NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

Station Segmentation of Lisbon bicycle sharing system based on users demand and supply

Marisa Martinho Fernandes

Project work presented as the partial requirement for obtaining a Master's degree in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

STATION SEGMENTATION OF LISBON BICYCLE SHARING SYSTEM BASED ON USERS DEMAND AND SUPPLY

Marisa Martinho Fernandes

Project Work presented as the partial requirement for obtaining a Master's degree in Information Management, Specialization in Knowledge Management and Business Intelligence

Advisor: Mauro Castelli

January 2021

ACKNOWLEDGEMENTS

I would like to thank everyone that crossed my path during my academic life and contributed to enrich my life and to my personal and academic growth.

A warm thank you to my family, in special to my parents that supported me in this journey and worked hard all their lives to make sure that I had the opportunity to educate myself and pursuit for a better and prosperous future for me.

A huge thank you to Mauro Castelli that tirelessly supported me throughout my master thesis, without him to achieving this mark on my life would have been much more difficult.

All my friends that kept on offering me all the support that I need and kept on reminding me that I was almost there, a big thank you followed by a big hug, without you this journey would have been lonelier.

At last, but not least, an enormous thank you to myself for keeping on this project while working, for not stopping believing that it was possible though difficult, for not accepting the tempting thought of “I can leave it for the next year”, for keeping determinedly faithful to my ambition.

Everyone thinks of changing the world, but no one thinks of changing himself.
Leo Tolstoy

ABSTRACT

Bike-sharing systems are well known in the sustainable mobility field and have several aspects that need optimization and improvement. One of the most relevant aspects is station segmentation based on user demand and supply, and it is the focus of the thesis. The segmentation work has an enormous potential to reduce complexity in predicting the bicycle demand and supply, thus improving the overall quality of service.

Several machine learning algorithms were used to investigate the aforementioned segmentation task. This work considers two popular and well-known clustering algorithms to extract and analyze interesting patterns, like the difference between arrivals and departures throughout time and stations: the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) and the hierarchical clustering.

The algorithms are applied to the specific case of GIRA, the bicycle sharing system (BSS) of the city of Lisbon. The obtained results suggest that considering the variables under analysis, the optimal number of clusters to be used in a second phase of the BSS optimization (demand and supply forecast) is the same as the number of stations in the Lisbon BSS. The results are very insightful and allow future work to focus either on the demand forecast or the enrichment of the variables under study.

KEYWORDS

Machine learning; Timeseries segmentation; Bike-sharing systems; Sustainable mobility

Index

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

PCA	Principal Component Analysis
IST	Instituto Superior Técnico
DTW	Dynamic Time Warping
BSS	Bicycle Sharing Systems
FFBSS	Free-Floating Bicycle Sharing System
SBRP	Static Bicycle Repositioning System
DBRP	Dynamic Bicycle Repositioning System
ANN	Artificial Neural Network
DBSCAN	Density-based spatial clustering of applications with noise

1. INTRODUCTION

A bicycle sharing system (BSS) can be defined as a network of bicycles spread in a city, available to users. A user can take a bicycle at the starting point, drive it until the destination point and leave the bicycle where the trip finished, moment in which the bicycle will become available to other users.

Bicycle sharing systems gained popularity in recent years and became a popular service in major cities. The first BSS world-wide was introduced in Amsterdam in 1965 (Shaheen, 2012) and the bicycles were unlocked and placed around the city. The next BSSs went through some changes and challenges: some of them were paid, and some have suffered from theft or even vandalism. Throughout the years, BSSs became more popular around the world and by the beginning of April of 2020 there were 2102 cities with BSS, with approximately 17866900 self-service public use bicycles and electric assisted bicycles. (DeMaio & DesJardins, s.d.).

Cities are characterized by an agitated life, traffic congestion, long waiting times between public transports connections, and bad air quality. BSSs can be seen as a way of counteracting or minimizing some of these issues. In Albiński and coauthors (Albiński, Fontaine, & Minner, 2018) work, a BSS is presented as a good alternative transport mode with respect to the existent ones (train, tram, metro and bus). In fact, if the city has a well-structured bicycle infrastructure, riding a bicycle can be the fastest way to go from one place to another and, as a consequence, a time-saving alternative. Besides that, it has a positive impact on user's health due to the exercise done by riding a bike. Moreover, the reduction of greenhouse gas emissions is also a factor that contributes for the BSS adoption and its increasing popularity. Forma and coauthors (Forma, Raviv, & Tzur, 2015) and Shui and coauthors (Shui & Szeto, 2018) pointed out that BSS is an environmentally friendly option and it can complement the public transportation.

This study uses machine learning techniques to cluster docking stations with similar demand and offer behaviors, taking into account the BSS of Lisbon. The city made available a BSS in 2017 with the project GIRA, and by September of 2019 it counted with 81 stations and around 600 (both casual and electric) shared bicycles. There are two types of BSS: the so-called *traditional BSS*, which is characterized by having the bicycles associated to a docking station and the *free-floating BSS (FFBSS)* in which there is not a fixed place for each bicycle and the bicycles free-float around the city (Liu, Szeto, & Ho, 2018). The BSS of Lisbon is a traditional BSS.

In the context of a traditional BSS, each journey typically starts from a specific docking-station, finishes in another, and the user does not need to return to the initial station. This type of behavior contributes to empty and full stations. It is important to note the BSSs are efficient when stations are balanced (not empty or full). Therefore, unbalanced stations lead to inefficiency in BSS which also leads to unsatisfied users and, consequently, to a potential loss of users. (Dell'Amico, Iori, Novellani, & Subramanian, 2018) The unbalanced station is also a problem in the context of the GIRA project that will be addressed in this work.

There are two common approaches to minimize the problem of unbalanced BSS, being the first user-based and the second truck-based. The user-based approach is usually less costly than the latter, but it rarely solves the problem by itself. In particular, the user-based approach gives a reward/incentive to users that start a trip in stations with an excess of bicycles and to the users that end a trip in stations with a deficit of bicycles. The truck-based approach addresses the problem by

picking up bicycles in stations with an excess of bicycles and drop them in stations with a deficit of them (Rudloff & Lackner, 2014). Clearly, this second approach requires the use of trucks and operators that are responsible for periodically guaranteeing the correct balance in each station.

