Identification of clients affected by a canceled journey on metro network in Porto

By

Miguel Antunes Pereira

Master Thesis in Modeling, Data Analysis
and Decision Support Systems

Supervised by:

Bruno Miguel Delindro Veloso

João Manuel Portela da Gama

Faculdade de Economia

Universidade do Porto

2021

Acknowledgements

I want to make a thank you note to my family, which gave me all the support and conditions to have a successful academic background. I can say for sure that if it were not for them, I would not be here today developing this thesis.

I also want to make an appreciation note to not only to my supervisors, Bruno Veloso and Joao Gama, but also to Carlos Ferreira. They have been tireless in helping me with any subject regarding this thesis and guiding my work towards the right direction to work efficiently on achieving the main goal of this thesis.

At last, I want to thank to my girlfriend and friends for providing me the proper support during this challenging period.

Miguel Pereira

Abstract

The growth and development of urban areas have been constant year by year. The urge to improve the public transport network not only to face this challenge but also to answer the pollution caused by the fast growth of the society opens the possibility for numerous improvements. Nowadays, the emphasis that each city places on its public transportation network can be seen by examining the continuous improvements to the network tracks and equipment in order to provide a more reliable service. Although many fields have been studied in recent years, it is critical to improve the customer experience in order to deliver a quality service to the client. In this digital revolution currently being faced, offering a more intelligent and trustworthy service is the key to staying on top. Therefore, Metro of Porto challenged itself to improve the customer experience by facing the daily challenges that this operation can bring. One challenge is to notice their clients of an eventual breakdown on the metro network as soon as possible. With this information, the client will plan other ways of making his journey and avoid a massive traffic flow on the metro stations.

This thesis intends to investigate the possibility of improving customer experience by evaluating data collected from consumers who have a monthly metro subscription in Porto. In order to achieve this it will be proposed an approach using sequential pattern mining. By examining the frequent item sets and rules generated by a sequential pattern mining algorithm, it will be possible to investigate each individual profile of each network user utilizing this methodology. The obtained output will then contain the information needed to understand if a specific client will be affected by a breakdown of the network.

To demonstrate how this approach may be applied in a real-world setting, a Python script will be developed that will mimic a network collapse. The algorithm's result will then be a boolean variable that shows whether or not a user is affected by the simulated breakdown.

Keywords: Sequential Pattern Mining; Metro Network; Public Transportation Network

Resumo

O crescimento e o desenvolvimento das áreas urbanas têm sido constantes ano após ano. A necessidade de melhorar a rede de transporte público não só para enfrentar este desafio, mas também para responder à poluição causada pelo rápido crescimento da sociedade abre a possibilidade para inúmeras melhorias. Hoje em dia, é possível perceber a importância que cada cidade dá à sua rede de transporte público, analisando a constante melhoria das vias e equipamentos da rede para se tornar um serviço mais confiável. Embora vários campos possam ser explorados nos últimos anos, para prestar um bom serviço ao cliente, é essencial melhorar a experiência do cliente. Nesta revolução digital que se enfrenta, oferecer um serviço mais inteligente e confiável é a chave para se manter no topo. Assim, a rede de metro do Porto desafiou-se a melhorar a experiência do cliente, enfrentando os desafios diários que esta operação pode trazer. Um desafio é perceber quais os clientes que são afetados numa eventual falha da rede do metro. Com essas informações, o cliente vai planear outras formas de fazer sua viagem e evitar um fluxo maciço de tráfego nas estações do metro.

Esta tese pretende explorar a possibilidade de melhoria da experiência do cliente através da análise dos dados obtidos junto dos clientes que possuem assinatura mensal do metro do Porto. É possível aplicar métodos de extração de padrões sequenciais para obter o perfil de cada. Este método irá fazer com que seja obtido para cada cliente um conjunto de padrões frequentes assim como regras entre os padrões obtidos. Com isto é assim possivel determinar se um determinado cliente vai ser afetado por uma eventual avaria da rede do metro.

De modo a demonstrar como esta metodologia pode ser aplicada em tempo real, foi criado um script em Python que irá simular uma avaria da rede de metro e verificar, consoante os dados obtidos, se o cliente é afetado ou não pela avaria.

Keywords: Padrões Sequenciais, Padrões Frequentes

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This section of the thesis will provide a brief overview of the suggested thesis topic, including motivation, problem objectives, and dissertation structure.

1.1 Motivation and Context

The project "Who Is In", proposed by Metro of Porto, is an exciting challenge that will improve the quality of life of the persons who use the network daily. A network outage is unforeseeable, and being able to properly forecast if a client will be affected will allow network users to plan their trips differently and avoid the massive crowds that occur in metro stations when such a problem occurs.

What motivates the most about this challenge is how it can improve the way users of the Porto metro network go about their daily lives. When a metro network goes down, it may be rather distressing for someone who uses it on a daily basis. Having the opportunity to be informed about this in advance would encourage people to try new modes of transportation and, most importantly, better manage their time when using the network. Another thing that motivates me is the opportunity to work with Metro of Porto, a firm that provides a vital service to the inhabitants.

1.2 Problem Formulation and Objectives

The goal of this thesis is to create an algorithm that can anticipate which clients with a monthly subscription to the Porto metro network will be impacted by a network outage. The objective in this thesis is to figure out how to use the available data to design an algorithm that can reliably anticipate which customers would be impacted by a network outage, with proper data pre-processing being important before building the algorithm.

A user with a monthly subscription is likely to use the network frequently to do their everyday tasks. It is feasible to determine each customer's flows and, as a result, trace a profile for each one by examining the station where each client checks his monthly subscription. Subsequently, analyzing a large array of clients, the validated metro stations paired with the weekday and hour are obtained.

The scope of this thesis will be divided according two different stages:

1. Create a sequence comprising the location where each client validates his monthly subscription, along with the matching time of day and day of the week.

2. Develop and algorithm that can simulate a network outage and check if one specific client is affected by the simulated breakdown.

This thesis tries to offer the required knowledge so that the following questions can be answered clearly in the end:

1. Is it possible to analyze a metro of Porto client's behavior by looking at his history of sequence frequent patterns?

2. Is it possible to analyze a metro of Porto client's behavior by looking at his history of sequence frequent patterns associated with a period of the day?

3. Is it possible to analyze a metro of Porto client's behavior by looking at his history of sequence frequent patterns associated with the day of the week?

The three example offered in this thesis were created in such a way that it could answer the issues posed.

As previously stated, the goal of this thesis is to design an algorithm that can properly anticipate if a given client would be harmed by a potential breakdown. By generating a sequence of items containing the most frequent metro stations where the user validates his monthly subscription with the corresponding time of day, it will be possible to match the user's information with the simulated breakdown, allowing to determine whether the user in question will be affected or not.

1.3 Dissertation Structure

The development of the thesis is going to be divided into two parts. The first one will focus on the literature review for the topics related to the development of the subject and the respective methodology that will be applied to fulfill the proposed challenge successfully. Origin-Destination Matrices and Sequential Pattern Mining are two subjects that will be discussed as part of the literature review. Both of these approaches have the potential to address the problem that this thesis is attempting to tackle. The method that will be employed is sequential pattern mining due to the uniqueness of the problem and the available data. The reason why this method is better suited to the suggested issue will be addressed in the methodology part, which will include all steps for the thesis development process as well as a section outlining the available data to address the problem.

The second section of this thesis will detail all of the processes that will be used to complete this thesis goal. First, an exploratory data analysis of the available data will be carried out. Then, all of the procedures for pre-processing data in order to create a profile for each metro network user will be explained. For this, it will be used a sequential pattern mining algorithm in order to fulfill this objective. Once every user's profile is traced, it will be possible to progress to the final part of the thesis, which will be the development of an algorithm using Python. The ultimate goal will be to create an algorithm that can forecast if a given user will be impacted by a metro network outage.

Chapter 2

Related Work

Origin-Destination Matrices and Sequential Pattern Mining are two methodologies for addressing the proposed problem that will be discussed in this chapter. To acquire a good understanding of these methodologies, it was conducted a literature review concentrating on these themes.

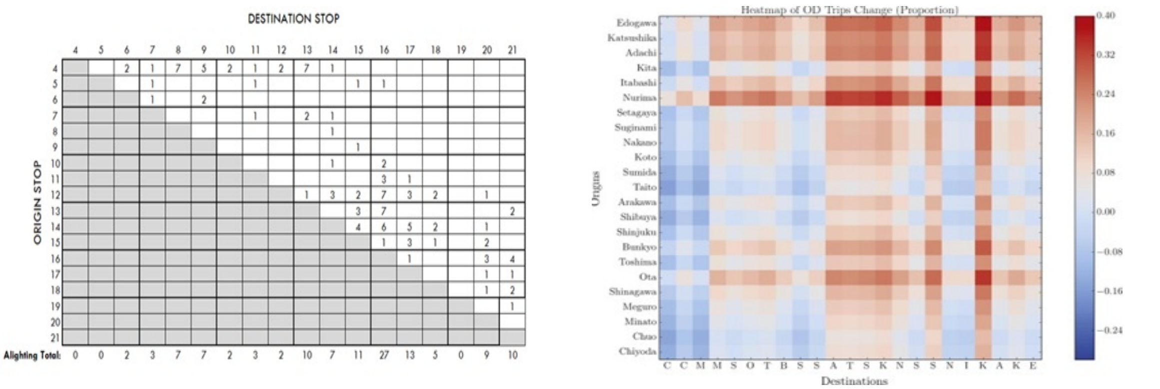
The approach of Origin-Destination Matrices is commonly used for studying mobility patterns on a network, such as the one in this thesis. It creates matrices that combine the information of all network users. Sequential Pattern Mining has grown in popularity in recent years as a result of its versatility in solving problems in various contexts. With the features of the accessible data, it's critical to keep track of a sequence between each user's patterns. When searching for the appropriate strategy to solve the issue provided in this thesis to overcome this problem, sequential pattern mining algorithms became relevant.

2.1 Origin Destination Matrix

This section will present a literature review on the studies performed regarding different Origin-Destination Matrices methods. For this, it will be described both the process used for transportation networks and traffic counts.

Origin-Destination matrices are a crucial part when performing an analysis on a transportation network Abrahamsson (1998) since they are constructed and planned according to the information on this matrix. They represent the number of travelers who commute between two locations points. Figure 2.1 describes two Origin-Destination Matrices: (i) a more generic matrix containing the number of flows between two stops (on the left); and (ii) represents an Origin-Destination Matrix of the flows between neighborhoods in Tokyo in a heat map format (on the right).

According to the article written by Abrahamsson (1998), the methods used to estimate these matrices have different characteristics. For instance, some procedures require an existing Origin-Destination Matrix (obtained from an outdated matrix built using information acquired from a survey, for example). In contrast, other methods do not need an existing Origin-Destination Matrix and create a matrix without a previous matrix.



DESTINATION STOP

ORIGIN STOP

	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
4			2	1	7	5	2	1	2	7	1							
5				1				1				1	1					
6				1		2												
7								1		2	1							
8											1							
9												1						
10											1		2					
11													3	1				
12										1	3	2	7	3	2		1	
13												3	7					2
14												4	6	5	2		1	
15													1	3	1		2	
16														1			3	4
17																	1	1
18																	1	2
19																		1
20																		
21																		
Alighting Total:	0	0	2	3	7	7	2	3	2	10	7	11	27	13	5	0	9	10

(a) Generic Origin Destination Matrix

(b) Origin Destination in Tokyo neighborhoods

Figure 2.1: Examples of Origin Destination Matrices by Ge and Fukuda (2016)

We can split Origin-Destination Matrices into two different groups for different ranges of time according to their purposes. On one side, static Origin-Destination Matrices consider flows as time-independent, averaging the observed flows as suggested by Hu, Yang, Guo, Jensen, and Xiong (2020). We can typically use these matrices for the long-term planning and design of transportation networks. Since this kind of matrices represents a constant flow, it enables the possibility of understanding the mobility patterns of the population. Bera et al. used that information to improve the current network by adding new routes or changing the existing one Bera and Rao (2011). On the other side, Dynamic Origin-Destination Matrices provide beneficial insights from a short-term perspective. In opposition to the static Origin-Destination Matrices, these kinds of matrices consider the time and represent the traffic variations through time as stated by Xiong, Ozbay, Jin, and Feng (2020a). This method can be used, for instance, for traffic control and management of the busiest hours on a transportation network.

2.1.1 Techniques to Estimate Static Origin-Destination Matrices

This section will present several techniques to estimate static Origin-Destination Matrices, from the most traditional to current methods. Estimating these matrices is a process that is usually done for years. Nonetheless, the ways of estimating these matrices have been following the digital evolution over the last years.

2.1.2 Mobility surveys and census information

These are the most traditional ways of obtaining an Origin-Destination Matrix. Depending on the kind of census, it is possible to get information about the commute of the population. The con about this method is that census has a high time interval between them - usually 10 years. However, some countries do it every 5 years.

Mobility Surveys are conducted to the population door by door on residences or workplaces to obtain information about the daily trips of each people. The main difference between this method and the census is the covered area. While the censuses are done on a country level, the

Mobility Surveys are done in restricted areas. Another difference is the censuses' scope, since it collects more demographic information about the population, while the mobility survey's focus is to capture information about the mobility patterns of society.

The principal advantage of these methods is that they allow gathering more information besides the number of trips done. For instance, it can explain why the population commute, which depends on the analysis performed. However, applying this kind of method can consume many human resources, which can be expensive. Also, the periodicity of these surveys is too large. In a fast-growing society like ours, these surveys can become outdated quickly, and it does not count the movements done by tourists, for instance.

2.1.3 Travel Demand-Based Models

Some researchers tried to build an Origin-Destination Matrix using gravity models. Gravity Models are used to predict interactions between two elements and require parameters that need to be tuned as stated by Aerde, Rakha, and Paramahamsan (2003). For instance, it is possible to estimate those parameters on a simple network without using an existing Origin-Destination Matrix and obtain good results as was explained by Högberg (1976). Although, Bera and Rao (2011) noticed that main gravity-based models do not handle external trips accurately - trips which have their origin and/or destination outside the area of interest. According to Willumsen (1978) these models are also not very appropriated for inner-city regions due to the short distance between origin and destination.

2.1.4 Statistical Approaches

A different method for estimating Origin-Destination Matrices is the Statistical Inference approach. The used techniques, suggested by Abrahamsson (1998), are the following: Bayesian Inference (BI), Maximum Likelihood (ML), and Generalized Least Squares (GLS). These approaches assume that probability distributions generate the traffic volumes and the target Origin-Destination Matrix.

Maximum likelihood performs using a target Origin-Destination Matrix. It aims to maximize the likelihood of observing the obtained traffic counts and the initial matrix on the produced matrix. Abrahamsson (1998) says that this method assumes that the observed traffic counts and the target of the Origin-Destination Matrix are independent. According toSpiess (1987), it is possible to obtain a feasible model if the link Counts are consistent. Hazelton (2000) also proposed a method that can estimate a correct Origin-Destination matrix without the need of a previous matrix in contrast with the previously mentioned method. Although, the method proposed by Hazelton (2000) can also incorporate into the existing matrix.

We can use Generalized Least Squares to create Origin-Destination Matrices. Similarly, with the previously mentioned method, it has the assumption that the observed traffic counts and the target Origin-Destination Matrix are independent. Bell (1991) described this method and showed a good outcome. This model is sensitive to incorrect traffic counts, which makes this method less feasible due to the randomness of traffic data.

Bera and Rao (2011) proposed Bayesian Inference approaches. This method uses an existing target matrix as a probability of the new estimated matrix. It uses the observed traffic counts as a different source of information. Then, these sources are combined to produce a new Origin-Destination Matrix. According to Maher (1983), the author proposed an approach using Bayesian Inference and reference that the mains strength of this method is the flexibility of the belief used on the previous Origin-Destination Matrix.

