input node is the node with the identifier 2. The results show that the introduction of the node 3 significantly increased the probability of the user leaving the chat. In this way, a user who passes to node 3 is 15 p.p. more likely to leave the chat than a user who is in the interaction 2 (Fig. C.13).

From Fig. C.14, it is possible to see that this node corresponds to a delay node. As mentioned in the results of the measure of interest that combines the average dropout with the global reference, this dropout suggest that the delay that is occurring may be too high, which causes impatient users to leave that node.

These constraints apply to both approaches, although the second approach is faster. This pattern can be found from the introduction of the restrictions listed in table C.4.

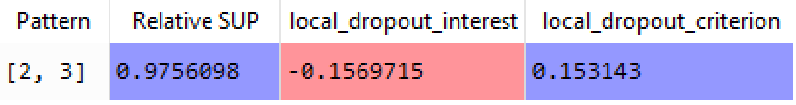


Node	Text	Type	Next Nodes
10	Muito bem. Por questões de privacidade, e para sua segurança, ao continuar a simulação está a concordar com os <a style="color: …	smarkioDisplayInformation	[11]
11	Antes de mais, diga-me só. Tenho o prazer de estar a falar com um homem ou com uma mulher?	smarkioOptions	[12]

Figure C.12: Text shown to the user in interactions 10 and 11, regarding the chat "Credit".

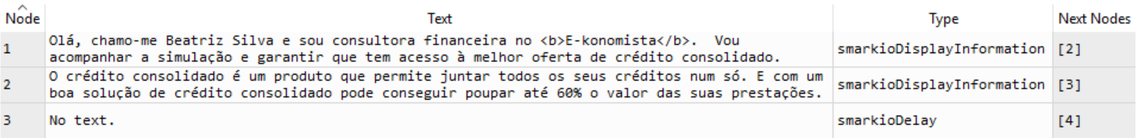
Table C.5: Constraints needed to obtain the most interesting pattern for the dropout indicator of the chatbot "Credit" and quality measure "Minimum Dropout & Global Reference" (Eq. 3.9).

Restriction	**Value**
Maximum gap	1
Maximum pattern length	2
Maximum pattern length	3
Minimum support	$\leq$0.4



Pattern	Relative SUP	local_dropout_interest	local_dropout_criterion
[2, 3]	0.9756098	-0.1569715	0.153143

Figure C.13: Pattern with the highest value obtained with the quality measure "Average Dropout & Local Reference" (Eq. 3.11), regarding the chatbot "Credit".



Node	Text	Type	Next Nodes
1	Olá, chamo-me Beatriz Silva e sou consultora financeira no <b>E-konomista</b>. Vou acompanhar a simulação e garantir que tem acesso à melhor oferta de crédito consolidado.	smarkioDisplayInformation	[2]
2	O crédito consolidado é um produto que permite juntar todos os seus créditos num só. E com um boa solução de crédito consolidado pode conseguir poupar até 60% o valor das suas prestações.	smarkioDisplayInformation	[3]
3	No text.	smarkioDelay	[4]

Figure C.14: Information about the nodes with the identifiers 2 and 3, regarding the chatbot "Credit".

Chatbot "Credit" Results

Appendix D

Chatbot "Christmas" Results

As previously mentioned, chatbot "Christmas" aims to provide Christmas gift ideas to its users. Most of the interactions in this chat are information gathering interactions. The indicators of interest defined by the company for this chatbot are similar to the previous chatbot. Smarkio intends to understand the interesting behavioural patterns regarding the collection of information of date of birth (indicator 20), email (indicator 22) and mobile phone number (indicator 27). In addition, it also intended to understand the interesting patterns regarding the interaction with the identifier 16. This interaction represents users who have not accepted the proposed terms and conditions.

In order to have a clearer understanding of the indicators of interest, Table D.1 lists the probabilities of each indicator/interaction happening.

Table D.1: Probability of the indicators of interest for chat "Christmas"

Indicator	**Probability (%)**
terms (16)	2
birthdate (20)	20
email (22)	14
phone (27)	12

From the Table D.1, it is possible to see that the probability of the indicators are considerably lower than those of the previous chat, especially the probability of the indicator "terms". As mentioned previously, it was possible to verify that 301 sessions only reached node 3 at most. Given that there are 705 sessions in total, only 56% of users pass the initial chat nodes. This makes the probability of the indicators considerably lower.