The repositioning problem is a well-studied problem, in which two types of repositioning are commonly considered: the static bicycle repositioning problem (SBRP), where the reposition is done during the night when the BSS is closed or with little activity, and the dynamic bicycle repositioning problem (DBRP) in which the reposition is performed during the day (Dell'Amico, Iori, Novellani, & Subramanian, 2018). Focusing on the GIRA project, the bicycle reposition is performed during the night. Thus, this work falls under the dynamic bicycle repositioning problem (DBSP).

Considering the existing literature presented in chapter 2, and focusing on the framework proposed by Regue and Recker (Regue & Recker, 2014), there are some prior steps to the repositioning problem, namely 1) understanding the demand, 2) finding the optimal occupational rate for each station throughout the day and, subsequently, optimizing the redistribution of the bicycles. This study is focused on the first step of the framework suggested by Regue and Recker.

In this study, the focus will be on understanding the demand and the offer at the docking station level. More in detail, the demand corresponds to the bicycles that leave the docking station at a certain time, while the offer corresponds to the bicycles that arrive to the docking station. To reduce the dimensionality of the problem, instead of modeling the demand and offer for each docking station, we will create clusters of docking stations presenting similar behaviors of bicycle variation (difference between the offer and the demand) using machine learning techniques.

This work is organized as follows: Section 2 presents a critical literature review that explores previous and related work. In section 3, the data attributes and the context of this study are fully detailed. Section 4 describes the pre-processing operations considered in this study. Section 5 discusses dimensionality reduction: in section 5.1 the concepts of unsupervised learning and supervised learning are presented, and in section 5.2 the specific case of unsupervised learning applied to time-series is considered; in section 5.3 the subject of distance measure between time-series is discussed, while section 5.4 presents the algorithms used in this study. Section 5.4 defines, and the measure used to compare the models developed. In section 6, the results are presented and compared. Finally, section 7 presents the conclusions and discusses possible future work.

2. RELATED WORK

Recent years have seen a rising interest in the definition and application of techniques and algorithms to optimize BSS tasks. This is mostly due to the widespread of BSS in the most important cities worldwide that contributed to the popularity of these systems. With respect to the problem addressed in this work, there is one main area of interest: the clustering of docking-stations with similar usage behaviors to ease the forecasting of future demand and offer (variation of the number of bicycles at station level).

Forecasting the demand and/or offer of the bicycles at the docking-station level allows gaining insights concerning the number of bicycles or docks available in future periods. Armed with such information, it will be possible to improve the dynamic redistribution plans in order to sustain the BSS balanced as long as possible (Rudloff & Lackner, 2014). The work of Regue and Recker (Regue & Recker, 2014) also highlights the importance of demand forecast and its pivotal role in reaching a more efficient user-based redistribution strategy by having dynamic incentives (instead of static, which do not adapt to changes in the demand).

Research shows evidence that clustering stations with similar temporal activity patterns and further use those clusters for demand forecast can improve the demand forecast capability and its results. In the work of Froehlich and coauthors (Froehlich, Neumann, & Oliver, 2009), temporal and spatiotemporal patterns regarding the number of bicycles checked out at each station were considered. The objective was to group stations with similar behaviors/activity, using clustering techniques and DTW as a distance measure. The DTW distance measure was used to overcome the limitations of the Euclidean distance because the authors were interested in allowing temporal shifts. The clusters identified were subsequently analyzed to forecast station-level demand. Vogel and coauthors (Vogel, Greiser, & Mattfeld, 2011), focused on understanding activity patterns (temporal and spatial) with the use of clustering techniques to group stations according to bicycle delivery and pick-up. These clusters were evaluated using Dumn and Silhouette index (the higher the value the better the clustering). The authors suggested, for future work, to use the clusters identified in the work as support for the demand forecast task. Besides improving the demand forecast results, clustering the stations also reduces the number of models needed to forecast demand. Instead of having as many models as stations, the clustering techniques group stations with similar behavior, thus allowing to have as many models as clusters.

The work of Rudloff and Lackner showed the dependency of stations on time and weather. (Rudloff & Lackner, 2014). As this work identifies with the previous statement, it is considered a time-series clustering problem. Reddy and Aggarwal (Reddy & Aggarwal, 2013) empirically showed that the distance measure between observations in time-series problems is of such importance that, in some cases, its choice is more important for reaching a satisfactory result than the choice of the clustering technique. DTW is a well-known distance/similarity algorithm for time-series that finds the optimal match between two time series allowing time shifts. This is a valuable characteristic when addressing problems like docking-station clustering. (Izakian, Pedrycz, & Jamal, 2015)

The work of Liu (Liu, Sun, Chen, & Xiong, 2016) highlighted that there are two main issues in the process of solving the bike sharing rebalancing problem: estimate the ideal inventory level for each station and the vehicle routing optimization. To address the estimation of the inventory level, the authors created two predictors that will support the inventory level estimation: the first predictor estimates the bike drop-off demand, and the second predictor estimates the bike pick-up demand. Regarding the second part of the problem, the bike sharing rebalancing problem, the authors first clustered the stations to be rebalanced and, subsequently, they applied the vehicle routing problem to each cluster individually.

Elhenawy and Rakha (Elhenawy & Rakha, 2017) approach to the static bicycle rebalancing problem (SBRP) has two steps, being the first the tour construction algorithm that constructs N optimized tours, each tour with M stations to be visited. The second step consists of improving the tours previously defined by finding the best route (with the lower cost) for each tour.

Caggiani and Ottomanelli (Caggiani & Ottomanelli, 2013) underlined that in the bicycle rebalancing problem the first essential step is to predict future demand and they proposed an approach to address that task. The authors use two feed-forward Artificial Neural Networks (ANN) with back-propagation that are able to relate the hour od the day with the entering or exited bicycles number. The first ANN estimates the number of bicycles arriving at each station and the second ANN estimates the number of bicycles leaving from each station.

Zhou and coauthors (Zhou, Wang, Zhong, & Tan, 2018) called attention to the difference between station-level demand forecast and forecasting the demand of the whole BSS and to the inability that a model dedicated to forecasting the whole demand has in suiting the stations demand behavior when compared to the station-level demand forecast model. The authors propose a hybrid model based on a Markov chain to forecast station-level demand.

Schuijbroek and coauthors (Schuijbroek, Hampshire, & van Hoeve, 2017) propose a two-step approach to bicycle sharing system rebalancing. First stations are clustered such that the cluster is "self-sufficient" in terms of inventory levels. This way the target inventory levels that make the station balanced can be reached by performing within-cluster bicycle drop-offs and pick-ups. The authors propose the decomposition of the routing problem into as many smaller routing problems as clusters identified in the first step. For each cluster/routing problem the author assigned one vehicle, thus reducing the combinatory complexity of the routing problem.