2.1.5 Gradient-Based Techniques

Gradient-Based techniques suggested by Spiess (1990) have as it core an initial Origin-Destination Matrix which is then assigned to the network. The traffic counts observed on this matrix are then used to perform several adjustments on the Origin-Destination Matrix. The resulting matrix produces traffic counts similar to the original one. This method is an iterative process where changes are made according to the gradient function at each iteration, which minimizes the objective function. The authors tested this method on real networks, and the resulting Origin-Destination matrices provided very accurate results.

Despite the good results that were provided by these methods, Doblas and Benitez (2005) tried to control the changes caused by the updating process on the existing matrix, thus limiting the deviation from the target Origin-Destination Matrix.

2.1.6 Entropy Maximization and Information Minimization Methods

Entropy Maximization is a method that seeks the most likely configuration of elements within a constrained situation according to Johnston and Pattie (2000). The objective of this method is to find the flows between the points in a network with the observed traffic counts as the constraints Lam and Lo (1991).

These approaches can use an existing Origin-Destination Matrix, updated according to the observed traffic counts. An Entropy Maximization model was firstly proposed by Willumsen (1978) and presented good results. However, this method requires that inconsistencies on traffic count be removed before applying the algorithm.

J. Van Zuylen (1978) proposed Information Minimization method for estimating Origin-Destination Matrices. Although H. J. Van Zuylen and Willumsen (1980), the authors of the approaches above discussed that these methods were very similar. The only difference between these methods is the unit of observation. However, both of these methods share the same advantages, such as the possibility of using an Origin-Destination matrix obtained from other sources and the excellent fit into small areas - in opposition to the gravity methods previously mentioned. These models also share the same disadvantages like for instance, the inability to deal with incorrect data. The authors did not test this method with real data.

2.1.7 Other Approaches

Other methods were studied and proposed by different authors. Gong (1998) used a Hopfield Neural Network to obtain an Origin-Destination Matrix on a simple network using traffic counts. However, the authors did not test this approach on a real network.

There were more attempts on using neural networks to predict trip distribution, like the one proposed by Mozolin, Thill, and Usery (2000). Still, the model did not present a good outcome.

In recent years, Tesselkin and Khabarov (2017) has tried to estimate Origin-Destination Matrices using Markov Chains. When the Markov property was not violated, the results were entirely satisfactory, i.e. when the transition probability on one node does not depend on the previous node's transition probabilities. However, when this property was violated, the results were inferior, thus affecting the reliability of the obtained matrix. Therefore, the authors mentioned that this approach is suitable for weakly connected networks as between cities. For a strongly interconnected network like in a city, this method is not reliable.

In more recent years, some researchers have been using mobile data to infer Origin-Destination Matrices. Calabrese and Lorenzo (2011) explored the usage of sightings data to estimate these Origin-Destination flows. Y. Yang and Gonz (2016) made a similar approach but used probabilities of transportation mode choices, vehicle occupancy, and Call Detail Records (CDR's) to estimate the Origin-Destination Matrices.

Charisma and Development (2018) fused CDR's with data coming from traffic sensors to estimate Origin-Destination matrices. The traffic counts of the different locations are used to validate the origin-destination predictions. It was necessary to scale a transient Origin-Destination matrix to match the actual traffic flows to determine real Origin-Destinations. As it was explained by Yang, Widhalm, Athavale, and Gonzalez (2016), the heterogeneity in call rates from different locations - for instance, more calls may be generated to and from railway stations compared to and from offices with land telephone lines.

Augusto (2018) used non-supervised learning algorithms to infer users' trips from their CDR's and then, from the travel behaviors, characterize the mobility of the population - for instance, what is the percentage of commuters or how many of them work far from home.

2.2 Dynamic Origin Destination-Matrices

Dynamic Origin Destination-Matrices are matrices that vary over time. According to Bera and Rao (2011) the most popular techniques to estimate these matrices are the following two: State-Space Modelling and Kalman filters.

As proposed by Zhou and Mahmassani (2007), in State-Space Modelling, the first thing to do is to set up an initial state and then specify a transition equation and measurement equation. While the transition equation describes the condition over time, the measurement equation relates the state to the observed indicators, typically the measured traffic counts.

Kalman filtering is a method used to estimate the next state by using a series of observations over time as proposed by Xiong, Ozbay, Jin, and Feng (2020b). This approach handles measurement and state errors. Although, Bera and Rao (2011) mentioned that this method has few disadvantages, like the intensive matrix operations. Chang and Tao (1999) mentioned one implementation of this approach. The author was able to estimate time-varying Origin-Destination Matrices without the need for a previous one.

Perrakis et al. (2012) showed that through generative algorithms like Bayes Nets and Markov random field classification, it is possible to detect activity locations with reasonable accuracy.

Hazelton (2008) proposed a Bayesian Inference method to estimate Origin-Destination Matrices by applying it on a small part of a real network.

Cascetta, Papola, Marzano, Simonelli, and Vitiello (2013) presented an approach that uses Quasi-Dynamic estimations. The estimators are more accurate since the Kalman filtering estimates strongly rely on the initial Origin-Destination Matrix. In the more recent years, this approach was used again by Bauer et al. (2017) to eliminate the need for an initial Origin-Destination Matrix. This method is feasible where path choices have minor importance.

Hai, Akiyama, and Sasaki (1998) describes an approach using neural networks for predicting the flow on points of a network, and the results were pretty exciting. Although, the authors mention that studies with real data should be performed to validate the presented method.

2.3 Sequential Pattern Mining

Discovering unexpected and useful patterns in a database is a fundamental data mining task. In the past years, data mining has been to design algorithms for finding patterns in sequential data. One of the most popular data mining tasks is sequential pattern mining. This task consists of discovering an interesting subsequence, where the interest according to Fournier Viger et al. (2017), its occurrence frequency, length, and profit. Sequential pattern mining has many real applications such as bioinformatics, e-learning, market basket analysis, text analysis, and many more.

2.3.1 The task of Sequential Pattern Mining

Pattern mining consists of discovering interesting, valuable, and unexpected patterns in databases. This field of research appears in the early years of the 1990 decadeAgrawal and Srikanty (1994). The Apriori algorithm was present in the same year, which was a huge breakthrough in finding frequent itemsets. Itemsets are a group of items frequently appearing together in a database, as mentioned by Fournier Viger et al. (2017). Although pattern mining has become very popular due to its applications in several areas, researchers have not explored the sequential ordering of events. For instance, when performing a basket analysis, it does not matter the order of the itemsets. Still, suppose such pattern mining techniques are applied to data with time or sequential ordering information. In that case, there will be information discarded. In many domains, the ordering of events or elements is essential. For instance, to analyze texts, it is often crucial to consider the order of words in sentences as mentioned by Pokou, Fournier-Viger, and Moghrabi (2016). Pramono (2014) references that in network intrusion detection, it is also essential to take into consideration the order of the events.

In the problem faced in this thesis, it is essential to understand each element's order to reach a proper conclusion, as will be mentioned further.

Han, Kamber, and Pei (2011) stated that there are two types of sequential data in data mining: time-series and sequences. A time series is an ordered list of numbers. At the same time, a sequence is an ordered list of nominal values Fournier Viger et al. (2017). Figure **??** it is possible to observe on the left an example of a time series showing amounts of money, while on the right are represented a sequence of letters.

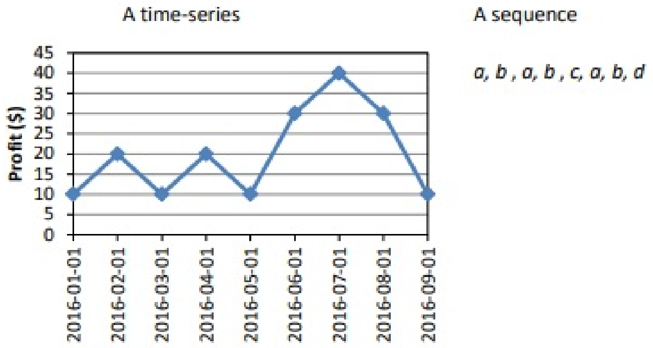


Figure 2.2: Time-Series (left) and a Sequence (right) examples by Fournier Viger et al. (2017)

Srikant and Agrawal (1996a) were the authors who proposed sequential pattern mining as the problem of interesting mining subsequences in a set of sequences. Initially designed to sequences, it also is applied to time series after converting time-series to sequences using discretization techniques Fu (2011). For instance, to transform time-series to sequences, some of the most popular methods are the SAX proposed by J. Lin, Keogh, Wei, and Lonardi (2007), and iSAX Camerra, Palpanas, Shieh, and Keogh (2010).

The task of sequential pattern mining is an enumeration problem. It aims at enumerating all patterns that have support no less than the minimum support threshold. Although, discovering the value of every subsequence can be very time-consuming, and even in some scenarios, impossible. When performing a sequential pattern mining task on a vast database - as it often occurs in real life - the naive approaches to calculating all sequences' support are inefficient due to the high number of subsequences. Therefore there was a need to make sequential pattern mining more efficient to solve the challenging problems proposed by real-life scenarios. With this, several algorithms were proposed so that it is possible to face this challenge.

2.3.2 The beginning of Sequential Pattern Mining Algorithms

GSP was one of the first algorithms proposed by Srikant and Agrawal (1996b) inspired in the Apriori algorithm on the approach for discovering sequential patterns. The input of GSP is a sequence database and a user-specified threshold named minimum support threshold - a value in [0,1] representing a percentage - also known as minsup. The aim of this algorithm is then to find frequent sequential patterns occurring in a sequence database. Spade was another algorithm proposed by M. J. Zaki (2001) which uses a vertical id-list database format and a minimum support threshold as inputs. The principal advantage of this algorithm is to minimize costs by reducing database scans and minimize computational costs by using efficient search schemes. These were two examples of sequential pattern mining algorithms that use a sequential database and a minimum support threshold as inputs where the output is frequent sequential patterns. Nevertheless, besides these two algorithms, many more exist that use the same outputs and have the same goal.

The most popular ones are the PrefixSpan Pei et al. (2004), Spam Ayres, Flannick, Gehrke, and Yiu (2002), Lapin Bhuiyan and Al Hasan (2014), and CM-Spam Fournier-Viger, Gomariz, Campos, and Thomas (2014). The principal difference between these algorithms is not the output since it will always calculate the frequent sequential patterns in the database in analysis. The difference between these algorithms is how they discover sequential patterns. The results show more efficient algorithms for some cases, according to the data in the study.

In general, sequential pattern mining algorithms are typically depth-first search or breadth-first search algorithms. Breadth-first search algorithms, such as GSP, first find sequential patterns containing only one single item. Then they generate sequences of two and proceed until no sequences can be generated. Depth-first search algorithms such as Spade, Spam, Prefix, Lapin, and CM-Spam, explore the search space differently. They start from the sequences containing only single items and then generate larger sequences by recursively performing i-extensions and s-extensions. A sequence Sb is said to be an s-extension of a sequence $S_a = (I_1, I_2, ...I_h)$ with an item x, if $S_b = (I_1, I_2, ...I_h, x)$, i.e. S_a is a prefix of S_b and the item x appears in an itemset occurring after all the itemsets of S_a. A sequence S_c is said to be an i-extension of S_a with an item x, if $S_c = (I_1, I_2, ...I_h \bigcup x)$, i.e. S_a is a prefix of S_c and the item x is appended to the last itemset of S_a, and the item x is the last one in I_h as it is mentioned in Fournier Viger et al. (2017).

In sum, the sequential pattern mining algorithms differ in: (i) if they use a Breadth-first or a Depth-first search; (ii) the type of database representation that they use; (iii) how the algorithms count the support of patterns to determine if they satisfy the minimum support constraint; and (iv) how they explore the determined patterns.

2.3.3 The evolution of Sequence Pattern Mining Algorithms

Although the numerous applications where sequence pattern mining can be applied, there are some fundamental limitations in some cases. One drawback is the vast number of patterns found by the algorithm, which depends on the data and how the minsup threshold is defined. This limitation can be an issue because it can become impossible to analyze such a long number of patterns. Besides this, when the number of patterns found on a database increase, the performance of the algorithm decreases.

To solve the identified issue Fournier-Viger, Gomariz, Campos, and Thomas (2014) proposed a solution that consists of discovering concise representations of sequential patterns instead of all sequential patterns. A concise representation is a subset of all sequential patterns that are meaningful and summarize the whole set of sequential patterns, as stated by Fournier-Viger, Gomariz, Šebek, and Hlosta (2014).

In the literature, we can find several algorithms designed to extract these concise representations without automatically extracting all sequential patterns. These algorithms lead to a more efficient way than the traditional sequential pattern mining algorithms and find a much smaller set of patterns, as stated by Huang, Chang, Tung, and Ho (2006). There are three main concise representations of sequential patterns:

- Closed Sequential Patterns, as mentioned by N. P. Lin, Hao, Chen, Chueh, and Chang (2007) is the set of sequential patterns that are not included in other sequential patterns having the same support. Discovering closed sequential patterns instead of all sequential patterns thus reduces considerably the result set presented to the user, as stated by Gomariz, Campos, Marin, and Goethals (2013). The closed sequential patterns are interesting because they are a lossless representation of all sequential patterns. That is, using the closed sequential patterns, it is possible to recover all the sequential patterns and their support without accessing the database, as mentioned by Fournier-Viger, Gomariz, Campos, and Thomas (2014). Gomariz et al. (2013) also mentioned another reason why the closed sequential patterns are interesting is that they represent the largest subsequences common to sets of sequences. A few examples of algorithms for closed sequential pattern mining are CloSpan, proposed by Yan, Han, and Afshar (2003) and Bide proposed by Wang, Han, and Li (2007).

- Maximal Sequential Patterns Lu and Li (2004) is the set of maximal patterns that is not always larger than closed sequential patterns and all sequential patterns. There are several algorithms proposed for maximal sequential patterns like AprioriAdjust Fournier-Viger, Wu, Gomariz, and Tseng (2014), and MaxSP Fournier-Viger, Wu, and Tseng (2013).

- Generator sequential patterns, as proposed by Pham, Luo, Hong, and Vo (2012) are the set of sequential patterns that have no subsequence having the same support. The set of sequential generators is a subset of all sequential patterns. Still, it can be larger, equal, or smaller than the set of closed patterns as stated by Lo, Khoo, and Li (2008). Nevertheless, it can be argued that generators are preferable to other representations according to the MDL (Minimum Description Length) principle Barron, Rissanen, and Yu (1998). These generators are the smallest subsequences that characterize the group of sequences in a database Lo et al. (2008). Fournier-Viger, Gomariz, Šebek, and Hlosta (2014) also suggested that the combination of generators and white closed patterns generate rules with a minimum antecedent and a maximum consequent. This strategy allows deriving the maximum amount of information based on the minimum amount of information. Some of the most known algorithms are GenMiner Lo et al. (2008) and VGEN Fournier-Viger, Gomariz, Šebek, and Hlosta (2014).

Pei, Han, and Wang (2007) proposed the implementation of constraints in sequential pattern mining. A restriction on this scenario will be an additional field that the user uses as an input to indicate the wanted types of patterns more precisely. The GSP algorithm was the first algorithm that integrated constraints, like the constraints of minimum and maximum amount of time between two consecutive itemsets in sequential patterns and the maximum time duration (known as duration constraint). Zheng, Zhao, Zuo, and Cao (2009) also mentioned the problem of mining negative sequential patterns, which is another limitation of the traditional sequential pattern mining algorithms since their focus is on discovering positively correlated items in a sequence database.

Fournier-Viger, Nkambou, and Nguifo (2008) and Pinto et al. (2001) described an extension of the problem of sequential pattern mining, which is the multi-dimensional sequential pattern mining. These studies considered an extended type of database where each sequence has annotations with dimensions having symbolic values, allowing analysis with more than one dimension. We have two main approaches to perform these analyses: (i) using an item set mining algorithm; and (ii) mining the sequential patterns followed by mining the dimensions.