D.1 Indicator of Interest & Global Reference

In the current section, the most interesting patterns found for each indicator of interest compared to the total population are described.

The pattern with the highest score of this measure for each indicator of interest is shown below.

- **Indicator "terms" (16)** - For this indicator the most interesting pattern found was the pattern $< 6,9,10 >$. Node 6 identifies the users who provided the full name. Node 9 refers to users who have identified the category of products that wish to receive suggestions. Finally, node 10 represent the users who provided their postal code. From Fig. D.5 it is possible to conclude that users who fit the three previous statements are 4 p.p more likely to not accept the terms, compared to the total population. A pattern having a positive deviation from the reference population was defined as a positive consequence for the organisation. However, in this case, the indicator of interest has a negative consequence for the chat, which is the non-acceptance of terms. Therefore, in this case, a positive deviation translates into something negative for the chatbot company.

- **Indicators "birthdate" (20) and "email" (22)** - The pattern $< 13,19 >$ was the most interesting pattern regarding the "birthdate" indicator. The pattern with the highest score for this measure and for the "email" indicator was the pattern $< 13,19,20,21 >$. From node 13 it is possible to go to node 19 directly or indirectly. Comparing the output of the program without the maximum gap constraint and with *max_gap=1*, it was found that the most interesting patterns obtained were the same for both indicators. Although these two patterns are not equal, it is possible to make the same interpretation of both. In this way, this pattern shows that someone who accepts the terms, without the bot having to insist again, has an additional 69 p.p. chance of also providing their date of birth and an additional 59 p.p. chance of giving their email as well (Fig. D.5).

- **Indicator "phone" (27)** - The most interesting pattern found for this indicator was the same as in the previous indicator. Thus, it is possible to conclude that people who accept the terms at the first time and also provide their date of birth have an additional 49 p.p. chance of giving their email, compared with the total population (Fig. D.5).

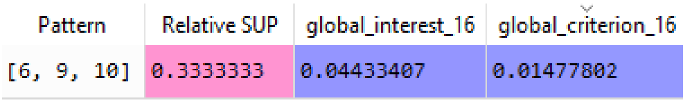
The results obtained by the second approach with beam width equal to 1 were the same as those obtained by the first approach.

Regarding the restrictions, for all indicators the results with *max_gap=1* and without the max gap defined were the same. The same is true for these indicators with the option of removing or not removing flat patterns.

Table D.2 lists the optional constraints to obtain the interesting patterns for all the indicators mentioned above in a faster way.

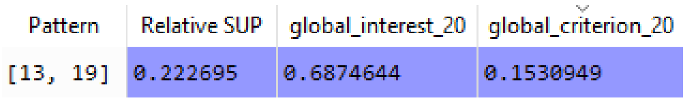
D.2 Indicator of Interest & Local Reference

In this section, it was possible to find the most interesting patterns found in chatbot "Christmas", regarding the indicators of interest and the local reference. Bellow the most interesting patterns found for each chat indicator are described.



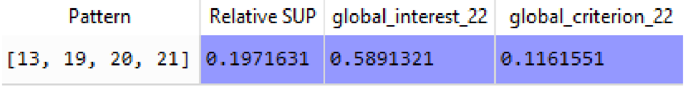
Pattern	Relative SUP	global_interest_16	global_criterion_16
[6, 9, 10]	0.3333333	0.04433407	0.01477802

Figure D.1: Patterns with the maximum score found for the indicator "terms" (16)



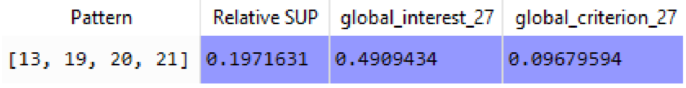
Pattern	Relative SUP	global_interest_20	global_criterion_20
[13, 19]	0.222695	0.6874644	0.1530949

Figure D.2: Patterns with the maximum score found for the indicator "birthdate" (20)



Pattern	Relative SUP	global_interest_22	global_criterion_22
[13, 19, 20, 21]	0.1971631	0.5891321	0.1161551