Hua and coauthors (Hua, Chen, Zheng, Cheng, & Chen, 2020) studied the case of free-floating bike sharing systems, more specifically the estimation of the parking demand with a final goal of recommending a facility construction for parking planning. The first step to accomplish that was to create a virtual station that would represent a set of bicycles with similar spatiotemporal characteristics. In order to create those, the authors proposed a spatiotemporal clustering technique. The approach divides the spatiotemporal clustering into two layers: the first is the temporal clustering in which time dependent patterns of the trips are considered, while the second layer performs a spatial aggregation based on the temporal clustering resulting from step one. To perform the clustering, k-means and DBSCAN were used.

Freng and coauthors (Feng, Affonso, & Zolghadri, 2017) focused their work in analyzing the bike sharing system of Paris, the Vélib. The authors used unsupervised learning techniques to process and identify patterns among stations and group stations with similar behavior in clusters and further use those clusters for supporting the system control and redesign. Are highlighted two complications, first, the curse of dimensionality of analyzing station by station and second the inexplicability of considering all stations as one. As a solution for these two problems, it is proposed to group stations with similar availability (using clustering techniques) of bicycles in the same cluster and then analyze that cluster. Regarding clustering techniques, k-means and hierarchical clustering were used with the advantage of hierarchical clustering flexibility of determining how many clusters should be considered.

Feng and coauthors (Feng, Chen, Du, Li, & Jing, 2018) pointed out that demand forecast is a necessary step as it is the basis for the redistribution problem, and that the bike usage patterns are more regular when using clusters of stations instead of a single station. The proposed framework to address the demand forecast problem consists of four steps. First, a station clustering based on the bike usage that is based on a Spectral Clustering algorithm. Second, the prediction of the number of check-outs in the entire BSS using Gradient Boost Regression Tree (GBRT) algorithm. Third, check-outs prediction at cluster level also using GBRT: this number is calculated by estimating the importance (i.e., weight) that each cluster has of the entire BSS. Fourth, as a journey that starts in one station finishes in another, the authors suggest predicting the number of check-ins at the cluster level (also using GBRT) by estimating the inter-cluster transition proportions.

Xu and coauthors (Xu, Ying, Lin, & Yuan, 2013) focused their work on station segmentation for Hangzhou Public Bicycle System by proposing an improved k-means algorithm. The authors pointed out the common use and good performance vs complexity of k-means in clustering problems, but also a disadvantage derived from the random initial cluster centers and its sensibility to them. To overcome this disadvantage, they proposed to use Simulated Annealing, an optimization algorithm, to optimize the initial cluster centers values.

Zhao and coauthors (Zhao, Hu, Liu, & Meng, 2019) proposed a clustering model for understanding patterns in human behaviors applied to Beijing bicycle-sharing system. The authors first transform spatiotemporal points into temporal sequences. Secondly, they apply clustering techniques to those temporal sequences in order to discover patterns in the data. Density-based spatial clustering of applications with noise (DBSCAN) was used as a clustering technique with dynamic time wrapping (DTW) as a measure distance.

3. DATA UNDERSTANDING

This study is under a partnership, with municipality of Lisbon, which made available the initial data. The data delivered was composed by two csv files, the first one with data regarding bicycle station information and the second with data regarding bicycles trips. In order to have a better understanding of the fields containing the csv files, the Table 1 and Table 2 can be consulted.

Table 1 - Bicycle station file - information

Field name	Description	Data type
Id	Row unique identifier	Numeric
Geom	Geo location of the station	Hexadecimal
Id_expl	Station unique identifier	Numeric
Id_planeamento	Station unique identifier	Numeric
Desig_comercial	Station unique identifier and the station name	String
Tipo_servico_niveis	Type of serve (A and B) which both refer to EMEL service	String
Num_bicicletas	The number of bicycles present in the station	Numeric
Num_docas	The number of docks in the station	Numeric
Racio	The ratio of bicycles in the station (num_biciclestas/num_docas)	Numeric
Estado	The station state (active, repair, stock)	String
Update_date	The datetime of the snapshot	Date

Table 2 - Bicycle trips file - information

Field name	Description	Data type
Id	Row unique identifier	Numeric
Date_start	Datetime for the beginning of the trip	Date
Date_end	Datetime for the end of the trip	Date
Distance	Distance of the trip in meters	Numeric

Station_start	Station unique identifier from which the trip began	Numeric
Station_end	Station unique identifier from which the trip ended	Numeric
Bike_rfid	Bicycle unique identifier	Hexadecimal
Geom	Geo location of the station	Hexadecimal
Num_vertives	Geographical information (number of vertices)	Numeric
Tipo_de_bicicleta	Type of bicycle (C: Non-electrical, E: Electrical)	String

4. PRE-PROCESSING

This chapter will explore one of the most time consuming and also one of the most important tasks for the success of the algorithm's results. (Kamber, Han, & Pei, 2011)

The approach taken in this chapter was guided by (Witten, Pal, Hall, & Frank, 2016). The authors of the book split pre-processing step into 4 major categories:

1. **Data cleaning** routines are applied to have clean the data. Those routines are put in practice through: filling missing values; smoothing noisy data; removing outliers and identifying and resolving inconsistencies or incoherences.

2. **Data Integration** is used when there is the need or the possibility to get more data from another data source. Getting more data, will very likely enrich the explanability of the phenomena under analysis. Data integration is not all roses, it is common that with it, redundancies, noise and inconsistencies will rise. Therefore, it's important that this step is explored along side with data cleaning.

3. **Data Reduction** goal is to have a dataset with less variables but with the same or almost the same explanability of the phenomena. There are some techniques that can be used and fall under data reduction step, such as: dimensionality reduction techniques (e.g., PCA, factor analysis, forward feature selection, etc.), feature subset selection (e.g., irrelevant or redundant features) or feature creation (e.g., creation of new features that summarize others such as creating the variable duration instead of having date start and data end).

4. **Data Transformation** step is responsible for transforming data (if needed) into a state that will make the modeling process mode efficient. This process includes several possible techniques, such as: feature construction (e.g., creation of new features from existing ones), aggregation (e.g., aggregate data by datetime every 10 minutes instead of having data to the second), normalization (e.g., scale features to a smaller range) or discretization (e.g., convert or partition continuous values into intervals or into new features).