There are other extensions of sequential pattern mining to make the process more efficient and extract more reach patterns. Since it is one of the hottest topics in the Data Mining field, multiple extensions appear every day to overcome the initial limitations of sequential pattern mining applications. With this literature review, it is possible to apply some of the obtained insights to the development of the thesis.

Chapter 3

Methodology

Initially, the use of Origin-Destination matrices was recommended to find each user's patterns. When it comes to analyzing patterns in networks, this methodology is one of the most common. There has been numerous studies on this topic that have been conducted and applied to society's transportation networks. However, in order to solve the problem addressed in this thesis, it is necessary to examine each client's movement sequence. Origin-Matrices aggregate the information of many users, indicating the flow of different users between different points within the network. This information would not meet the requirements of the Porto metro because it would be impossible to determine if a specific user would be impacted by a network outage.

On this challenge, it is necessary to analyze the data of each user individually. Therefore, it was used an approach using sequential pattern mining to analyze itemsets. This methodology will allow to have a more personalized information about the users of the network allowing for the metro of Porto's criteria to be met.

The development of this thesis was separated into two phases to answer the difficulty provided by the Porto metro. To begin, a sequential pattern mining technique will be used to track the user's profile. The objective is to figure out where the user validates his monthly subscription. It is therefore feasible to trace a sequence of stations that the user visits, thus acquiring the user's travels throughout the metro network.

This challenge has a particularity: a multi-dimensional sequential pattern mining problem. Each user's movement will have a time component that will be the time of the day and the day of the week. This sequential pattern mining problem with a time component was introduced by Fournier-Viger et al. (2008) and Pinto et al. (2001). It will be an essential point of this thesis because it will allow the discovery of the daily patterns of each user.

The goal is to achieve the following data pattern for each of the monthly subscribers of the Metro network:

$$UserA =< \{Pias\}, \{Trindade\} > \tag{3.1}$$

The prior example is an item set that shows that the customer validated his monthly subscription in Pias and Trindade. To select which item sets will be relevant for the analysis, it will be considered the support measure introduced by Agrawal and Srikanty (1994). Therefore, it will need to set a minimum support threshold to find the more relevant item sets. Support represents the percentage of transactions that contain Pias ∪ Trindade, on a daily basis, and can take values from 0 to 1. As support gives the proportion of transactions which has Pias and Trindade and it is often called as frequency constraint and is given by the following formula:

$$Support(Pias \rightarrow Trindade) = P(Pias\ \cup\ Trindade) = \frac{Freq.\,Trindande\ and\ Pias}{Total\ of\ Events} \tag{3.2}$$

Afterward, the item sets can be transformed into rules. These rules will be generated in a format that contains an Antecedent and a Consequent. The items that belong on the Antecedent will be on the left. The items that belong to the Consequent will be on the right, thus giving the information that every time the antecedent happens, the consequent will happen with a certain probability. To evaluate the rules, one of the parameters which will be used will be the confidence parameter. The Confidence is given by the likelihood that a sequential rule $\{Pias\}->\{Trindade\}$ happens among the validations containing the item $\{Pias\}$, under the constraint that the item $\{Pias\}$ happens before the item $\{Trindade\}$. This means that there is a certain probability that every time that the user validates his monthly subscription on the station $\{Pias\}$ it will also validate afterward on the station $\{Trindade\}$. Confidence can also be interpreted as a conditional probability given a previous event, which in this case, is the validation on the station $\{Pias\}$. A high confidence value implies a high likelihood that the validation on the station $\{Trindade\}$ happens in the future. On other words, Confidence is the probability of observing the user validating in Trindade when he validated previously on Pias from 0 to 1, and is given by the following formula:

$$Confidence(Pias \rightarrow Trindade) = P(Pias\,|\,Trindade) = \frac{Support\ (Pias\ \rightarrow\ Trindade)}{Support\ (Pias)} \tag{3.3}$$

Another parameter that is used to evaluate the rules is the Lift. This parameter evaluates the strength of the association. Giving the previous example of the rule $\{Pias\}->\{Trindade\}$, the lift parameter will make it possible to understand how much more likely it is possible to see the item $\{Pias\}$ and $\{Trindade\}$ together compared to what it would be expected if $\{Pias\}$ and $\{Trindade\}$ were completely independent of each other. In other words, a high value of the parameter lift, which is when the lift is higher than 1, implies that the presence of a preexisting item set $\{Pias\}$ increases the probability that the item $\{Trindade\}$ to happen on a future validation. Contrarily, when the lift assumes a low value, which is less than 1, it suggests a negative dependence between the item sets. Lift was first introduced by Brijs, Vanhoof, and Wets (2004) and is the most used measure to evaluate dependencies between the antecedent and consequent and can have values from 0 to infinite, and the following formula gives it:

$$Lift(Pias \rightarrow Trindade) = \frac{P(Pias\ \cup\ Trindade)}{P(Pias)\ *\ P\ (Trindade)} = \frac{Support(Pias\ \rightarrow\ Trindade)}{Support(Pias)\ *\ Support(Trindade)} \tag{3.4}$$

All of this information will be detailed explained in the following sections with the proper examples. To obtain all of this information it will be used the SPADE algorithm proposed by M. Zaki (2001).

3.1 Methodology Steps

The first step will be to transform the information on Metro of Porto's given data into a transaction matrix. A transaction matrix is a specific codification to represent the data to be used as an input for the sequential pattern mining algorithm. Once this data preparation is achieved, the obtained data can then be transformed into frequent item sets and sequence rules, using a sequential pattern mining approach.

Once the data related to the validations of each user is analyzed with a sequential pattern algorithm, it will be possible to develop a script that needs the following three inputs:

- Station where the breakdown happened.
- Direction where the trains traveled.
- Period of the day.

By adding these inputs, it is possible to achieve the following:

- List of metro stations affected by the breakdown.
- List of users affected by the breakdown.

The list of users affected by the breakdown will include the users who were inside the train when the breakdown happens and the users waiting for the train at the following stations.

3.1.1 Description of the available data

The Porto Metro made three different types of data sets available for this study. Over each network client, one has information on every validation detail for the last month. The other comprises information about all of the vehicles on the network's history logs, and the final document contains information about the various metro stations.

The data set containing the information regarding each customer validation will be the one that will be focused. On the table 5.1 are represented the variables that will be used when performing the Sequential Pattern Mining algorithm, achieved from the referred data set.

Variable	Description	Type
Data	Date in the format DD-MM-YYYY when the validation happens	Numerical Variable
diaSemana	Day of the week where the validation happens	Categorical Variable
Time	Time in the format HH:MM:SS when the validation happens	Numerical Variable
Hora	Hour of the day when the validation happens	Categorical Variable
NomeEstacao	Name of the station where the validation happens	Categorical Variable
SN Titulo	ID of the client	Numerical Variable

Table 3.1: Variables Description

This data will allow to trace each user's profile by having a sequence pattern between the different metro stations. On the exploratory data analysis, this data set will be thoroughly discussed.

3.1.2 Software

The software used to develop the pattern sequences, and the mentioned script will be R and Python, respectively. These programming languages are the most popular among data scientists because they provide greater flexibility, allowing programmers to tackle a variety of difficulties with a wide range of options.

To conduct the exploratory data analysis it was used Power BI, which is a business intelligence tool. Then, using Python Pandas and NumPy libraries, all of the data pre-processing and data treatment for each client profile was completed. After that, SPADE on R was used to apply a sequential pattern algorithm to the altered data. The indicated Python script will be built once all of the customers' data has been examined.

Chapter 4

Data Processing

This chapter will describe all the steps that were performed from the original data set provided by Metro of Porto, to the final data that will be used to create the script which can find the users of the network affected by an eventual.

The first step of this chapter will be a data exploratory analysis to the provided data from Metro of Porto, which contains the information where a specific user validates his monthly subscription - described on Table 3.1. The exploratory data analysis will be divided in two parts:

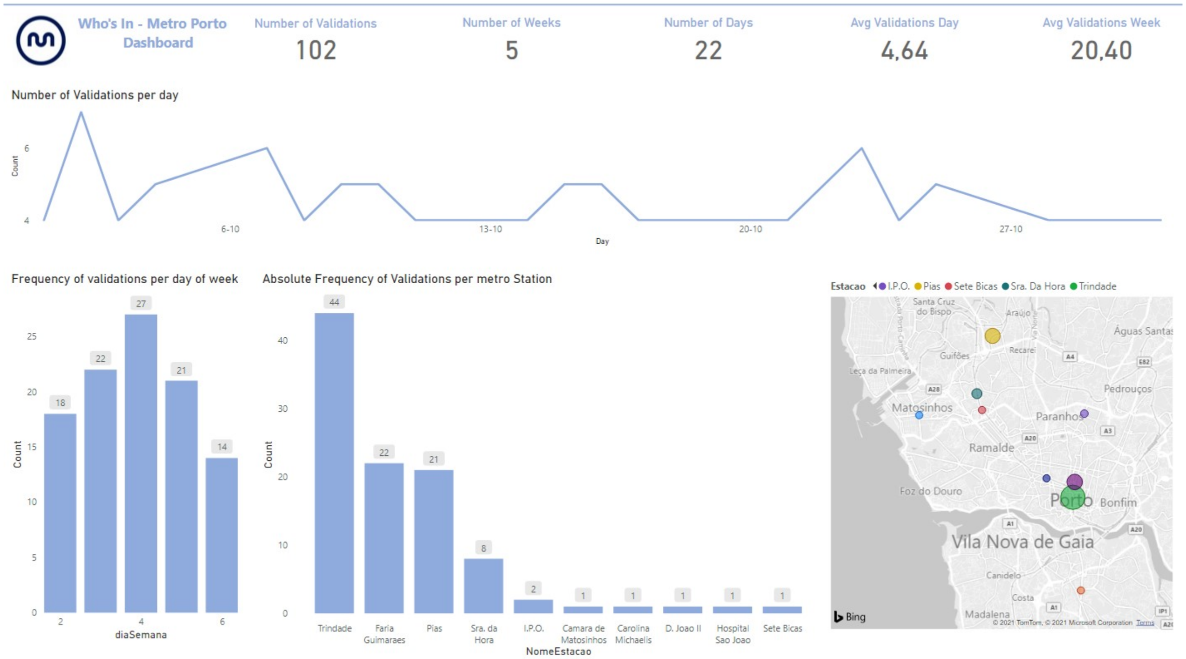
1. The first part will contain a univariate and bivariate analysis of the relevant variables.
2. On the second part it was added two new variables that will help the analysis of the frequent item sets further on the thesis.

Once all insights from the data are collected, the available data will be encoded into a transaction matrix type. A transaction matrix is a way of codifying the data and in order to run the SPADE algorithm, the data needs to be in this specific format, otherwise the algorithm would not work. By using this sequential pattern algorithm will then be possible to find frequent patterns on the data thus tracing the profile of the user in study inside the network.

The final topic of this chapter will be related to the demonstration of the output obtained by the SPADE algorithm. The demonstration of the results will be divided in to three different examples, according to the different transaction matrix that were achieved. It will also be described how the three different transaction matrices were obtained.

4.1 Exploratory Data Analysis

The performed exploratory data analysis will focus on a univariate and bivariate analysis of the relevant variables for the problem in study. For this analysis it was created the following interactive dashboard on (see Figure 4.1) which contains the information regarding the user in study.



Who's In - Metro Porto Dashboard

Number of Validations 102

Number of Weeks 5

Number of Days 22

Avg Validations Day 4,64

Avg Validations Week 20,40

Figure 4.1: Dashboard containing the information about the user in study.

As can be seen, relevant details about the available data can be located at the top of the dashboard, which may be analyzed as following:

- The number of times a user validates his monthly subscription is referred to as the **Number of Validations**.
- The **Number of Weeks** is equivalent to the number of weeks in the data set.
- The **Number of Days** is equivalent to the number of days in the data set.
- The **Avg Validation Day** metric represents the user's average number of validations each day.
- The **Avg Validation Week** metric represents the user's average number of validations each Week.

Some interesting conclusions can already be gleaned from the top measures. For instance, the average number of validations per day is 4,67. This means that the average user travels on metro are four to five times every day.

4.1.1 Univariate Analysis

The following univariate analysis will cover the investigation of the relevant variables. The variable **NomeEstacao** was chosen as the first to be investigated. The information about the metro stations where the user validated his monthly subscription is stored in this variable. The metro stations are depicted by the number of validations in the following Figure 4.2.

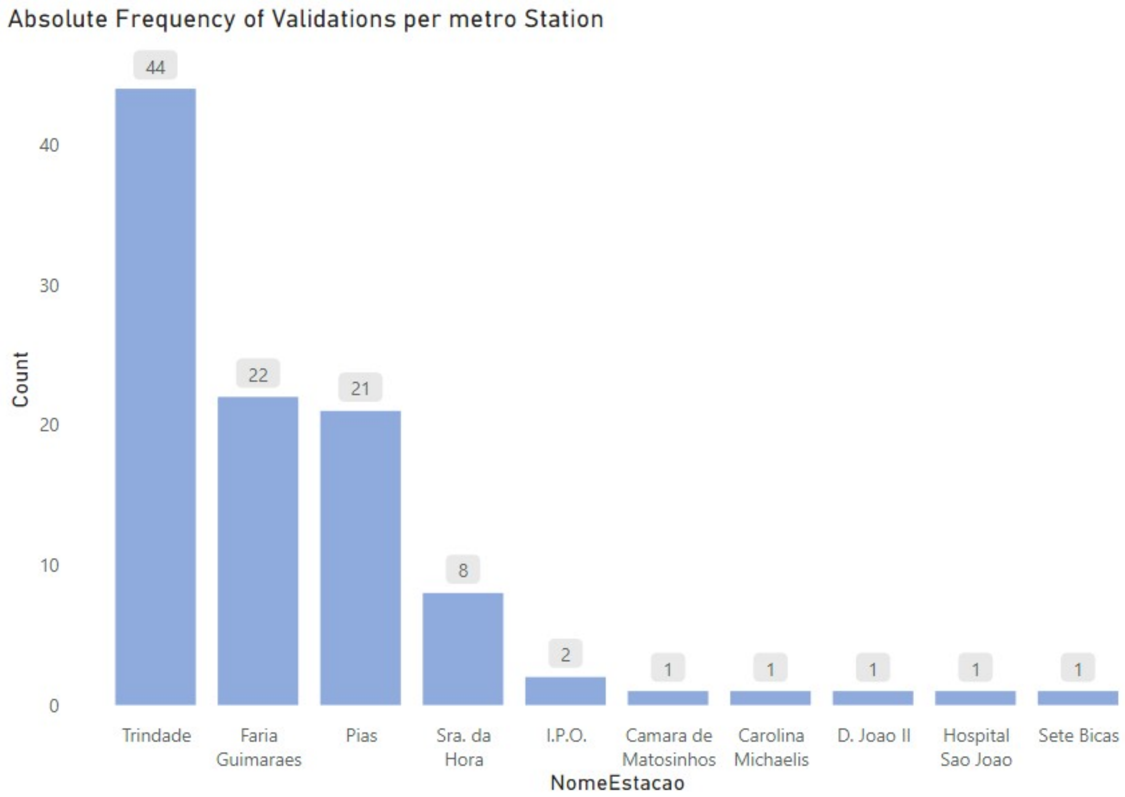


Figure 4.2: Distribution of the variable NomeEstacao represented by a bar chart

Trindade is the station where the user validates the most, followed by Faria Guimaraes and Pias station, as can be seen. Because Trindade is frequently a point of transition between distinct network lines, it is easy to deduce that the user typically commutes between Faria Guimaraes and Pias. In order to have a better knowledge of the user's movements throughout the city of Porto, the distribution of metro stations frequented by the user was indicated on a city map represented on the following Figure 4.3:

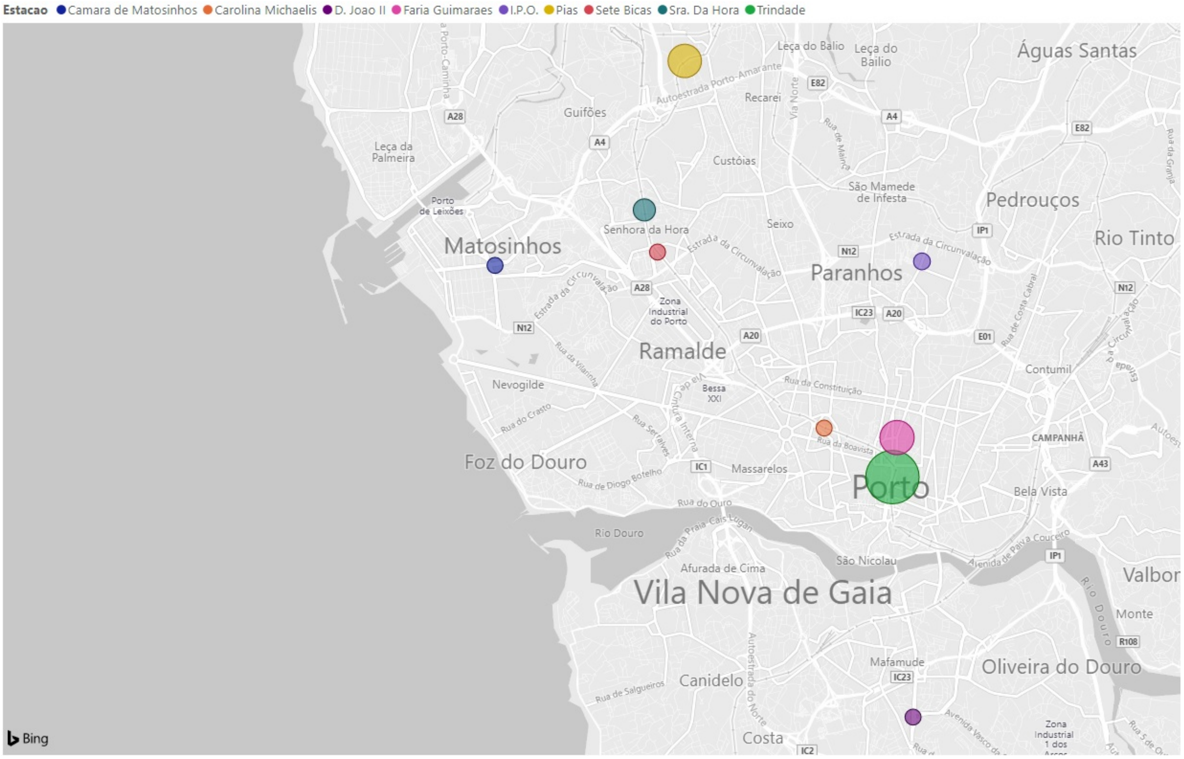


Figure 4.3: Map of Porto city marked by the metro stations where the user validates his subscription

The size of the bubbles represents the number of validations performed on that particular station. The larger the bubble, the more validations there will be at that metro station.

Then, the **diaSemana** variable was chosen as the following variable to be investigated. This is a categorical variable that contains values from 2 to 6 that indicates the day of the week :

- 2 is when the validation happened on a Monday.
- 3 is when the validation happened on a Tuesday.
- 4 is when the validation happened on a Wednesday.
- 5 is when the validation happened on a Thursday.
- 6 is when the validation happened on a Friday.

This variable was drawn in the same way as the previous one, by displaying the distribution of the day of the week according to the number of validations represented on the following Figure 4.4.

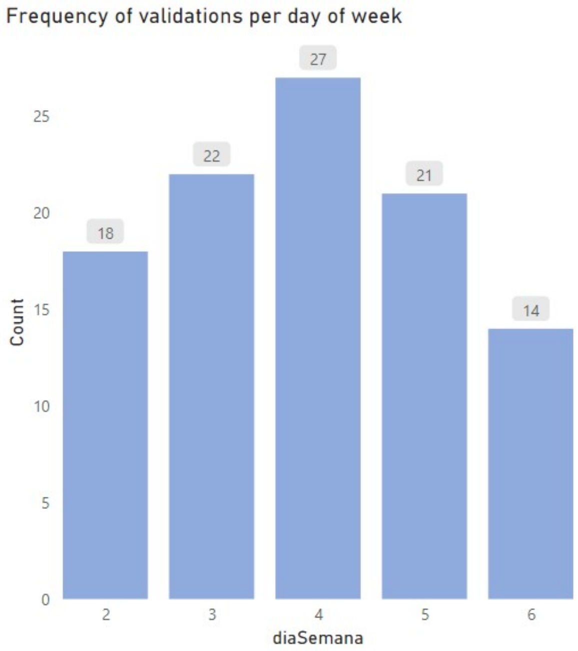


Figure 4.4: Distribution of the variable diaSemana represented by a bar chart

The variable **Data** was the next variable to be looked into. The type of this variable has the Date format since it contains information about the date when the validation occurred. To better understand how the user validated his monthly membership according to the month in question, the following line chart in Figure 4.5 was constructed, which shows the frequency of the user's validations based on the days of the month.

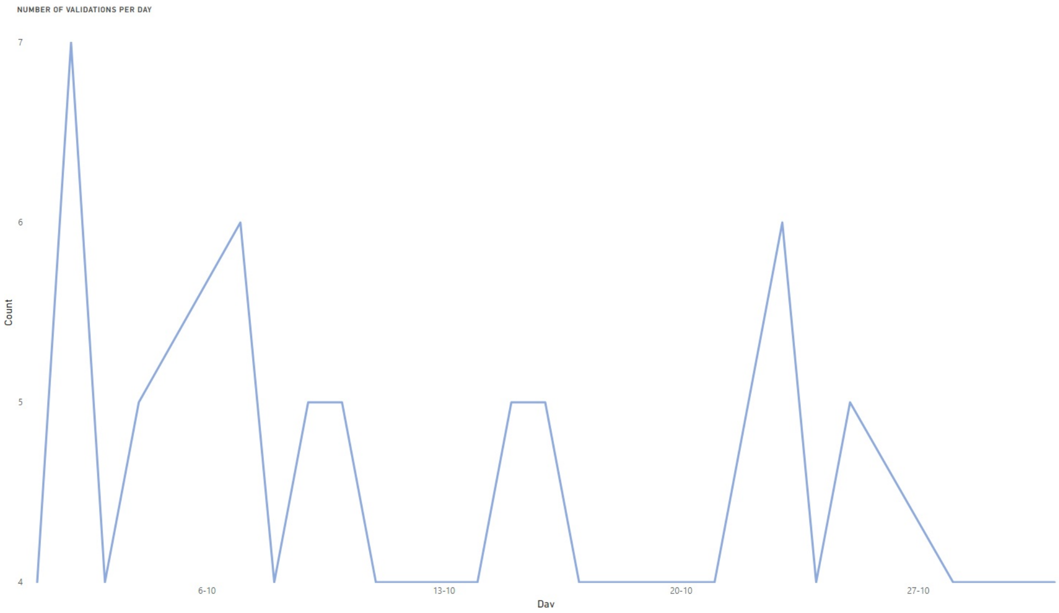


Figure 4.5: Distribution of the variable Data represented by a line chart.

The next variable that came to interest to analyze which is relevant to analyze is the variable **Hora**. This variable contains the information regarding the hour of the day when the validation happens. To make the process of studying this variable easier, a new variable was developed that groups the results from variable A into four categories.

- The validations from 6 a.m. to 11 a.m. were assigned the string ”HP1”, which is related to the traffic hour on the morning period.
- The validations from 11 a.m. to 4 p.m. were assigned the string ”Almoco” which relates to the period that Portugal’s working population uses for their lunch break.
- The validations from 4 p.m. to 8 p.m. were assigned the sting ”HP2”, indicating the period when the working class from Portugal usually leaves their job.
- The validations from 8 p.m. to 6 a.m. were assigned the string ”Noite”. Although the metro schedule usually ends at 1 AM, the metro network can be open 24 hours per day on some special occasions. It is then possible to cover every possible event that can happen inside the network.

With this, a bar plot was built to examine the distribution of the validations among these four categories.

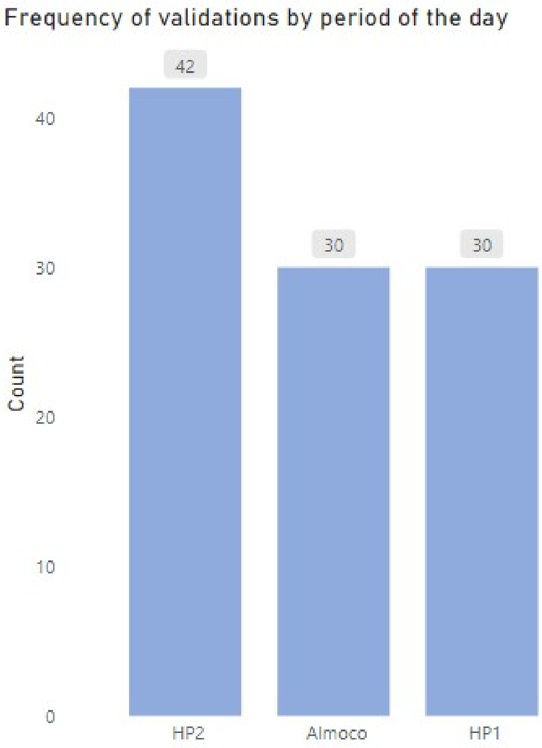


Figure 4.6: The distribution of the variable Hora is represented by the displayed bar plot

When compared to the other schedules, the user validates his subscription more times on the HP1 period, as can be shown.

4.1.2 Bivariate Analysis

The following phase in the exploratory data analysis was to perform a bivariate analysis based on the most important variables for this study. The aim of the thesis is to implement a Sequential Pattern Mining where the sequence of events is essential. Therefore, it was decided to study the time variables existing in the data set. By performing this study it is expected to start to get a understanding of the user behavior inside the network by studying how the validations of the user vary according the day of the week and the period of the day.

As a result, the following contingency table was generated, which represents the percentage of times that both of these variables NomeEstacao and Hora occur on the given data.

NomeEstacao/Hora	Almoco	HP1	HP2	Total
Camara de Matosinhos			0,98 %	0,98 %
Carolina Michaelis		0,98 %		0,98 %
D.Joao II			0,98 %	0,98 %
Faria Guimaraes	2,94 %		18,63 %	21,57 %
Hospital Sao Joao	0,98 %			0,98 %
I.P.O.	1,96 %			1,96 %
Pias	8,82 %	11,76 %		20,59 %
Sete Bicas			0,98 %	0,98 %
Sra. da Hora	1,96 %	4,90 %	0,98 %	7,84 %
Trindade	12,75 %	11,76 %	18,63 %	43,14 %
Total	29,41 %	29,41 %	41,18 %	100,00 %

Table 4.1: Contingency Table for the variables NomeEstacao and Hora.

Because it is feasible to analyze, the user does more of his validations during the HP2 period, which covers the afternoon period. It's also feasible to figure out that the user doesn't authenticate his monthly subscription on Pias during the morning hour. As a result, the idea of the user's house being near Pias gained traction. On the other hand, the user never validates his monthly subscription on Faria Guimaraes during the morning time, implying that the user's place of work is close to this metro station.

Following that, the variables NomeEstacao and diaSemana were examined. As a result, these two variables are depicted in the following contingency table:

NomeEstacao/diaSemana	2	3	4	5	6	Total
Camara de Matosinhos	0,98 %					0,98 %
Carolina Michaelis			0,98 %			0,98 %
D.Joao II			0,98 %			0,98 %
Faria Guimaraes	3,92 %	4,90 %	4,90 %	4,90 %	2,94 %	21,57 %
Hospital Sao Joao			0,98 %			0,98 %
I.P.O.		0,98 %	0,98 %			1,96 %
Pias	3,92 %	4,90 %	3,92 %	4,90 %	2,94 %	20,59 %
Sete Bicas			0,98 %			0,98 %
Sra. da Hora	0,98 %	0,98 %	2,94 %	0,98 %	1,96 %	7,84 %
Trindade	7,84 %	9,80 %	9,80 %	9,80 %	5,88 %	43,14 %
Total	17,65 %	21,57 %	26,47 %	20,59 %	13,73 %	100,00 %

Table 4.2: Contingency Table for the variables NomeEstacao and diaSemana.

Tuesday is the day when the user verifies the most of his monthly subscription, as shown in the table, and Friday is the day when the user validates his monthly subscription the least. The remaining days of the week, have a distribution of validations similar.

One intriguing point is that, with the exception of Wednesday, the validations at metro stations Pias and Faria Guimaraes are the same. This indicates that the user travels between these two metro stations on a regular basis.

Finally, the variables Hora and diaSemana, two time variables found in the data set that will be important to the suggested problem, were investigated. The following contingency table represents the reference variables.

Hora/diaSemana	2	3	4	5	6	Total
Almoco	5,98 %	2,94 %	8,82 %	5,88 %	5,88 %	29,41 %
HP1	1,96 %	8,82 %	5,88 %	4,90 %	7,84 %	29,41 %
HP2	9,80 %	9,80 %	11,76 %	9,80 %		41,18 %
Total	17,65 %	21,57 %	26,47 %	20,59 %	13,73 %	100,00 %

Table 4.3: Contingency Table for the variables Hora and diaSemana

Comparing the amount of validations performed by the user at different times of the day is an interesting analysis to perform. For example, on Mondays, the user confirmed his monthly subscription many fewer times in the morning than in the afternoon. This pattern continues through the rest of the week.

With the exploratory data analysis completed, the thesis may go on to the following phase, which will discuss the methods used to create a transaction matrix, which is a specialized technique of codifying data that is required to perform the sequential pattern mining algorithm.

4.2 Transaction Matrix

The first stage in converting the available data into a transaction matrix was to choose the features that were important to the proposed task. These characteristics are listed in Table 4.4 which contains the first five records of the data set, that will be utilized to generate a Transaction Matrix in the SPADE algorithm's proper structure.

Data	diaSemana	Time	Hora	NomeEstacao
01-10-2019	3	16:39:00	16	Faria Guimaraes
01-10-2019	3	10:11:00	10	Trindade
01-10-2019	3	16:44:00	16	Trindade
01-10-2019	3	09:49:00	19	Pias
01-10-2019	3	18:16:00	18	Faria Guimaraes

Table 4.4: Data set before being transformed in to a transaction matrix.

The SPADE algorithm described by M. Zaki (2001), which is implemented in the R package arulesSequences produced by Buchta and Hahsler (2020), will be utilized for sequential pattern mining. The data input for this R package must be in a specified format. The objective is to convert the given data set into the format shown in Table 4.5, which represents a conventional transaction matrix.

sequenceID	eventID	Size	items
1	10	2	C, D
1	15	3	A, B, C
1	20	3	A, B, F
1	25	4	A, C, D, F
2	15	3	A, B, F
2	20	1	E
3	10	3	A, B, F
4	10	3	D, G, H
4	20	2	B, F
4	25	3	A, G, H

Table 4.5: Example of a transaction Matrix.

The following table contains a description of each variable found on the previous transaction matrix. 4.6.

Variable	Description
sequenceID	Sequence or user identifier
EventID	Event Identifier or transaction information
Size	Number of items for each event
Items	Items per event

Table 4.6: Variables Description

In order to obtain these variables it was used Python to transform the given data format in to a transaction matrix. The following sections will be divided according to the necessary columns obtained progressively with the following order: Items, Sequence ID, eventID, and SIZE.