Figure D.3: Patterns with the maximum score found for the indicator "email" (22)



Pattern	Relative SUP	global_interest_27	global_criterion_27
[13, 19, 20, 21]	0.1971631	0.4909434	0.09679594

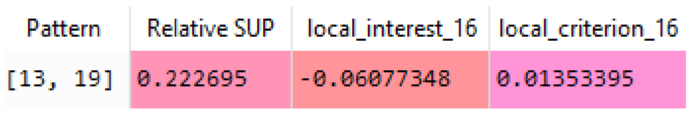
Figure D.4: Patterns with the maximum score found for the indicator "phone" (27)

Figure D.5: Patterns with the maximum score found for the quality measure "Indicator of Interest & Global Reference" (Eq. 3.7), regarding the chatbot "Christmas".

Table D.2: Constraints needed to obtain the most interesting pattern for all the indicators of interest of the chatbot "Christmas" and quality measure "Indicator of Interest & Global Reference" (Eq. 3.9).

Restriction	**Value**
Maximum pattern length	4
Remove flat patterns	Activated
Minimum support	$\leq$0.1
Maximum gap	1
Beam width	1

- **Indicator "terms" (16)** - Regarding the "terms" indicator, the most interesting pattern found was the pattern $< 13, 19 >$. This pattern was the pattern found with the highest score, both with the constraint *max_gap=1* and without the constraint. It was possible to verify that this pattern presents a deviation of the input node equal to -0.06 (Fig. D.6).



Pattern	Relative SUP	local_interest_16	local_criterion_16
[13, 19]	0.222695	-0.06077348	0.01353395

Figure D.6: Most interesting pattern found for the indicator "terms" with the measure "Indicator of Interest & Local Reference" (Eq. 3.8), regarding the chatbot "Christmas".

This means that, compared to users who passed through node 13, those who went to node 19 are 6 p.p. less likely to not go through node 16. Information about these nodes is shown in Fig. B.3. According to this pattern, in comparison with people who are questioned about accepting terms (node 13), those who accept immediately (node 19) are 6 p.p. less likely to not accept the terms. As previously mentioned, the value of this deviation comes from the following formula:

$$p(16| < 13, 19 >) - p(16|13) = 0 - p(16|13) = 0 - 0.06 = -0.06$$

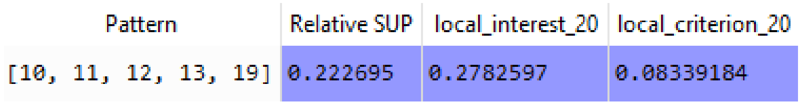
Which can be read as:

$$p(\textit{Do not accept terms}) - p(\textit{Accept the terms immediately}) =$$
$$= 0 - p(\textit{Accept the terms immediately}) = 0 - 0.06 = -0.06$$

In fact, this deviation should be 0 or impossible, since someone who accepted the terms, can not not have accepted them. Thus, with this indicator, a weakness of the proposed measure of interest was found.

From the $< 13, 19 >$ pattern it is not possible to reach node 16, and because of this, the first part of the formula is 0. In this way, the deviation of the pattern is equal to the negative value of the probability of the indicator occurring knowing that the user started the pattern. Thus, the deviation of a pattern that will never reach the indicator of interest will always be $-p(\textit{reach the interest indicator}$ | beginning of the pattern$)$. If we were faced with a high value of this probability, i.e. 0.75, the pattern $< 13, 19 >$ would be considered an extremely interesting pattern relative to the indicator 16. The deviation value would be -0.75. This would lead to erroneous conclusions about the influence of the pattern in the indicator. The pattern $< 13, 19 >$, although considered the most interesting, has a very low final score (close to 0). This means that no interesting pattern was found for this indicator.

- **Indicator "birthdate" (20)** - The pattern with the highest score for the current interest measure and for the birthday indicator is the pattern $< 10, 11, 12, 13, 19 >$ (Fig. D.7). From this pattern it is possible to verify that whoever gives his full name and immediately accepts the proposed terms will have an additional 28 p.p. probability of also providing his date of birth.