4.1. DATA CLEANING

This sub-chapter will go deeper in how the first step of pre-processing, the data cleaning, was applied in the context of this study. The steps taken will be described alongside with some specific example to a better understanding of the work done:

Resolve incoherencies

1) As it was mentioned in Data understanding chapter, the columns "id_expl", "id_planeamento" and "desig_comercial" (the numbers in the beginning) reference the same value, the station number. Therefore, those were checked against each other to make sure that its values were correct. In the majority of the cases, the values were coherent, in the remaining cases where the values were not coherent, the data was corrected having the "desig_comercial" number has a decision maker. For example, as it can be seen in Table 3, the id_expl and

id_planeamento are the same but the design_comercial isn't. As the desig_comercial has prevalence over the others, the records in question were corrected to 103.

Table 3 - Incoherence example 1

Id_expl	Id_planeamento	desig_comercial
1	1	103-Jardim da Água

2) The second case of incoherencies verified is the verification of the "desig_comercial" values. The stations names were analyzed, and some irregularities were noticed under the following cases categorization:

   a. The id was the same, but the name was slightly different.

   b. The name was the same, but the id was different (typically sequential, e.g., 224 and 225).

   c. There was no name, only id.

   In order to resolve these cases, the website (https://www.gira-bicicletasdelisboa.pt/descobre-as-estacoes/) where resides the official map with all the stations in Gira's network was taken under consideration and observation.

   Regarding case a), the station names were corrected to the official name (the name present in the official website). The cases observed can be seen in Table 4.

Table 4 - Incoherence example 2

Desig_comercial-1	Desig_comercial-2	Desig_comercial-decision
488 - Rua Fernando Namora N35 / Rua António Quadro	488 - Rua Fernando Namora n35 / Rua António Quadro	488 – Rua Fernando Namora / Rua António Quadros
410 - Rua da Mesquita / Rua Dr. Júlio Dantas	410 - Rua da Mesquita /Universidade Nova de Lisboa	410 - Rua da Mesquita /Universidade Nova de Lisboa

Regarding case b), the two cases identified were correct, meaning that there were two stations with the same name. The cases observed can be seen in Table 5.

Table 5 - Incoherence example 3

Desig_comercial-1	Desig_comercial-2
224 - Martim Moniz	225 - Martim Moniz
307 - Marquês de Pombal Rua Dr. Júlio Dantas	308 - Marquês de Pombal

Regarding case c), the decision made was simply assigning the official name to the ones that were missing it. The example can be seen in Table 6**Error! Reference source not found.**

Table 6 - Incoherence example 4

Desig_comercial-1	Desig_comercial-2	Desig_comercial-decision
303	303 - Avenida da Liberdade / Rua das Pretas	303 - Avenida da Liberdade / Rua das Pretas

Remove noisy data

Regarding removing or not considering certain data, the columns "estado" specifies if the station is active, in repair or in stock. Only the records associated with the "estado" in stock or in repair were removed from the dataset. That choice was made, based on the fact that the phenomena under analysis only studies the stations that are active and by consequent have trips associated.

4.2. DATA INTEGRATION

Data integration chapter will go through the taken steps in order to get more features that can explain the phenomena under analysis. Through those steps, data from 3 different external sources were used:

1) **Portuguese holidays website**: from this website, it was possible to gather all the official Portuguese holidays for the 2018 year and further convert those into a python dictionary.

2) **Instituto Superior Técnico weather station API:** IST has a free API with weather data, based on a weather station located in Alameda. In this case, Alameda was considered representative of the city of Lisbon has a weather proxy. The data collected from this API contains hourly data.

3) **Sunrise and sunset API:** In order to have the data needed to further understand when it was daylight or not, data regarding the sunset and sunrise time in the city of Lisbon was collected. The data is a datetime with detail until the second.

Table 7 - New features information

Variable	Description	Source
Sunset time	Containing the datetime (until seconds) of sunset and sunrise for each day	https://sunrise-sunset.org/api
Sunrise time		
Portuguese holidays	Contains all official holidays.	https://www.calendarr.com/portugal/calendario-2018/
Total precipitation	How many millimeters of rain	http://meteo2-ciist.ist.utl.pt:8080/api
Atmospheric pressure	Atmospheric pressure in milibars	
Solar radiation	Solar radiation in watts per square	
Relative humidity	Percentage of humidity	
Temperature	Temperature in Celsius degrees	
Wind direction	Degrees of wind direction	
Wind gust	How strong is the wind gust in meters per second	
Wind speed	The wind speed in meters per second	

4.3. Data Reduction

This chapter will explain what was done in terms of data reduction. As it was explained in Pre-processing, there are three major set of actions that usually are performed, from which only two were used:

1) **Feature subset selection:** Some features were no longer considered due to the fact that either they didn't add information that wasn't already in other variables, or they didn't have complete information in order to be used. The cases were:

   a. "id_expl" and "id_planeamento" since their information is already in "desig_comercial"

   b. "estado" because the dataset will only have records with "estado" to active, so it wouldn't add new information.

c. "tipo_servico_niveis" because in the file that has information from stations, this column has incomplete information, or in other words, the data in this column is not sufficient or conclusive about how many electrical bicycles or normal ones are in the station at each time. Therefore, to assure quality of the data, it can't be used.

d. "bike_rfid", "geom", "num_vertices" are geographical variables. Since in this study, the geographical dimension won't be considered, these variables won't apply.

e. "id", since it is only a unique number for each trip and won't add relevant information.

f. "distance", this feature is also incomplete, more than 49% of the dataset has null values. Since it's not a good approach to interpolate 49% of the data, the feature was removed from the analysis.

2) Feature creation

a. A variable called "duracao_min" was created, which is a product of the difference between the date start and the date end. Instead of 2 features we now have 1 with the same information.