4.2.1 Items

To build the column ”items” needed for the transaction matrix, the first step was to assign an individual letter to each metro station in the data set. This stage was completed to minimize the item’s complexity, allowing for a better grasp of the rules. The letter assigned to each metro station is shown in the table below. 4.7

Items	Description
A	Trindade
B	Faria Guimarães
C	Pias
D	Sr^{a}. da Hora
E	Hospital São João
F	I.P.O.
G	Carolina Michaelis
H	Sete Bicas
I	D. João II
J	Câmara de Matosinhos

Table 4.7: Variables Description

In Table 4.7 it is possible to see the letter issued to each metro station when the user validates his monthly subscription can be shown. With this, the column ”items” from the transaction matrix may now be created according to the Sequence ID, which is the next step.

4.2.2 Sequence ID

When performing a market basket analysis, the Sequence ID column is usually related to all the different customers available on the data. Although, the analysis that is being performed here has another objective from a market basket analysis. On this issue, it is vital to examine the profile of one user and comprehend his or her everyday movements in conjunction with the metro network. Therefore, the Sequence ID will be related to the day when the user validates his monthly subscription.

The next step will be related to achieving the column Event ID, which will determine how to aggregate the items with the Sequence ID.

4.2.3 Event ID

To begin, when considering the Event ID column, every combination of the Sequence ID and the Event ID on the Transaction Matrix must be unique in order for the SPADE algorithm to locate the frequently item sets and rules. To put it another way, every collection of records from the columns Sequence ID and Event ID should be unique.

After examining the particulars of the presented situation and looking at the available data, the Event ID column will be given by four digits corresponding to the day's time in the following format: HHMM.

This format allows for a unique combination of values from the columns Sequence ID and Event ID, allowing the SPADE algorithm to run correctly.

4.2.4 Size

The Size column is the last column on the transaction matrix. This column shows how many items are associated with an event. A set of unique values from the Sequence ID and Event ID columns constitutes an event.

In order to achieve the correct number for each event, the following logic was applied:

"df_itemset["SIZE"] = df_itemset.itemset.str.count(',')+1"

The previous expression analyzes how many commas exist on the column containing the items and adds one at the end. This ensures that the Size column always contains the exact amount of items.

sequenceID	eventID	Size	items
1	949	1	C
1	1011	1	A
1	1639	1	B
1	1644	1	A
2	810	1	C

Table 4.8: Transaction Matrix.

By achieving this transaction matrix is now possible to run the SPADE algorithm and analyze the given frequent itemsets and the respective rules given by the algorithm.

The following section of this thesis will analyze the outcome of the SPADE algorithm for the given transaction matrix and for three different variants of the transaction matrix, which will be further explained.

4.3 Sequential Pattern Mining

Sequential Pattern Mining is a methodology applied to answer common business questions usually related to market basket analysis. On this problem, this methodology will determine the profile of a user of the metro network. With this, it is able to examine the profile of each network user who has a monthly subscription, determining whether the client would be impacted by a network outage at a given time of day.

Using a Sequential Pattern Mining approach makes it possible to determine if going to a specific metro station will determine a higher likelihood of going to other specific metro stations in the future. For instance, it is possible by using this methodology to understand if the user validates his subscription in Pias indicates a higher likelihood of validating his subscription on Trindade in the future.

Sequential Pattern mining, sometimes called Sequential Itemset Mining, introduces a time component analysis essential on this problem. There are various Sequential Pattern Mining algorithms, as discussed in the Literature review chapter. The SPADE algorithm will be the one to confront this problem, which stands for sequential pattern discovery using equivalence classes. This algorithm works recursively to find the frequent patterns by starting with an item set with a length equals to one, then moving for the itemsets with a length of two, and so on. This will be implemented by using the R package named arulesSequences.

The frequent sequences and respective rules of three separate transaction matrices which generated from data provided by the Porto metro will be investigated in the following sections.

4.3.1 Example I

The first transaction Matrix which is going to be analyzed is the one given on Table 4.8. Since the process of building the transaction matrix is already described in the previous section, the focus will be to analyze the output of the SPADE algorithm, which will be the frequent sequences and the respective rules.

It is required to select a value for the minimum support value to run the SPADE algorithm, also known as minsup. We chose a minsup of 0.3 and the obtained frequent sequences are the following represented in the following table :

Sequence	Support
{A}	1.0000000
{B}	1.0000000
{C}	0.9545455
{D}	0.3636364
{C},{D}	0.3181818
{A},{B}	1.0000000
{C},{B}	0.9545455
{D},{B}	0.3181818
{D},{A},{B}	0.3181818
{C},{A},{B}	0.9545455
{A},{A}	1.0000000
{B},{A}	1.0000000
{C},{A}	0.9545455
{D},{A}	0.3181818
{D},{B},{A}	0.3181818
{D},{A},{A}	0.3181818
{C},{B},{A}	0.9545455
{C},{A},{A}	0.9545455
{A},{B},{A}	1.0000000
{D},{A},{B},{A}	0.3181818
{C},{A},{B},{A}	0.9545455

Table 4.9: Frequent Item Sets for Example I

The most frequent elements are given by the following table:

Elements	Count
{A}	15
{B}	12
{C}	8
{D}	8
Others	28

Table 4.10: Most Frequent Elements

As it is possible to analyze, the item {A}, which is related to the metro station Trindade, occurs the most. Since this is a central station where multiple lines cross, it is expected to be where the user validates most of his monthly subscription. The quality measures for the most frequent itemsets are given by the following table 4.11:

Measure	Value
Minimum	0.3182
First Quartil	0.3182
Median	0.9545
Mean	0.7273
Third Quartil	1.0000
Maximum	1.0000

Table 4.11: Quality measures for the support of the most frequent item sets

The minimum support value is very close to what was defined as minimum support, which matches the first quartile. The maximum value is also the same as the value for the third quartile, which is 1.

On the data set in the study, the items {A} and {B} has a support value of 1, indicating that the user validates his monthly subscription on Trindade and Faria Guimares, respectively, every time he uses the network, so it is expected that these two metro stations will appear together as a frequent pattern with support equal to 1. To better comprehend the acquired frequent item sets, the following bar plot was generated in figure 4.7, which presents the frequent itemsets with the highest support.

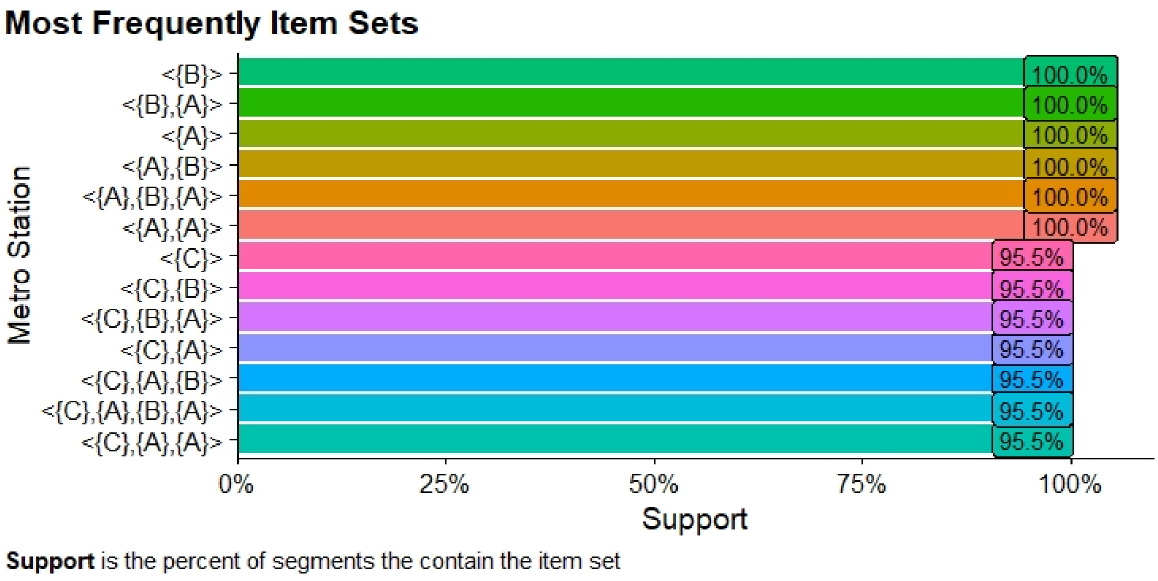


Figure 4.7: Most Frequent Item Sets from the first Transaction Matrix.

It is already feasible to gain a clear picture of this user's profile within the metro network. By analyzing the frequent itemsets with the highest support, it is possible to conclude that the user commutes with a high frequency between the stations $\{A\}$, $\{B\}$, and $\{C\}$, which are the stations "Trindade", "Faria Guimarães", and "Pias" respectively.

Because “Trindade” serves as a junction for the network’s many metro lines, it’s reasonable to assume that this user home and workplace are close to “Pias” and “Faria Guimares”. The collected frequent item sets can be turned into rules to determine the station where the user frequently begins his journey on the network.

With the obtained rules, it is possible to understand the sequence of the station where the user validates his monthly subscription, thus understanding where the user starts his journey and where it ends. By looking at the frequent itemsets with two or more items, it is possible to get some ideas of how the rules will look since the comma used to separate the items indicates a sequence.

The input for this R function will be the minimum confidence value instead of the minimum support value used to get the most frequent item sets. The confidence is given by the likelihood that a sequential rule $\{A\}- > \{B\}$ happens among the validations containing the item $\{A\}$, under the constraint that the item $\{A\}$ happens before the item $\{B\}$. This means that there is a certain probability that every time that the user validates his monthly subscription on the station $\{A\}$ it will also validate afterward on the station $\{B\}$. Confidence can also be interpreted as a conditional probability given a previous event, which in this case, is the validation on the station $\{A\}$. A high confidence value implies a high likelihood that the validation on the station $\{B\}$ happens in the future.

Another parameter that is used to evaluate the rules is the Lift. This parameter evaluates the strength of the association. Giving the previous example of the rule $\{A\}- > \{B\}$, the lift parameter will make it possible to understand how much more likely it is possible to see the item $\{A\}$ and $\{B\}$ together compared to what it would be expected if $\{A\}$ and $\{B\}$ were completely independent of each other. In other words, a high value of the parameter lift, which is when the lift is higher than “1”, implies that the presence of a pre-existing item set $\{A\}$ increases the probability that the item $\{B\}$ to happen on a future validation. Contrarily, when the lift assumes a low value, which is less than “1”, it suggests a negative dependence between the item sets.

The parameters Confidence and Lift, as shown in Table 4.12, were used to evaluate rules based on the preceding information.

Rule	Support	Confidence	Lift
<{A} -> {B}>	1.0000000	1.000	1.000
<{C} -> {B}>	0.9545455	1.000	1.000
<{D} -> {B}>	0.3181818	0.875	0.875
<{D},{A} -> {B}>	0.3181818	1.000	1.000
<{C},{A} -> {B}>	0.9545455	1.000	1.000
<{A} -> {A}>	1.0000000	1.000	1.000
<{B} -> {A}>	1.0000000	1.000	1.000
<{C} -> {A}>	0.9545455	1.000	1.000
<{D} -> {A}>	0.3181818	0.875	0.875
<{D},{B} -> {A}>	0.3181818	1.000	1.000
<{D},{A} -> {A}>	0.3181818	1.000	1.000
<{C},{B} -> {A}>	0.9545455	1.000	1.000
<{C},{A} -> {A}>	0.9545455	1.000	1.000
<{A},{B} -> {A}>	1.0000000	1.000	1.000
<{D},{A},{B} -> {A}>	0.3181818	1.000	1.000
<{C},{A},{B} -> {A}>	0.9545455	1.000	1.000

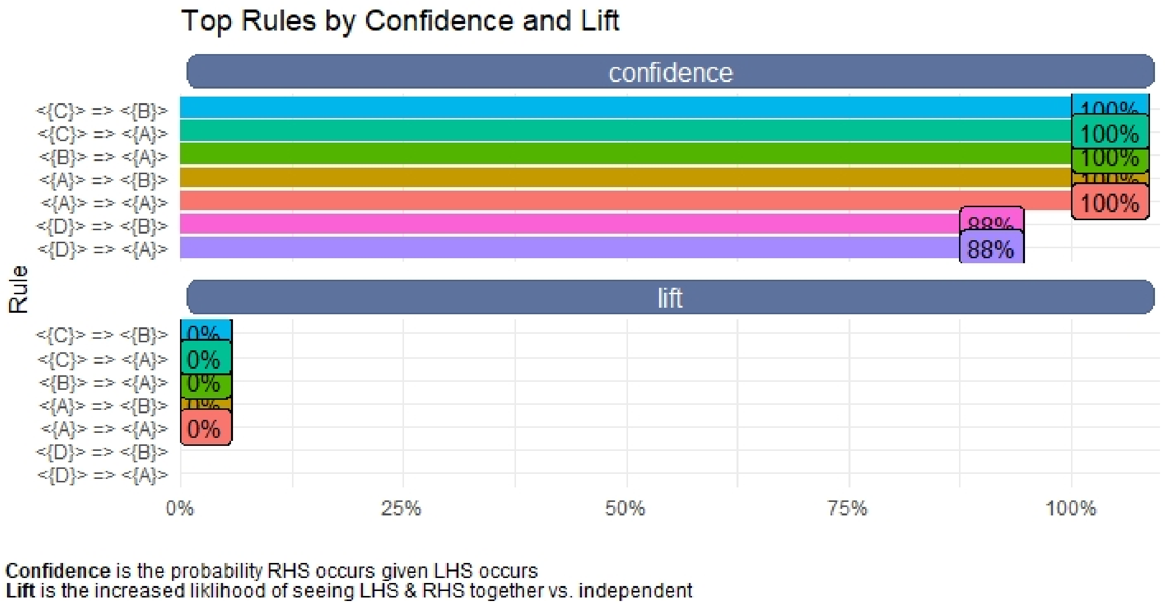
Table 4.12: Generated rules for Example I

To better visualize the most significant rules, Figure 4.8 displays the obtained rules according to the parameters of Confidence and Lift after redundant rules were removed.

A rule is categorized as redundant when a subset of the left side has higher confidence than the rule with more items on the left side, as it is mentioned by M. Zaki (2004). In other words, removing the redundant rules makes it possible to obtain the most simple rule with a higher confidence value without losing any relevant information.

On this scenario, for example if it is took in considering the rule $< \{D\}, \{B\} - > \{A\} >$ is redundant when compared to the rule $< \{B\} - > \{A\} >$ since the addition of the item $\{B\}$ does not make the confidence of the rule to increase, therefore is redundant.

Once all the redundant rules are removed, the following bar plot with all remaining rules was achieved:



Confidence is the probability RHS occurs given LHS occurs
Lift is the increased liklihood of seeing LHS & RHS together vs. independent

Figure 4.8: Rules obtained for Example I.

By removing the redundant rules, it is lost a considerable amount of rules. On the other hand, a more detailed understanding of the user's movement inside the metro network is achievable. For example, it is now able to focus on the stations from which the user validates his monthly subscription on a first and second instance. With this, is possible to understand better where the user starts and ends his journey on the metro network.

Although the time variables available are not taken into account in this analysis, which is essential for solving the proposed problem. Therefore, the next example will have a different variant of the transaction matrix that can be used as input for SPADE, which adds a time component to this analysis by adding the time of the day when the validation happens.

4.3.2 Example II

In this second example, the goal is to create a transaction matrix that contains not only the information from the previous example, but also a component related to the time of day when the validation occurred. The new variable obtained from the Time feature from the provided data set from Metro of Porto was used to do this.

The field Time indicates when a validation occurs during the day. It's in a format that allows to identify the exact moment the validation took place by including the hour, minutes, and seconds.

Because the variable Time was specified as a string, the first step was to convert it to a datetime format. This was done to avoid problems in the future when examining the data.