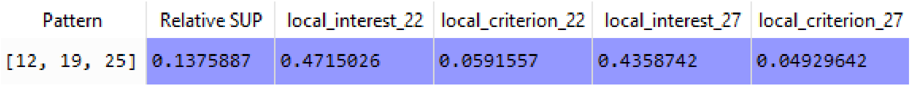


Pattern	Relative SUP	local_interest_20	local_criterion_20
[10, 11, 12, 13, 19]	0.222695	0.2782597	0.08339184

Figure D.7: Most interesting pattern found for the indicator "birthdate" with the measure "Indicator of Interest & Local Reference" (Eq. 3.8), regarding the chatbot "Christmas".

Regarding the first approach, this pattern was obtained with the restriction *max_gap=1* and without the restriction of max gap. Regarding the second approach, this pattern was only found with a beam width of 10. This was due to the fact that there were several patterns found with the highest score with length equal to 2. Thus, in order to allow the extension of all patterns with the highest score found in level 2, the beam width needed to be increased.

- **Indicators "email" (22) and "phone" (27)** - The pattern with the highest interest for the quality measure 3.8 and for the indicators "email" and "phone" was the pattern $< 12, 19, 25 >$ (Fig. D.8). Node 12 refers to users who provided their postal code. Node 19, as previously mentioned, represents the users who accepted the terms. Lastly, node 25 is an information display node that comes after the user provide his/her email.



Pattern	Relative SUP	local_interest_22	local_criterion_22	local_interest_27	local_criterion_27
[12, 19, 25]	0.1375887	0.4715026	0.0591557	0.4358742	0.04929642

Figure D.8: Most interesting pattern found for the indicators "email" and "phone" with the measure "Indicator of Interest & Local Reference" (Eq. 3.8), regarding the chatbot "Christmas".

Regarding the "phone" indicator, this means that, of the users who provided their postal code, those who accepted the terms and who made their email available have a further 44 p.p. to provide their mobile number.

In relation to the indicator "email", the pattern $< 12, 19, 25 >$ showed a positive deviation of 47 p.p.. This means that, from users who provided their postal code, those who also accepted the proposed terms are 47 p.p. more likely to provide also their email. With regard to this indicator, node 25 does not add new information.

The above pattern was obtained by the two approaches for both indicators. There were several patterns with length equal to 2 and with the highest score. Therefore, in order to obtain the pattern with the highest score (length equal to 3) it was necessary to increase the beam width to 10 in both situations. The similarities between these indicators are due to the fact that these interactions are very close to each other. In addition, they do not have any bifurcation between themselves and are interactions closer to the end of the chat, which makes the distributions more similar.

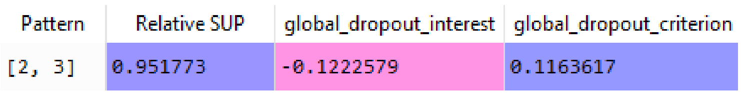
Table D.3 lists the constraints that could be applied to both approaches to obtain the patterns mentioned above.

Table D.3: Constraints needed to obtain the most interesting patterns for all the indicators of interest of the chatbot "Christmas" and quality measure "Indicator of Interest & Local Reference" (Eq. 3.8).

Restriction	**Value**
Maximum pattern length	5
Minimum support	$\leq$0.2
Maximum gap	1
Beam width	10

D.3 Average Dropout & Global Reference

From Fig. D.9, it is possible to notice that the most interesting pattern regarding the average dropout of the total population is $< 2,3 >$. The chatbot "Christmas" has an average dropout of 1%. This means that at each node, there is a 1% probability that the user will leave the chat on that node. According to this pattern, a user who is on node 2 or 3 is 12 p.p. more likely to leave the chat than a user on another node in the chat. The same result was obtained with the second approach and with beam width equal to 1.



Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[2, 3]	0.951773	-0.1222579	0.1163617

Figure D.9: Pattern with the highest score of the quality measure "Average Dropout & Global Reference" (Eq. 3.9), regarding the chatbot "Christmas".

Since 44% of the users leave the chat in the first three nodes, this result was expected. From the text shown to the users in interactions 2 and 3 it is possible to realise that the first question made to the user is their gender (Fig. B.3). This result indicates that possibly someone who uses a gift suggestions chat does not feel it necessary to provide their gender. Maybe because of this question the user leaves the chat early on.