4.4. DATA TRANSFORMATION

In this chapter all the transformations in the data will be explained. The transformations were made into three major types:

a) Binary transformation was used in the case of the holidays and daylight variables. In the holidays case, now that variable has value 1 if it was holiday in Portugal and 0 if it wasn't. In the daylight case, from the sunset and sunrise datetimes, now the variable has also binary values, 1 if it was daylight and 0 if it wasn't.

b) Aggregation was used for the time dimension. The decision of aggregating the time dimension into bins of 20 minutes was made because of 2 reasons, first the periodicity of the data was inconsistent (1 second, 2 minutes, 5 minutes, etc.), second in terms of the study scope, the bike sharing rebalancing proposal won't be made every second, so the decision of 20 minutes periodicity was made based on the already mentioned paper. (Regue & Recker, 2014)

c) New features created fall into two types: time related and number of bikes related. The time related ones are variables telling (trip wise) the hour, minute and if it was a weekday or not. The number of bikes related ones, are the number of bicycles present in the station 1, 24 and 168 hours before (1 hour, 1 day and 1 week, respectively).

5. BIKE STATIONS DIMENSIONALITY REDUCTION

5.1. UNSUPERVISED LEARNING VS SUPERVISED LEARNING

Supervised learning and unsupervised learning are two machine learning tasks that are commonly employed for addressing different kinds of problems and have some main differences (Jones, Johnston, & Kruger, 2019)

Table 8 - Differences between unsupervised and supervised learning (Jones, Johnston, & Kruger, 2019)

Unsupervised learning	Supervised learning
No labels provided	Labels provided
Finds structure in unlabeled data	Finds patterns in existing structure
Uses techniques such as clustering or dimensionality reduction	Uses techniques such as regression or classification

5.1.1. Unsupervised learning

Unsupervised learning is used when the target value of each observation is unknown and the input data is the only available information. This type of learning task is commonly used for identifying groups of similar observations: this process is particularly helpful when trying to find the meaning in the data, assign the observations to similar groups or reduce dimensionality.

5.1.2. Supervised learning

Supervised learning is used to solve problems where the target value of each observation is known, so this information can be used to either classify (e.g. predicting if a person will has tendency to be an alcoholic or not based on theirs brain cells activity) or fit a regression (e.g. predicting the price of a personal computer based on how much memory and processing capacity it has).

5.2. UNSUPERVISED LEARNING APPLIED TO TIME SERIES

As outlined in the work of Reddy and Aggarwal, time-series segmentation can have two main formulations, which strongly depend on the problem taken into account (Reddy & Aggarwal, 2013). If the problem consists of finding sets of time series with similar trends, we have a *correlation-based online clustering* problem. On the other hand, if the problem consists of time series with similar shapes, we have a *shape-based off-line clustering* problem.

Correlation-based online clustering: This formulation is commonly used in cases of financial markets domain for identifying groups of stocks that have similar trends or correlated trends.

Shape-based off-line clustering: Conversely to the previous explained formulation, in the shape-based formulation, the time series are evaluated and clustered off-line. This formulation is used in the cases in which the objective is to find time series with similar shapes. The biggest challenge and most

determinate aspect of grouping based on shapes is the definition of how to measure similarity in the shapes. Depending on the problem being solved, there are several good similarity functions, such as Euclidean function or dynamic time wrapping.

Having both formulation types for time-series segmentation and knowing that the goal with the segmentation with this work is to find groups of time series (stations) that have similar behaviors, the most suitable formulation is the shape-based off-line clustering.

5.3. DISTANCE MEASURE

As explained in the work of Reddy and Aggarwal, when the clustering problem is a timeseries problem the similarity concept and how it is measured is very important to have in consideration. In this section we will go through similarity/distance metrics.

As it was mentioned in Section 3.2 and as stated by Izakian and coauthors, "Selecting a distance function to evaluate similarities/dissimilarities of time series has a significant impact on the clustering algorithms and their final results produced by them" (Izakian, Pedrycz, & Jamal, 2015).

Due to this significant impact, some well-known and commonly used distance functions will be explained and considered:

- Euclidean distance: Considering that the datapoints have *n* dimensions, this metric that takes the difference of the coordinates between two data points *p* and *q*, squares it and sums it. The distance between the two points is given by the square of that sum.

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

- Manhathan distance: Considering that the datapoints have *n* dimensions, this metric returns sum of the absolute difference among the *n* coordinates of the data points *p* and *q*. The Manhathan distance between two data *p* and *q* is formalized as follows:

$$d(p,q) = \sum_{i=1}^{n} |p_i - q_i|$$

Besides the distance metrics just mentioned above, there is an important aspect of this study that will affect the choice of the metric, the fact that the type of observations that compose our dataset are time series.

Working with time series brings some concerns with respect to their comparison. For example, two time series can be exactly equal but with a time shift of 2 hours, and if the Euclidean distance is used, the two time-series will appear different to each other, which is not in fact true if we consider the time shift of 2 hours.

To address that important aspect and commonly faced problem, dynamic time wrapping is the best distance to deal with that. This measure "determines an optimal match between two time series by stretching or compressing some segments of the series. As a result, patterns occurring at different time instances of time series are considered as similar". (Izakian, Pedrycz, & Jamal, 2015)

DTW faculty to address this problem has a lot to do with the ability to consider time sifts by comparing each point belonging to time series **a** with any point from time series **b**. (Izakian, Pedrycz, & Jamal, 2015) DTW algorithm pseudo code can be consulted in Table 9.

Table 9 - Calculations of the DTW distance between time series a and b (Izakian, Pedrycz, & Jamal, 2015)

Calculations of the DTW distance between time series a and b

Given:
a = $a1, a2, \dots an$, the first time series with length n
b = $b1, b2, \dots bm$, the second time series with length m
Output:
cost: a matrix of size $n \times m$ containing the cost values $cost_{n,\mathbf{m}}$ is the DTW distance between a and b
path: a matrix of size $n \times m$ containing a warping path
DTW(**a, b**):
Let δbe a distance between coordinates of sequences
$cost_{1,1}$ **=** $\delta(a_1; b_1)$;
$path_{1,1} = (0,0)$;
for i = 2,3,…, n do
$cost_{i,1} = cost_{i-1,1} + \delta(a_i , b_1)$
end
for j = 2,3,…, m do
$cost_{1,j} = cost_{1,j-1} + \delta(a_1 , b_j)$
end
for I = 2,3,…, n do
for j = 2,3,…, m do
$cost_{i,j} = \min (cost_{i-1,j} , cost_{i,j-1} , cost_{i-1,j-1}) + \delta(a_i , b_j)$
$path_{i,j} = \min\_index((i - 1, j),(i , j - 1),(i -1 , j - 1))$;
end
end

5.4. ALGORITHMS FOR UNSUPERVISED LEARNING

In this chapter, we explore the techniques and some parameters used to solve the bicycle stations clustering problem.