To make the procedure smoother, the new variable generated on the exploratory data analysis that contained information on the times when the validation took place was used. It is now feasible to combine the time information with the prior transaction matrix obtained in Example I by using this new variable. The next step was to determine how to best combine this data with the previous transaction matrix. It can happen in one of two ways:

1. Combine the time and the station where the validation occurred within the same item, such as $\{AHP2\}$. Because there will only be one item per event, every value of the "Size" variable will be one.

2. Combine the information of the time with the station where the validation happens on different Items. By selecting this encoding option, the transaction matrix will have two different items for each event, such as $\{A, HP2\}$, and the "Size" variable will be equal to two for each event.

By running tests on both choices, the findings for the frequent item sets and rules were equal. This was expected, given how identical the data utilized as input for the SPADE algorithm was for each event. The only difference was in the representation of the item sets and the information in the "Size" variable. This second format was chosen since the information is more simply evaluated in it and it isolates the station and time of day variables as separate items.

With this, the aimed transaction matrix was achieved. Every event has two items determined by the station and the corresponding schedule when the validation happened. The first five rows of the obtained transaction matrix are represented in Table 4.13:

sequenceID	eventID	Size	itemset
1	949	2	C,HP1
1	1011	2	A,HP1
1	1639	2	B,HP2
1	1644	2	A,HP2
2	810	2	C,HP1

Table 4.13: First five rows of the second Transaction Matrix.

In this example, it was chosen a value for minimum support of "0.3", identical to the value used on the previous example. The twelve obtained frequent sequences with the highest support are represented in the following Table 4.14 :

sequence	Count
{A,HP2}	0.8636364
{B,HP2}	0.8636364
{B,HP2},{A,HP2}	0.8636364
{A,Almoco}	0.5909091
{A,HP1}	0.5454545
{C,HP1}	0.5454545
{C,HP1},{A,HP1}	0.5454545
{A,Almoco},{B,HP2}	0.4545455
{A,Almoco},{A,HP2}	0.4545455
{A,Almoco},{B,HP2},{A,HP2}	0.4545455
{C,Almoco}	0.4090909
{A,HP1},{B,HP2}	0.4090909

Table 4.14: Frequent Item Sets for Example II

Where the most frequent elements are given by the following Table 4.15:

Elements	Count
{A,HP2}	14
{B,HP2}	14
{A,Almoco}	8
{A,HP1}	8
{C,Almoco}	8
Others	8

Table 4.15: Most Frequent Elements for Example II

As it is possible to analyze, the item {A,HP2} and {B,HP2} have the same count. This means that the user validated his monthly subscription the same amount of times in Trindade and Faria Guimarães on the period between 4 PM and 8 PM. It is reasonable to deduce that the user leaves his employment at Faria Guimares and then goes to Trindade, where he validates his monthly subscription once more, because both of these items may be seen as an item set with extremely high support, as shown in Figure 4.19.

The quality measures for the most frequent itemsets are given by the following Table 4.16:

Measure	Value
Minimum	0.4091
First Quartil	0.4091
Median	0.4091
Mean	0.4865
Third Quartil	0.5000
Maximum	0.8636

Table 4.16: Quality measures for the support of the most frequent itemsets

The minimum value for support is much lower than the preceding example's minimum support. This means that according to the codification of this Transaction Matrix, the SPADE algorithm could not find any frequent itemsets with support between "0.3" and "0.4". The first quartile is the same as the Median and the minimum value due to the high frequency of itemsets with a support value equal to "0.4091". The remaining measures are self-explanatory.

In order to get a better visualization of the given frequent itemsets, it was created the following bar plot on figure 4.9:

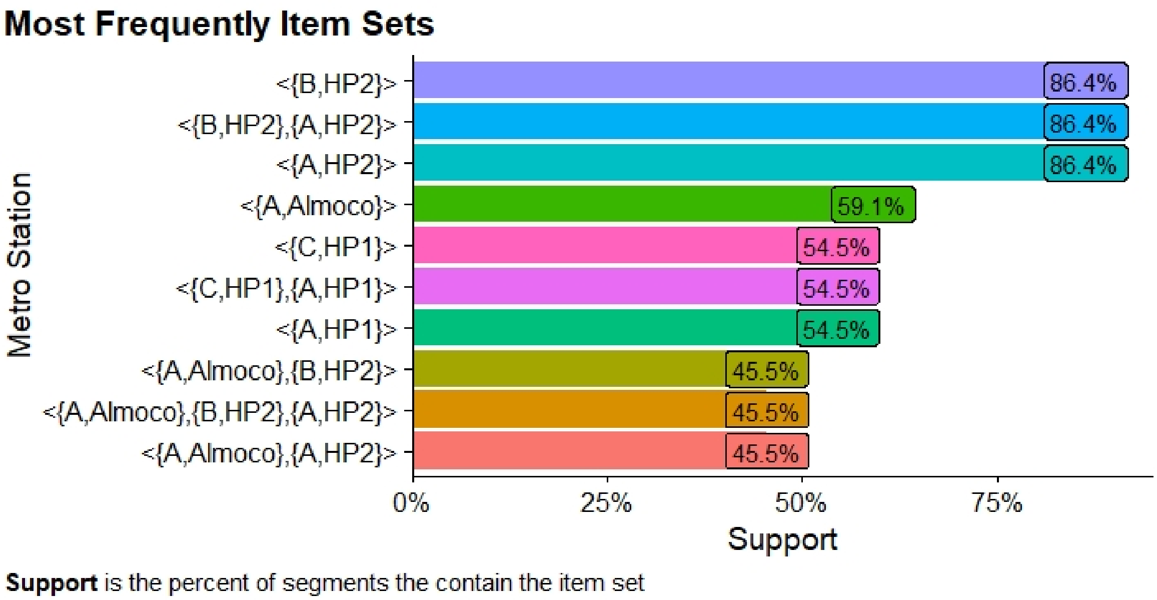


Figure 4.9: Most Frequent Item Sets for Example II.

It is possible to gain a better grasp of the collected item sets using the bar plot. As previously stated, the user between the hours of 4 and 8 p.m. from "Faria Guimares" to "Trindade" with a 86.4 % of support.

Another intriguing item set is $< \{C, HP1\} >$, which has a significant high support value and is the item set that contains the morning period with the highest support value. Knowing that the metro station “Trindade” serves as a transition between the network’s various lines, it can assumed that the user in study should leave around the metro station “Pias”.

When analyzing the frequent itemsets, it is logical to assume that the user lives where he validates more often his monthly subscription on the “HP1” period. A similar conclusion can be performed to determine the place where the user works, by analyzing the most frequent item set with the “HP2”. In this case is the item $< \{B, HP2\} >$, which indicates that the user validates his monthly subscription on “Faria Guimaraes” at the afternoon period with a support of 54.5 %.

The input given on R for the minimum confidence value to generate the rules was the same as the previous example, “0.6”. The obtained top fifteen rules sorted by the confidence value from high to low are represented in the below Figure 4.10:

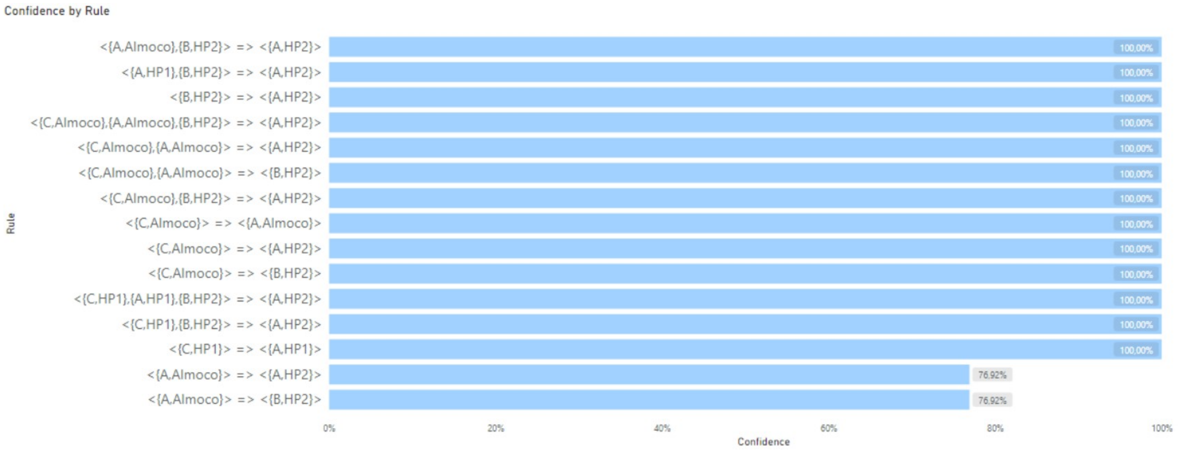


Figure 4.10: Rules obtained for Example II by Confidence.

As previously stated, and as a result of studying the collected rules, the confidence is given by the likelihood that the sequential rule $\{B, HP2\} -> \{A, HP2\}$ happens among the occurrences containing the item set $\{B, HP2\}$, under the constraint that the item set $\{B, HP2\}$ happens before the item set $\{A, HP2\}$. This means that there is a certain probability that every time which the user validates his monthly subscription on the station “Faria Guimarães” it will also validate afterward on the station “Trindade”, on the afternoon schedule “HP2”. Confidence can also be interpreted as a conditional probability given a previous event, which in this case is the occurrence of the item set $\{B, HP2\}$. A high confidence value implies a high likelihood that the item set $\{A, HP2\}$ happens in the future.

Figure 4.11 it is represented the obtained rules sorted by the lift value from high to low:

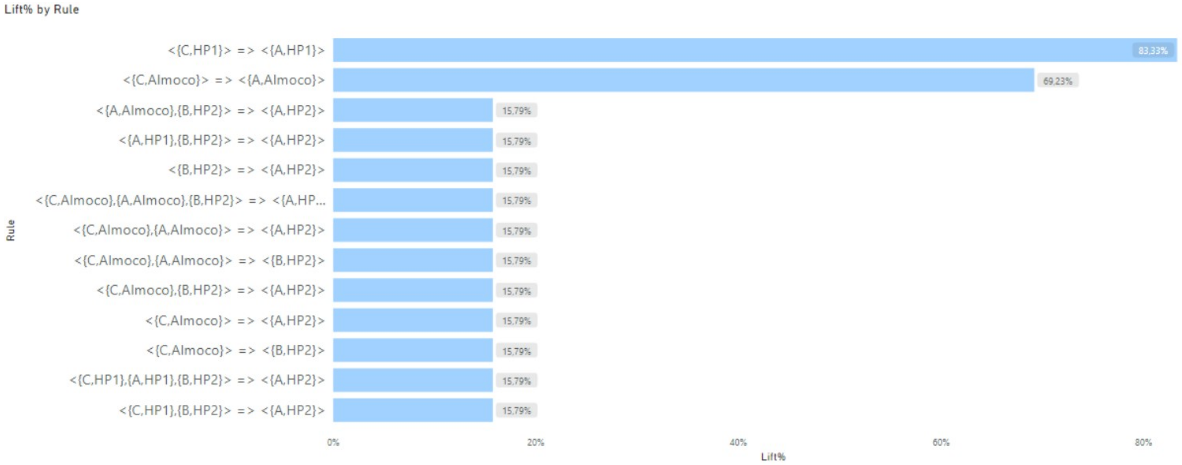


Figure 4.11: Rules obtained for Example II by Lift.

The Lift parameter, as was already mentioned before, evaluates the strength of the association. Giving the previous example of the rule $\{B, HP2\} - > \{A, HP2\}$, the lift parameter will make it possible to understand how much more likely it is possible to see the item sets $\{B, HP2\}$ and $\{A, HP2\}$ together compared to what it would be expected if $\{B, HP2\}$ and $\{A, HP2\}$ were completely independent of each other. In other words, a high value of the parameter lift, which is when the lift is higher than "1", implies that the presence of a preexisting item set $\{B, HP2\}$ increases the probability that the item $\{A, HP2\}$ to happen on a future validation. Contrarily, when the lift assumes a low value, which is less than "1", it suggests a negative dependence between the item sets.

The Figure 4.12 illustrates the acquired rules according to the parameters of Confidence and Lift after the redundant rules were deleted.

In this case, for example, if the rule is taken into consideration

In this case, for example, the rule $< \{A, HP1\}, \{B, HP2\} - > \{A, HP2\} >$ is redundant when compared to the rule $\{B, HP2\} - > \{A, HP2\}$ since the addition of the item $\{A, HP1\}$ does not make the confidence of the rule to increase, therefore is redundant.

Once all the redundant rules are removed, the following bar plot was achieved with all the top five rules according to the achieved confidence and lift.

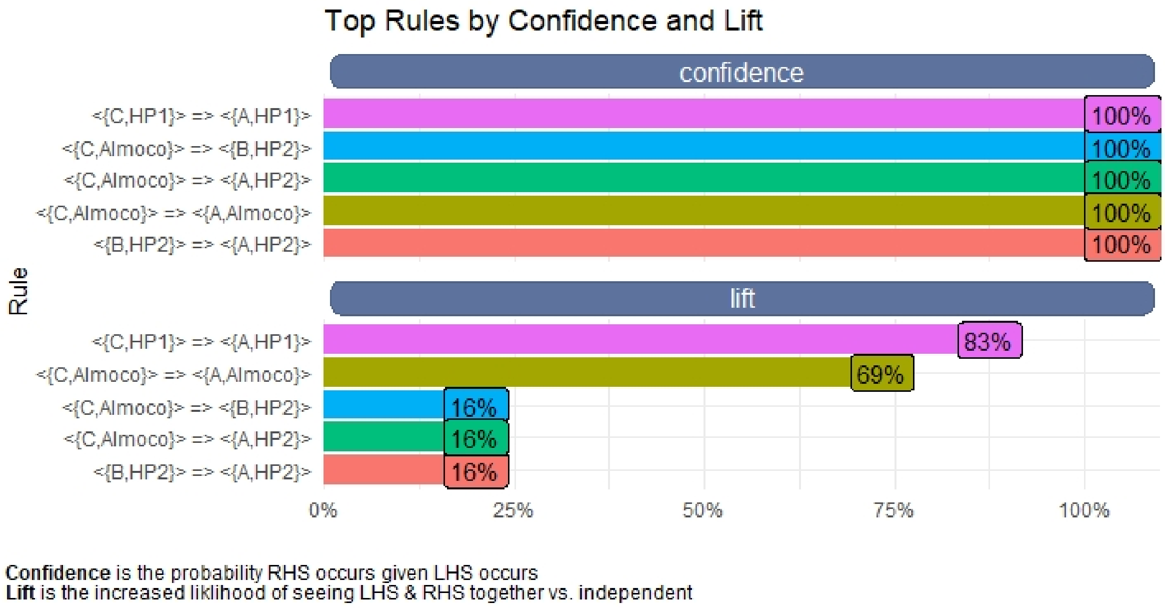


Figure 4.12: Rules obtained for Example II.

In the following example, instead of having a variable indicating the time where the validation happened, it will have a component indicating the day of the week when the validation happened

4.3.3 Example III

Instead of a time component indicating the day when the validation occurred, this third and final example will produce a transaction matrix with a component indicating the day of the week when the validation occurred. In order to achieve that it will be used the variable **diaSemana**, which is a categorical variable containing values from two to six indicating the day of the week as it was mentioned on the exploratory data analysis.

The purpose of including the component for the day of the week is to see if the acquired data can be supplemented with the day of the week when the validation occurred. For example, the validations for some users may differ depending on the day of the week. With this information, it may be feasible to supplement earlier data, resulting in a more accurate user profile.

Therefore, it was obtained the following transaction matrix on the Table 4.17:

sequenceID	eventID	Size	itemset
1	949	2	C,3
1	1011	2	A,3
1	1639	2	B,3
1	1644	2	A,3
2	810	2	C,4

Table 4.17: Transaction Matrix for Example III.

The format of this transaction matrix is identical to the previous example. The Size column has a value of two for every event. Each item have the station where the user performed the validation associated with the day of the week when it happens.