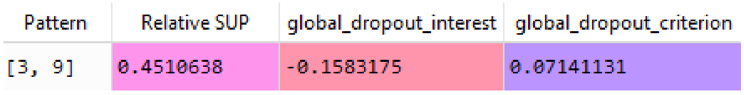
Table D.4 presents the constraints that lead to the most interesting pattern in a faster way. These constraints apply to both approaches.

Table D.4: Constraints needed to obtain the most interesting pattern for the dropout indicator of the chatbot "Christmas" and quality measure "Average Dropout & Global Reference" (Eq. 3.9).

Restriction	**Value**
Maximum pattern length	2
Minimum support	$\leq$0.9
Maximum gap	1
Beam width	1

D.4 Minimum Dropout & Global Reference

The pattern found with the highest score for this quality measure (Eq. 3.10) is $<3,9>$. Node 3 has an average dropout of approximately 41% and node 9 of 26%. Both nodes have a high average dropout. It was possible to conclude that, in these nodes, a user has at least an additional 16 p.p. probability of leaving the chat (Fig. D.10).



Pattern	Relative SUP	global_dropout_interest	global_dropout_criterion
[3, 9]	0.4510638	-0.1583175	0.07141131

Figure D.10: Pattern with the highest score of the quality measure "Minimum Dropout & Global Reference" (Eq. 3.10), regarding the chatbot "Christmas".

From the text shown to the user in nodes 3 and 9 (Fig. B.3), we can conclude that the users are not willing to provide their gender and do not want to choose one of the proposed categories. The hypothetical basis for the high dropout of node 3 is mentioned in section D.3. The high dropout at node 9 can be due to the fact that the user does not fit what he looks for in the categories available. These categories are "Homem" (men), "Jovem(Rapariga)" (girl), "Jovem (Rapaz)" (boy), "Criança(Menina)" (child - girl) and "Criança(Menino)" (child - boy). There is no information about user data. However, it is possible to verify that there is no category "Mulher" (Woman). This category may be highly sought after, and because this interaction does not have that option, users end up abandoning the chat on this node.

The same result was obtained with the second approach and with beam width equal to 1.

Table D.5 lists the constraints that could be introduced to obtain the pattern mentioned above.

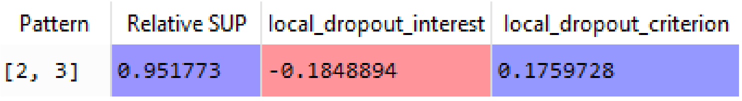
D.5 Average Dropout & Local Reference

For the local dropout measure of the current chatbot, the pattern $<2,3>$ was the pattern with the highest interest. As already mentioned, 44% of the users leave the chat in the first 3 nodes. Node 2 has an average dropout of 4% and node 3 of 41%, approximately. In this way, it is possible to

Table D.5: Constraints needed to obtain the most interesting pattern for the dropout indicator of the chatbot "Christmas" and quality measure "Minimum Dropout & Global Reference" (Eq. 3.10).

Restriction	**Value**
Maximum pattern length	2
Minimum support	$\leq$0.4
Beam width	1

notice that there is a high dropout deviation in this pattern. The calculation of this deviation is based on the average dropout of the pattern input node and on the average dropout of the pattern. This pattern leads to a dropout increase of approximately 18 p.p., relative to the initial node of the pattern (Fig. D.11).



Pattern	Relative SUP	local_dropout_interest	local_dropout_criterion
[2, 3]	0.951773	-0.1848894	0.1759728

Figure D.11: Pattern with the highest score of the quality measure "Average Dropout & Local Reference" (Eq. 3.11), regarding the chatbot "Christmas".

Note that the same result was obtained was obtain with with the restriction *max_gap=1* or without it. The same result was also obtained with the second approach and with beam width equal to 1.

In Table D.6 it is possible to visualise the constraints that can be used to obtain the most interesting pattern for this measure.

Table D.6: Constraints needed to obtain the most interesting pattern for the dropout indicator of the chatbot "Christmas" and quality measure "Average Dropout & Local Reference" (Eq. 3.11).

Restriction	**Value**
Maximum pattern length	2
Minimum support	$\leq$0.9
Maximum gap	1
Beam width	1