5.4.1. Hierarchical Clustering

Hierarchical clustering algorithm has two specifications: The *Agglomerative* and the *divisive.* Both follow different approaches to achieve the clusters formation. The usage of one or the other depends

on the followed approach. Either the clusters are reached by a bottom-up (merging) or by a top-down (splitting) approach.

The interested reader is referred to the works of Reddy and coauthors (Reddy & Aggarwal, 2013) and Kamber and coauthors (Kamber, Han, & Pei, 2011) for a comprehensive overview on these clustering techniques.

Agglomerative hierarchical clustering approach: This approach uses a bottom-up strategy, which means that the algorithm starts by considering as many clusters as observations and iteratively merges the closer two clusters (based on a similarity /distance measure). This process is iterated until the algorithm reaches one cluster containing all the observations or until some stopping criteria (typically the number of clusters obtained) is met.

Divisive hierarchical clustering approach: This approach uses a top-down strategy, which starts by assigning all the observations to one cluster and iteratively divides the clusters into smaller ones. This process is iterated until the algorithm reaches a number of clusters equal to the number of observations or until some stopping criteria (typically the number of clusters obtained) is met.

How **agglomerative hierarchical clustering** measures the closest two clusters to merge are explained below:

Single linkage: The single linkage distance between cluster a and b is the minimum of all distances between points of cluster a and cluster b.

Complete linkage: The complete linkage distance between cluster a and b is the maximum of all distances between points of cluster a and cluster b.

Average linkage: The average linkage distance between cluster a and b is the average of all distances between points of cluster a and cluster b.

5.4.2. Density-based spatial clustering of applications with noise (DBSCAN)

DBSCAN is a clustering algorithm that tries to identify clusters of observations using the fact that the inter-cluster density is higher than the density among the observations that do not belong to the same cluster (Kamber, Han, & Pei, 2011).

DBSCAN receives two parameters:

1) **Eps:** This parameter specifies the maximum distance between itself (point A) and its neighborhood. If the distance between itself and a point B is smaller than eps, point B is considered to be a neighbor of point A.
2) **minPts:** The minimum number of points that constitute a cluster. For example, if minPts is two, means that for a cluster to be formed needs to have at least two points.

DBSCAN categorizes observations into 3 different classes:

1) **Core points**: are the points that have in their neighborhood (eps) at least the minimum number of points a cluster needs to have (minPts), itself included.
2) **Not core points/border points**: are the points that do not comply with the rules of the core points. However, they include inside their neighborhood at least 1 core point.

3) **Noise/Outlier**: these are the points that do not comply with either rule. In other words, they are far away from any other core point. Therefore, they are considered outliers or noise.

In the following sequence of steps is explained how the algorithm performs in order to identify the clusters:

1. The parameters are determined (eps and minPts)
2. A random point A is selected
    a. The neighborhood of point A is calculated using eps.
    b. If there are at least minPts number of points in its neighborhood, point A is categorized as a core point and a cluster is formed with point A and its neighbors. If not, point A is categorized as noise.

Table 10 - DBSCAN algorithm (Chauhan, 2020)

```
DBSCAN (D, Eps, MinPts)
//All objects in D are unvisited
Begin
For all objects in D, select A:
  If A is unvisited:
    Neigh = Calculate A's neighborhood
    N = number of points in Neigh
    If N +1 >= eps:
      Classify A as core point
      Consider all of this points to be part of the same cluster
    Else:
      Classify A as noise
END
```

5.5. MEASURING CLUSTER QUALITY

There comes a point in which several models must be compared so that the best model for the problem under exam is selected. To select the best performer among the existing models, it is necessary to measure their quality and, subsequently, to compare them.

There are several methods to assess clustering quality, that fall in two categories:

1) Extrinsic methods: To use this method, the actual label for each observation must be available. The extrinsic methods compare the labels attributed by the clustering model with the actual labels and measure how accurate the classification was.
2) Intrinsic methods: There is not the need to have the actual label for each observation to use intrinsic methods. This category of methods measures how well the clusters are separated.

In the problem considered in this work, the actual label for each observation is not known. Thus, only the intrinsic methods were used to address this problem.

5.5.1. Intrinsic evaluation methods

The performance of the clustering algorithms and its parameters were addressed using a well-know and commonly used evaluation measure:

1) Silhouette coefficient: This measure quantifies how cohesive the cluster is when compared against other clusters (how separated the cluster is from the others). Basically, the measure can quantify how similar each observation is to its own cluster.

$$s(i) = (b(i) - a(i))/Max\{\ a(i)\ , b(i)\ \}\ \ , -1\ \leq s(i) \leq 1$$

Take an observation *i* from the dataset and let us call A to the cluster it belongs to and C to another cluster. In that case, we have the following:

- $a(i)$: The average dissimilarity of *i* to all objects of A
- $b(i)$: The minimum distance of *i* to all observations of cluster C, assuming that cluster C is not the same as cluster A and it is the closest cluster to cluster A.
- $Max\{\ a(i), b(i)\}$: The maximum values between a(i) and b(i)

Regarding the values that the silhouette coefficient can have:

- Close to 1: It can be said that the observation is well clustered, meaning that the dissimilarity within the cluster it belongs to is very small when compared to the minimum distance to an observation from another cluster.
- Closet to 0: It can be interpreted as that the clusters A and B are really similar to each other, and as for that, it is not clear if the observation *i* should belong to cluster A or cluster B.
- Close to -1: the dissimilarity within its cluster is higher than the distance to the closest observation belonging to another cluster which can be interpreted as the observation was misclassified as it is more similar to the closest cluster than to its own.

The choice of the evaluation metric was done in pair with the type of problem being addressed, as (Reddy & Aggarwal, 2013) said “Clustering of time-series data, like clustering for all types of data, has the goal of producing clusters with high intracluster similarity and low intercluster similarity”. Thus, considering the fact that the silhouette coefficient takes into account both inter-cluster and intra-cluster similarity in its formula, the evaluation metric used was the silhouette coefficient.

6. RESULTS, CONCLUSION AND FUTURE WORK

Research question: Can the use of clustering techniques find groups with similar behavior, thus reduce the problem of dimensionality for future applications such has predicting demand?

In order to answer this question, it is important to find a stable period of time within the dataset, in other words, a period where no station was neither introduced in the network nor stopped its operation in within the BSS network. A stable period is required because the changes effect in the BSS network is not part of the scope and research question of this work.