In this example, it was chosen a value for minimum support of "0.2" and it was obtained thirty-two different frequent sequences. The top twelve obtained frequent sequences with the highest support are represented in the following Table 4.18:

sequence	support
<{A,3}>	0.2272727
<{A,4}>	0.2272727
<{A,5}>	0.2272727
<{B,3}>	0.2272727
<{B,4}>	0.2272727
<{B,5}>	0.2272727
<{C,3}>	0.2272727
<{C,5}>	0.2272727
<{A,5},{B,5}>	0.2272727
<{C,5},{B,5}>	0.2272727
<{C,5},{A,5},{B,5}>	0.2272727
<{A,4},{B,4}>	0.2272727

Table 4.18: Frequent Item Sets for Example II

As it's possible to analyze, the support value for all the displayed rules is equal for all the frequent sequences. The same happens for the remaining rules that are not being displayed in Table 4.18. This happens because the SPADE algorithm computes the support by the ratio of a distinct count of how many times a item appears on the data with the amount of distinct values of the $sequenceID$. Since the user at least one time of the day validates his subscription on the metro stations $\{A\}$, $\{B\}$ and $\{C\}$, the support will be the same for each frequent item set.

Table 4.19 presents the count of the most frequent elements:

Elements	Count
{A,3}	10
{B,5}	10
{B,3}	8
{B,5}	8
{C,3}	8
Others	16

Table 4.19: Most Frequent Elements for Example III

By analyzing the given results, it is possible to say that this user validates his monthly subscription more often on Thursdays and Tuesdays. This can happen for multiple reasons, for instance, the month in the study could have public holidays, or simply the user did not use the network on some occasions for other specific reasons. This is the best that can be performed with the data currently available when adding the variable **diaSemana**.

Table 4.20 shows the quality measures obtained for the most frequent elements:

Measure	Value
Minimum	0.2273
First Quartil	0.2273
Median	0.2273
Mean	0.2273
Third Quartil	0.2273
Maximum	0.2273

Table 4.20: Quality measures for the support of the most frequent itemsets

As expected, the quality measures have the same values since all the obtained rules have the same values for the support measure. On Figure 4.13, the obtained frequent items with only two elements were shown:

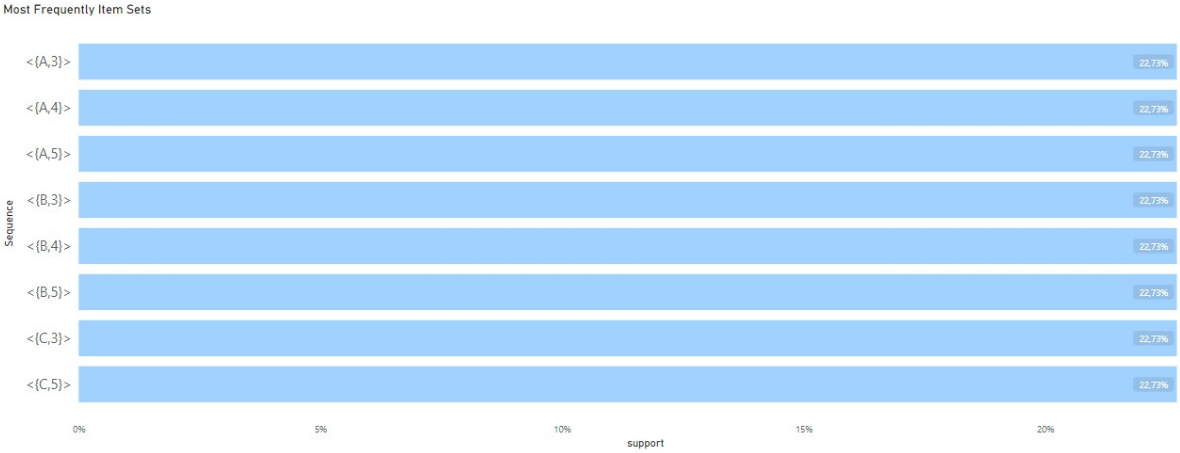


Figure 4.13: Most Frequent Item Sets for Example III.

The metro stations obtained from the most frequent item set list are identical to those obtained from the previous examples. This clearly shows that the user's most common pattern inside the network is to commute between "Pias", "Trindade", and "Faria Guimarães".

By adding this new component to the transaction matrix it was expected to gain new insights about the behavior of the user inside of the network. However, by analyzing the given results, it is only possible to conclude the metro stations where the user validates his monthly subscription more often. Therefore, it is possible to conclude that this information, for this specif user, will not add any more relevant information than the previous examples.

Nevertheless, the redundant rules where removed and plotted according the confidence measure, as it is possible to see on the following Figure 4.14:

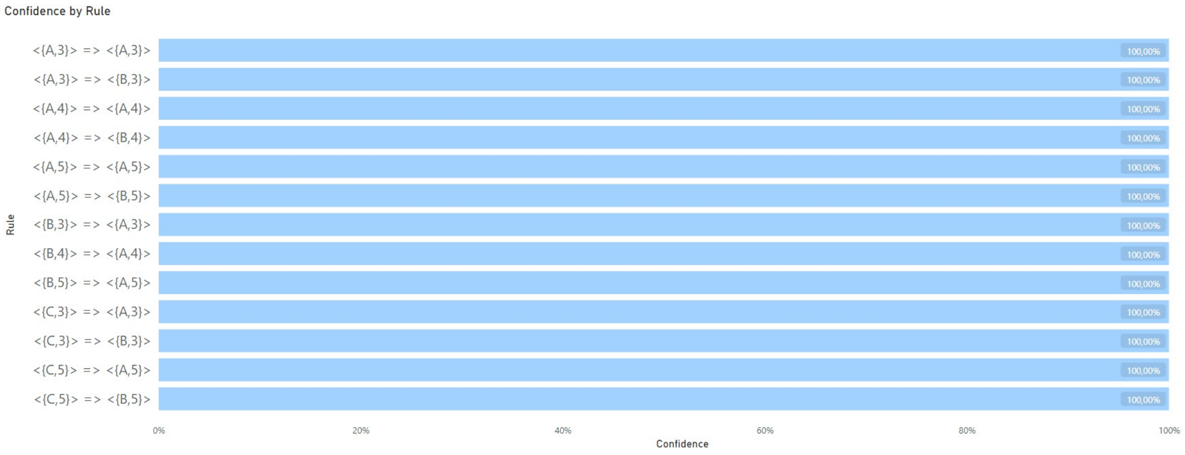


Figure 4.14: Rules obtained for Example III by Confidence.

As it was mentioned previously, and by analyzing the obtained rules, the confidence is given by the likelihood that a sequential rule $\{A, 3\}- > \{B, 3\}$ happens among the occurrences containing the item set $\{A, 3\}$, under the constraint that the item set $\{A, 3\}$ happens before the item set $\{B, 3\}$. This means that every time the user validates his monthly subscription on the station "Trindade", it will also validate afterward on the station "Faria Guimarães", on a "Tuesday". Confidence can also be interpreted as a conditional probability given a previous event, which in this case, is the occurrence of the item set $\{A, 3\}$. A high confidence value implies a high likelihood that the item set $\{B, 3\}$ happens in the future.

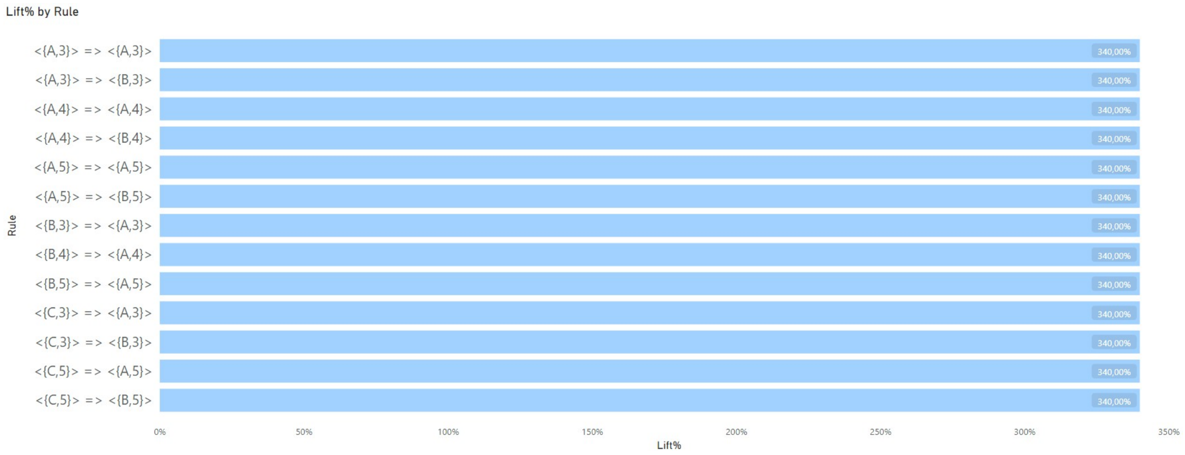


Figure 4.15: Rules obtained for Example III by Lift.

The Lift parameter, as was already mentioned before, evaluates the strength of the association. Giving the previous example of the rule $\{A, 3\}- > \{B, 3\}$, the lift parameter will make it possible to understand how much more likely it is possible to see the item sets $\{A, 3\}$ and $\{B, 3\}$ together compared to what it would be expected if $\{A, 3\}$ and $\{B, 3\}$ were completely independent of each other. In other words, a high value of the parameter lift, which is when the lift is higher than "1", implies that the presence of a preexisting item set $\{A, 3\}$ increases the probability that the item $\{B, 3\}$ to happen on a future validation. Contrarily, when the lift assumes a low value, which is less than 1, it suggests a negative dependence between the item sets.

The Lift value obtained for all of the rules in this scenario is extremely high. This makes sense because all of the obtained rules occur on the same weekday. Furthermore, based on the data, it can be assumed that when two separate validations, such as $\{A, 3\}$ and $\{B, 3\}$, occur on the same day, this rule will always be proven.

Exploring this transaction matrix was interesting in the sense that it could be when more users of the metro network are included. On in this particular case, it was possible to gain more insights from the previous example rather than this last example. Although the same can not happen when analyzing different users with different mobility patterns.

4.4 Analysis of Results

The previous examples represent three different approaches to analyze the profile of a user of the metro network by using sequential pattern mining. It is possible to perform an individual analysis of one person using this methodology, which can be further extended to analyze all the monthly subscribers of the metro of Porto network. To perform this analysis, it is needed to analyze all the validations of each user. Therefore it is essential to have as many validations per user as possible to achieve the most accurate results possible.

It is achievable to clearly comprehend that the user lives near station Pias and works near station Faria Guimarães using this method. It was also possible to create a list of the most frequent stations where the user in the study commutes at different times of the day, as well as a list of validations for each day of the week. On the basis of the exploratory data analysis, it was discovered that the user in question validates his monthly subscription more frequently in the afternoons and on Wednesdays.

After exploring the results achieved on the three examples, the example that provided the information needed to progress to the next stage of the thesis was the Example II. This example provides the sequence of stations where the user validates his monthly subscription and provides the period of the day when it happens. This component of the period of the day will be one of the inputs needed for the script to find if the user is affected by an eventual breakdown or not.

The following chapter of this thesis will focus on converting the findings of Example II into a Python script that will simulate a network outage on a certain station at a specific time of day and determine whether or not the user is affected by the breakdown.

Chapter 5

Solution Development

This chapter will go over all of the processes that were taken to get the Python code for the mentioned script. This solution will show how to use the previous results from sequential pattern mining to create a tool that can reliably anticipate whether a client will be impacted by a network failure. The final solution was created by focusing on a single network user, serving as a proof of concept on how to apply the same approach to all of the network's customers.

5.1 Implementation

The initial step in the implementation was to plan out how this script's algorithm would work.

The possibility of knowledge that the gathered data can retrieve was the driving force for using the result of Example II to develop the solution. It was previously feasible to determine which stations were closest to a user's home and workplace, therefore using the same logic, it should also be possible to determine whether the user will be at a specific station at a specific time of day.

By evaluating the data from the frequent itemsets acquired in Example II, the goal is to create a program that can anticipate if a user will be affected by a network breakdown. It will be necessary to determine the likelihood of a user being on a given station at a specific time of day in order to write the script. SPADE's frequent itemsets data will allow this information to be obtained. Setting a minimum threshold value for the support is required for the SPADE algorithm to determine the frequent itemsets.This allows you to keep only the item sets that occur the majority of the time, making the analysis more meaningful. For Example II it was set a minimum support value of 30 %. This means that the algorithm will search for only the items sets that happen at least 30 % of transactions, when considering the number of unique validations of stations per day divided by the number of days. For the current example, the minimum support value will remain the same.

To show how the information received from the SPADE algorithm may be utilized to determine metro stations affected by an eventual breakdown, a virtual metro network was developed, which includes a line that follows the connection between Pias and Hospital Sao Joao, as shown in the diagram below:

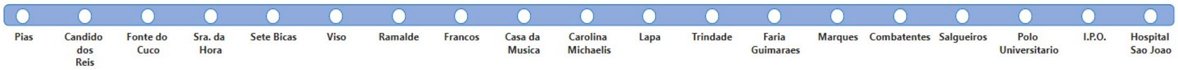


Figure 5.1: Virtual metro network.

This data can then be tailored to the Metro of Porto network, which includes a variety of transportation routes.

In order to simulate a breakdown, the final stage in designing the desired algorithm was to specify the inputs that were to be sent to the program, taking into account the previously mentioned description. To build a variable that represents a breakdown, the information must be represented in a manner that is compatible with the information collected from the frequent itemsets data. It is possible to create a breakdown by entering the following three values as input:

Input	Values
Station where the breakdown happened	Pias, Trindade, Sra da Hora, etc.
Direction where the trains traveled	1, 2
Period of the day	HP1, HP2, Almoco, Noite

Table 5.1: Inputs needed for the script.

The station input can take any value from what is represented in Figure 5.1. Regarding the direction input, it will be "1" if the breakdown happens on the direction $\{Pias\}$ -> $\{Hospital\ Sao\ Joao\}$ or "2" if the breakdown happens in the reverse direction. The period of the day can be any of the four mentioned different values.

The algorithm can then extract the stations that will be affected by the breakdown, which will be the one where the breakdown occurred as well as the ones that follow that station. For instance, if the breakdown happens in "Polo Universitário" on direction "1", the affected stations will be "I.P.O." and "Hospital São João". By having this information with all the affected metro stations and the period of the day when it happens, is it possible to cross this data to see if there is any match on the obtained frequent itemsets from the SPADE algorithm. For instance, a breakdown can be given by a list of item sets that contains the affected station and the corresponding period of the day when the breakdown happens. Considering the previous example given for the breakdown, the list will contain the following values:

- $\{PoloUniversitario, HP1\}$
- $\{I.P.O., HP1\}$
- $\{HospitalSaoJoao, HP1\}$

If one of the obtained frequent item sets is the same as one on the list generated by the algorithm there is a high likelihood that the user will be affected, and therefore should be warned about the breakdown. Because all stations following to the one where the failure occurs are designated as affected, this solution also allows users waiting for the metro at the following stations to be notified.

A pseudocode was built to better illustrate the created algorithm, which is represented in Algorithm 1.

Algorithm 1 Algorithm to find if a user is affected by a simulated breakdown on the network.

```
Use as input the data obtained from the most frequent item sets of Example II
Simulate a breakdown on the network by giving as input the metro station, the direction of
  the metro and the period of the day when the breakdown happens
Obtain a list with all the affected stations and the period of the day
for All the item sets given by the most frequently item sets data from Example II do
    if Obtained list with the affected metro stations combined with the period of the day
      equals to any of the obtained most frequently itemsets then
        The user can be affected by the breakdown
    end if
end for
```

Chapter 6

Conclusion

The major purpose of this thesis was to create an algorithm that can anticipate whether a network user will be affected by a network outage in the future. When using a sequential pattern mining approach, the most important component is to generate a transaction matrix, which is the algorithm's major input. All three examples provided by the SPADE algorithm offer a broad view of a user's behavior within the metro network. Each of the examples can explain with a high level of detail the behavior of a user within the metro network. As shown, when applying the SPADE algorithm, it was possible to achieve on every of the examples, not only the most commun stations where the user validates his monthly subscription, but also the most commun journeys of the user. Because it incorporated a time component relating to the time of day when the user validates his subscription, Example II ended up being the most relevant. The output of this example enables to determine if a user is affected by a network outage on a certain station at a specific time of day, as represented by the produced script. Furthermore, these examples provide the expertise to the Porto metro to adjust these examples to their standards if necessary.