Research work of Rudloff and Lackner (Rudloff & Lackner, 2014) revealed that meteorological phenomena is correlated with the usage of a BSS, therefore weather and daylight data was collected (more details in Section Data integration4.2), due to project limitation such as python library, this data was not used, though it would be a feature to consider in future work.

The study focused on analyzing the patterns of one variable throughout time, the bicycle variation in the Gira stations. Bike variation represents the difference between arrivals and departures at each station across the time dimension.

This chapter is dedicated to show the results of the study. As it was mentioned in Section 5.5.1, the metric used to compare models was the silhouette coefficient. To ensure that the results were not biased by the choice of the parameters of the distance/similarity measure (DTW), a reference distance matrix was created, and all the silhouette score values were computed based on that reference matrix. By observing Figure 1 is clear that as the number of clusters grow, the silhouette score decreases, therefore by using the score as a decision maker, the number of clusters selected to address the research question would be 2.

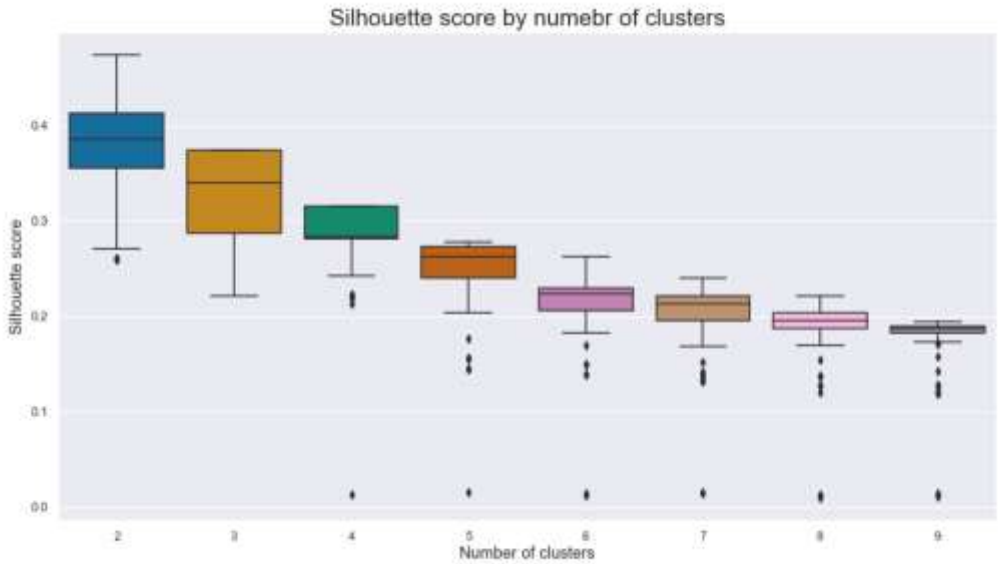

Figure 1 - Silhouette score by number of clusters

Besides the number of clusters, the effect of other parameterizations in the silhouette score were tested, such as the window size of DTW and the periodicity of the data (20 minutes, 40 minutes or 60 minutes). The previously mentioned parameterizations were tested with two algorithms, DBSCAN and Agglomerative Clustering using algorithm-specific features, such as the linkage in Agglomerative Clustering casa and the eps in the DBSCAN case. The top 5 results can be observed in:

Table 11 - Time series clustering results

Algorithm	Time periodicity	Distance metric	Window size	Linkage	Nr of clusters	Silhouette Score
AgglomerativeClustering	60 min	DTW	0	Single	2	0.583
AgglomerativeClustering	60 min	DTW	0	Complete	2	0.583
AgglomerativeClustering	60 min	DTW	0	Average	2	0.583
DBSCAN (eps=77)	60 min	DTW	50		2	0.532
DBSCAN (eps=79)	60 min	DTW	50		2	0.532

Since several combinations of window size and number of clusters gave the same silhouette coefficient, the curse of choice between the combinations in Table 11Table 11 was led by minimizing the complexity of the algorithms, so the features selected with the best score in this study would be:

- DTW window size equal to zero.
- Number of clusters equal to 2.
- Agglomerative Hierarchical clustering with linkage to single and with DTW as distances between the observations.
- Time periodicity of 60 minutes.

Due to the best results being accomplished by two clusters and in order to present a descriptive analysis of each cluster behavior, a deeper analysis was conducted. During the course of that analysis it was observed that there would always be one cluster with all observations but one and the other cluster with the remaining one. This behavior leads us to one hypothesis: the ideal number of clusters would be one and the answer to the research question of this study would be that in this case is not possible to reduce stations dimensionality.

Ergo this theory could be proven by performing two tests:

- **Test 1:** Observing how dissimilar the observations are. The conclusion was that all the observations were very similar with each other.

- **Test 2:** Comparing the observation that composes a cluster against one other observation from the remaining cluster and see if they are close to each other. The results of this test, seen in Figure 2, allowed to conclude that in fact the observation that would compose a cluster by itself was very close to a randomly selected observation from the other cluster.

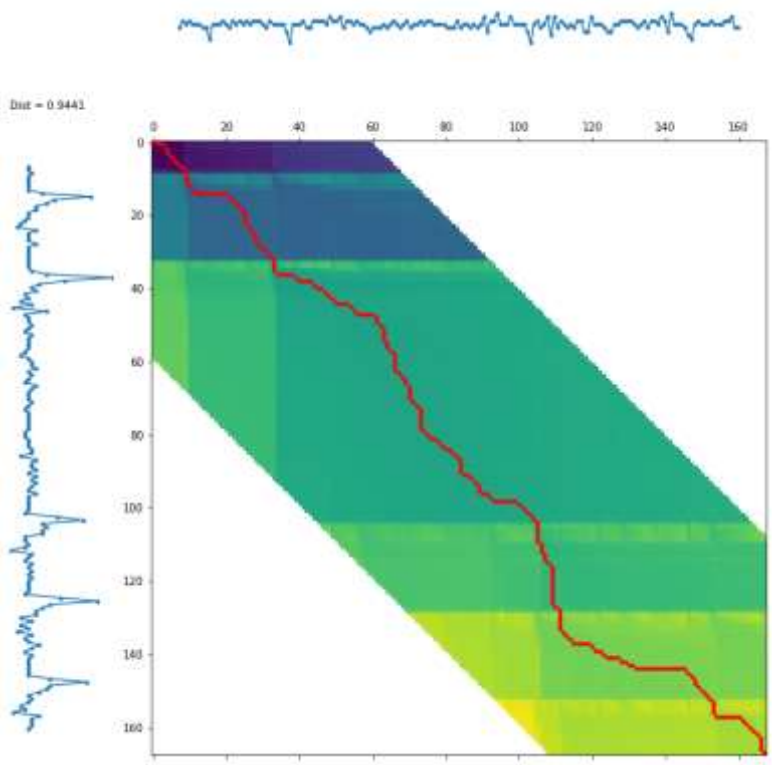

Figure 2 - Matrix with the shortest warping path

Having in consideration all the study developed and the hypothesis created in order to understand if there should be two clusters or just one, the data used for this study will not benefit from a clustering analysis prior to the demand prediction part because the observations are too similar.

Regarding future work due to the similarity of the observaions, a bigger variety of variables could be taken in consideration. Algorithms could be added to the equation, specially algorithms with a geographical dimension.

7. BIBLIOGRAPHY

Albiński, S., Fontaine, P., & Minner, S. (2018). Performance analysis of a hybrid bike sharing system: A service-level-based approach under censored demand observations. *Transportation Research Part E: Logistics and Transportation Review, 116*, 59-69.

Caggiani, L., & Ottomanelli, M. (2013). A Dynamic Simulation based Model for Optimal Fleet Repositioning in Bike-sharing Systems. *Procedia - Social and Behavioral Sciences, 87*, 203-210.

Chauhan, N. S. (04 de 2020). *DBSCAN Clustering Algorithm in Machine Learning*. Obtido de KDnuggets: https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html

Dell'Amico, M., Iori, M., Novellani, S., & Subramanian, A. (2018). The Bike sharing Rebalancing Problem with Stochastic Demands. *Transportation Research Part B: Methodological, 118*, 362-380.

DeMaio, P., & DesJardins, Z. (s.d.). *The Bike-Sharing World Map*. Obtido de Metrobike: http://www.metrobike.net/the-bike-sharing-world-map/

Elhenawy, M., & Rakha, H. (2017). A heuristic for rebalancing bike sharing systems based on a deferred acceptance algorithm. *5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2017 - Proceedings*, 188-193.

Feng, S., Chen, H., Du, C., Li, J., & Jing, N. (2018). A hierarchical demand prediction method with station clustering for bike sharing system. *Proceedings - 2018 IEEE 3rd International Conference on Data Science in Cyberspace, DSC 2018*, 829-836.

Feng, Y., Affonso, R. C., & Zolghadri, M. (2017). Analysis of bike sharing system by clustering: the Vélib' case. *IFAC-PapersOnLine, 50*, 12422-12427.

Forma, I. A., Raviv, T., & Tzur, M. (2015). A 3-step math heuristic for the static repositioning problem in bike-sharing systems. *Transportation Research Part B: Methodological, 71*, 230-247.

Froehlich, J., Neumann, J., & Oliver, N. (2009). Sensing and predicting the pulse of the city through shared bicycling. *IJCAI International Joint Conference on Artificial Intelligence*, 1420-1426.

Hua, M., Chen, X., Zheng, S., Cheng, L., & Chen, J. (2020). Estimating the parking demand of free-floating bike sharing: A journey-data-based study of Nanjing, China. *Journal of Cleaner Production, 244*.

Izakian, H., Pedrycz, W., & Jamal, I. (2015). Fuzzy clustering of time series data using dynamic time warping distance. *Engineering Applications of Artificial Intelligence*, 235-244.

Jones, A., Johnston, B., & Kruger, C. (2019). *Applied Unsupervised Learning with Python.* Packt Publishing.

Kamber, M., Han, J., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd Edition ed.). Morgan Kaufmann.

Liu, J., Sun, L., Chen, W., & Xiong, H. (2016). Rebalancing bike sharing systems: A multi-source data smart optimization. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1005-1014.

Liu, Y., Szeto, W. Y., & Ho, S. C. (2018). A static free-floating bike repositioning problem with multiple heterogeneous vehicles, multiple depots, and multiple visits. *Transportation Research Part C: Emerging Technologies, 92*, 208-242.

Reddy, C. K., & Aggarwal, C. C. (2013). *Data Clustering.* Champman and Hall/CRC.

Regue, R., & Recker, W. (2014). Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem. *Transportation Research Part E: Logistics and Transportation Review, 72*, 192-209.

Rudloff, C., & Lackner, B. (2014). Modeling Demand for Bikesharing Systems. *Transportation Research Record: Journal of the Transportation Research Board, 2430*, 1-11.

Schuijbroek, J., Hampshire, R. C., & van Hoeve, W. J. (2017). Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research, 257*, 992-1004.

Shaheen, S. A. (2012). Introduction: Shared-Use Vehicle Services for Sustainable Transportation: Carsharing, Bikesharing, and Personal Vehicle Sharing across the Globe. *International Journal of Sustainable Transportation, 7*, 1-4.

Shui, C. S., & Szeto, W. Y. (2018). Dynamic green bike repositioning problem – A hybrid rolling horizon artificial bee colony algorithm approach. *Transportation Research Part D: Transport and Environment, 60*, 119-136.

Vogel, P., Greiser, T., & Mattfeld, D. C. (2011). Understanding bike-sharing systems using Data Mining: Exploring activity patterns. *Procedia - Social and Behavioral Sciences, 20*, 514-523.

Witten, I. H., Pal, C. J., Hall, M. A., & Frank, E. (2016). *Data Mining.* Morgan Kaufmann.

Xu, H., Ying, J., Lin, F., & Yuan, Y. (2013). Station segmentation with an improved K-means algorithm for Hangzhou Public Bicycle System. *Journal of Software, 8*, 2289-2296.

Yıldırım, S. (22 de 04 de 2020). *DBSCAN Clustering — Explained*. Obtido de Towards data science: https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556

Zacharias Voulgaris, P. (2017). *Data Science: Mindset, Methodologies, and Misconceptions.* Technics Publications.

Zhao, X., Hu, C., Liu, Z., & Meng, Y. (2019). Weighted dynamic time warping for grid-based travel-demand-pattern clustering: Case study of Beijing bicycle-sharing system. *ISPRS International Journal of Geo-Information*.

Zhou, Y., Wang, L., Zhong, R., & Tan, Y. (2018). A Markov Chain Based Demand Prediction Model for Stations in Bike Sharing Systems. *Mathematical Problems in Engineering*.