The questions given at the start of this thesis were satisfactorily answered as a result of the thesis. With Example I, it was possible to discover not only the metro stations where the user validated his monthly subscription the most frequently, but also the most common path that the user takes. For instance, it was feasible to identify that the user travels between Pias and Faria Guimaraes frequently, implying that the user's place of work and residence are both close to these locations. When examining if the user is at a certain station at a specific time of day, including the time component relevant to the period of the day when the validation occurred improved the accuracy of the results in Example II. If a network outage happens on a specific station at a specific time of day, for example, the findings of Example II, which are a list of item sets containing the most commonly used stations during a specified period of time, can be utilized to determine whether or not the user in question is affected. On Example III, it was possible to get a list of frequently item sets that included the station where the validation took place, as well as the day of the week. Despite this, the support value for each item set was not particularly high. Considering all of the examples, it was possible to achieve the thesis's main goal of building an algorithm that can predict if a user will be affected by a network outage by studying the user's behaviors within the metro network using only the data obtained from Example II.

Taking on this assignment for the Porto Metro was relevant as it required to study and comprehend mobility patterns issues and how to address them in a business oriented solution approach, and the key is always to keep in track with the constant improving field that is Data Science.

The biggest challenge of this thesis was to implement a solution that had never been used before. When studying mobility patterns, typically, the methodology that is applied is Origin-Destination matrices. Due to the particularity of available data, tracking each user's sequence of movements on the metro network is essential. This thesis aims to develop an algorithm that can efficiently track which users will be affected by a breakdown on the metro network. To the best of our knowledge, this is the first service provided by a metro company. Indeed, it is possible in some cities to track what happens on a metro network. Still, most of it does not track the users who will be affected by an eventual breakdown.

Completing this thesis was the culmination of an incredible two-year learning experience unveiling the most fascinating lesson that Data Science allows for a wide range of scopes to handle a variety of business problems. The key is to keep up with the latest developments in this rapidly evolving sector.

Bibliography

Abrahamsson, T. (1998). Estimation of origin-destination matrices using traffic counts-a literature survey.

Aerde, M. V., Rakha, H., & Paramahamsan, H. (2003). Estimation of origin-destination matrices: Relationship between practical and theoretical considerations. *Transportation Research Record*, *1831*(1), 122-130. Retrieved from https://doi.org/10.3141/1831-14 doi: 10.3141/1831-14

Agrawal, R., & Srikanty, R. (1994). Fast algorithms for mining association rules. *The International Conference on Very Large Databases*, 487-499.

Augusto, G. (2018). A trip to work : Estimation of origin and destination of commuting patterns in the main metropolitan regions of haiti using cdr. , *3*, 133–166.

Ayres, J., Flannick, J., Gehrke, J., & Yiu, T. (2002). Sequential pattern mining using a bitmap representation. In *Proceedings of the eighth acm sigkdd international conference on knowledge discovery and data mining* (pp. 429–435).

Barron, A., Rissanen, J., & Yu, B. (1998). The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, *44*(6), 2743–2760.

Bauer, D., Richter, G., Asamer, J., Heilmann, B., Lenz, G., & Kölbl, R. (2017). Quasi-dynamic estimation of od flows from traffic counts without prior od matrix. *IEEE Transactions on Intelligent Transportation Systems*, *19*(6), 2025–2034.

Bell, M. G. (1991). The estimation of origin-destination matrices by constrained generalised least squares. *Transportation Research Part B: Methodological*, *25*(1), 13–22.

Bera, S., & Rao, K. (2011). Estimation of origin-destination matrix from traffic counts: the state of the art.

Bhuiyan, M. A., & Al Hasan, M. (2014). An iterative mapreduce based frequent subgraph mining algorithm. *IEEE transactions on knowledge and data engineering*, *27*(3), 608–620.

Brijs, T., Vanhoof, K., & Wets, G. (2004, 01). Building an association rules framework to improve product assortment decisions. *Data Min. Knowl. Discov.*, *8*, 7-23. doi: 10.1023/B:DAMI.0000005256.79013.69

Buchta, C., & Hahsler, M. (2020). Mining frequent sequences.

Calabrese, F., & Lorenzo, G. D. (2011). Estimating origin-destination flows using mobile phone location data. , 36–44.

Camerra, A., Palpanas, T., Shieh, J., & Keogh, E. (2010). isax 2.0: Indexing and mining one billion time series. In *2010 ieee international conference on data mining* (pp. 58–67).

Cascetta, E., Papola, A., Marzano, V., Simonelli, F., & Vitiello, I. (2013). Quasi-dynamic estimation of o–d flows from traffic counts: Formulation, statistical validation and performance analysis on real data. *Transportation Research Part B: Methodological*, *55*, 171–187.

Chang, G.-L., & Tao, X. (1999). An integrated model for estimating time-varying network origin–destination distributions. *Transportation Research Part A: Policy and Practice*, *33*(5), 381–399.

Charisma, F., & Development, M. C. (2018). Development of origin — destination matrices using mobile phone call data.

Doblas, J., & Benitez, F. G. (2005). An approach to estimating and updating origin–destination matrices based upon traffic counts preserving the prior structure of a survey matrix. *Transportation Research Part B: Methodological*, *39*(7), 565–591.

Fournier-Viger, P., Gomariz, A., Campos, M., & Thomas, R. (2014). Fast vertical mining of sequential patterns using co-occurrence information. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 40–52).

Fournier-Viger, P., Gomariz, A., Šebek, M., & Hlosta, M. (2014). Vgen: fast vertical mining of sequential generator patterns. In *International conference on data warehousing and knowledge discovery* (pp. 476–488).

Fournier Viger, P., Lin, C.-W., Rage, U., Koh, Y. S., & Thomas, R. (2017, 02). A survey of sequential pattern mining. *Data Science and Pattern Recognition*, *1*, 54-77.

Fournier-Viger, P., Nkambou, R., & Nguifo, E. M. (2008). A knowledge discovery framework for learning task models from user interactions in intelligent tutoring systems. In *Mexican international conference on artificial intelligence* (pp. 765–778).

Fournier-Viger, P., Wu, C.-W., Gomariz, A., & Tseng, V. S. (2014). Vmsp: Efficient vertical mining of maximal sequential patterns. In *Canadian conference on artificial intelligence* (pp. 83–94).

Fournier-Viger, P., Wu, C.-W., & Tseng, V. S. (2013). Mining maximal sequential patterns without candidate maintenance. In *International conference on advanced data mining and applications* (pp. 169–180).

Fu, T.-c. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence*, *24*(1), 164–181.

Ge, Q., & Fukuda, D. (2016). Updating origin–destination matrices with aggregated data of gps traces. *Transportation Research Part C: Emerging Technologies*, *69*, 291-312. Retrieved from https://www.sciencedirect.com/science/article/pii/S0968090X16300705 doi: https://doi.org/10.1016/j.trc.2016.06.002

Gomariz, A., Campos, M., Marin, R., & Goethals, B. (2013). Clasp: An efficient algorithm for mining frequent closed sequences. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 50–61).

Gong, Z. (1998). Estimating the urban od matrix: A neural network approach. *European Journal of operational research*, *106*(1), 108–115.

Hai, Y., Akiyama, T., & Sasaki, T. (1998). Estimation of time-varying origin-destination flows from traffic counts: A neural network approach. *Mathematical and computer modelling*, *27*(9-11), 323–334.

Han, J., Kamber, M., & Pei, J. (2011). Data mining concepts and techniques third edition. *The Morgan Kaufmann Series in Data Management Systems*, *5*(4), 83–124.

Hazelton, M. L. (2000). Estimation of origin–destination matrices from link flows on uncongested networks. *Transportation Research Part B: Methodological*, *34*(7), 549–566.

Hazelton, M. L. (2008). Statistical inference for time varying origin–destination matrices. *Transportation Research Part B: Methodological*, *42*(6), 542–552.

Högberg, P. (1976). Estimation of parameters in models for traffic prediction: a non-linear regression approach. *Transportation Research*, *10*(4), 263–265.

Hu, J., Yang, B., Guo, C., Jensen, C. S., & Xiong, H. (2020). Stochastic origin-destination matrix forecasting using dual-stage graph convolutional, recurrent neural networks. In *2020 ieee 36th international conference on data engineering (icde)* (p. 1417-1428). doi: 10.1109/ICDE48307.2020.00126

Huang, K.-Y., Chang, C.-H., Tung, J.-H., & Ho, C.-T. (2006). Cobra: closed sequential pattern mining using bi-phase reduction approach. In *International conference on data warehousing and knowledge discovery* (pp. 280–291).

Johnston, R., & Pattie, C. (2000). Ecological inference and entropy-maximizing: An alternative estimation procedure for split-ticket voting. *Political Analysis*, 333–345.

Lam, W. H. K., & Lo, H. P. (1991). Estimation of origin□ destination matrix from traffic counts: a comparison of entropy maximizing and information minimizing models. *Transportation Planning and Technology*, *16*(2), 85-104. Retrieved from https://doi.org/10.1080/03081069108717474 doi: 10.1080/03081069108717474

Lin, J., Keogh, E., Wei, L., & Lonardi, S. (2007). Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, *15*(2), 107–144.

Lin, N. P., Hao, W.-H., Chen, H.-J., Chueh, H.-E., & Chang, C.-I. (2007). Fast mining maximal sequential patterns. In *The international conference on simulation, modeling and optimization* (pp. 405–408).

Lo, D., Khoo, S.-C., & Li, J. (2008). Mining and ranking generators of sequential patterns. In *Proceedings of the 2008 siam international conference on data mining* (pp. 553–564).

Lu, S., & Li, C. (2004). Aprioriadjust: An efficient algorithm for discovering the maximum sequential patterns. In *Proc. intern. workshop knowl. grid and grid intell.*

Maher, M. J. (1983). Inferences on trip matrices from observations on link volumes: a bayesian statistical approach. *Transportation Research Part B: Methodological*, *17*(6), 435–447.

Mozolin, M., Thill, J.-C., & Usery, E. L. (2000). Trip distribution forecasting with multilayer perceptron neural networks: A critical evaluation. *Transportation Research Part B: Methodological*, *34*(1), 53–73.

Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., … Hsu, M.-C. (2004). Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on knowledge and data engineering*, *16*(11), 1424–1440.

Pei, J., Han, J., & Wang, W. (2007). Constraint-based sequential pattern mining: the pattern-growth methods. *Journal of Intelligent Information Systems*, *28*(2), 133–160.

Perrakis, K., Karlis, D., Cools, M., Janssens, D., Vanhoof, K., & Wets, G. (2012). A bayesian approach for modeling origin–destination matrices. *Transportation Research Part A: Policy and Practice*, *46*(1), 200 - 212. Retrieved from http://www.sciencedirect.com/science/article/pii/S096585641100098X doi: https://doi.org/10.1016/j.tra.2011.06.005

Pham, T.-T., Luo, J., Hong, T.-P., & Vo, B. (2012). Msgps: a novel algorithm for mining sequential generator patterns. In *International conference on computational collective intelligence* (pp. 393–401).

Pinto, H., Han, J., Pei, J., Wang, K., Chen, Q., & Dayal, U. (2001). Multi-dimensional sequential pattern mining. In *Proceedings of the tenth international conference on information and knowledge management* (pp. 81–88).

Pokou, Y. J. M., Fournier-Viger, P., & Moghrabi, C. (2016). Authorship attribution using small sets of frequent part-of-speech skip-grams. In *Flairs conference* (pp. 86–91).

Pramono, Y. W. T. (2014). Anomaly-based intrusion detection and prevention system on website usage using rule-growth sequential pattern analysis. *The International Conference on Advanced Informatics, Concept Theory and Applications*, 203-208.

Spiess, H. (1987). A maximum likelihood model for estimating origin-destination matrices. *Transportation Research Part B: Methodological*, *21*(5), 395–412.

Spiess, H. (1990). A gradient approach for the od matrix adjustment problem. , *1*, 2.

Srikant, R., & Agrawal, R. (1996a). Mining sequential patterns: Generalizations and performance improvements. In *International conference on extending database technology* (pp. 1–17).

Srikant, R., & Agrawal, R. (1996b). Mining sequential patterns: Generalizations and performance improvements. In *International conference on extending database technology* (pp. 1–17).

Tesselkin, A., & Khabarov, V. (2017). Estimation of origin-destination matrices based on markov chains. *Procedia Engineering*, *178*, 107–116.

Van Zuylen, H. J., & Willumsen, L. G. (1980). The most likely trip matrix estimated from traffic counts. *Transportation Research Part B: Methodological*, *14*(3), 281–293.

Van Zuylen, J. (1978). The information minimizing method: validity and applicability to transport planning. *New developments in modelling travel demand and urban systems*.

Wang, J., Han, J., & Li, C. (2007). Frequent closed sequence mining without candidate maintenance. *IEEE Transactions on Knowledge and Data Engineering*, *19*(8), 1042–1056.

Willumsen, L. G. (1978). Estimation of an od matrix from traffic counts–a review.

Xiong, X., Ozbay, K., Jin, L., & Feng, C. (2020a). Dynamic origin–destination matrix prediction with line graph neural networks and kalman filter. *Transportation Research Record*, *2674*(8), 491-503. doi: 10.1177/0361198120919399

Xiong, X., Ozbay, K., Jin, L., & Feng, C. (2020b). Dynamic origin–destination matrix prediction with line graph neural networks and kalman filter. *Transportation Research Record*, *2674*(8), 491-503. Retrieved from https://doi.org/10.1177/0361198120919399 doi: 10.1177/0361198120919399

Yan, X., Han, J., & Afshar, R. (2003). Clospan: Mining: Closed sequential patterns in large datasets. In *Proceedings of the 2003 siam international conference on data mining* (p. 166-177). Retrieved from https://epubs.siam.org/doi/abs/10.1137/1.9781611972733.15 doi: 10.1137/1.9781611972733.15

Yang, Y., Widhalm, P., Athavale, S., & Gonzalez, M. (2016, Mar.). Mobility sequence extraction and labeling using sparse cell phone data. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1). Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/9927

Y. Yang, P. W., & Gonz, M. C. (2016). Mobility sequence extraction and labeling using sparse cell phone data. , 4276–4277.

Zaki, M. (2001, 01). Zaki, m.j.: Spade: An efficient algorithm for mining frequent sequences. machine learning 42(1), 31-60. *Machine Learning*, *42*, 31-60. doi: 10.1023/A:1007652502315

Zaki, M. (2004, 01). Zaki, m.j. mining non-redundant association rules. data mining and knowledge discovery. *Machine Learning*, *9*, 223-248. doi: 10.1023/B:DAMI.0000040429.96086.c7

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

Zheng, Z., Zhao, Y., Zuo, Z., & Cao, L. (2009). Negative-gsp: An efficient method for mining negative sequential patterns. In *Conferences in research and practice in information technology series.*

Zhou, X., & Mahmassani, H. S. (2007). A structural state space model for real-time traffic origin–destination demand estimation and prediction in a day-to-day learning framework. *Transportation Research Part B: Methodological*, *41*(8), 823 - 840. Retrieved from http://www.sciencedirect.com/science/article/pii/S0191261507000173 doi: https://doi.org/10.1016/j.trb.2007.02.004