Clustering Techniques for On-Demand Transport Data: A Case Study

Carlos Afonso[1]
[1]Department of Informatics and Systems Engineering
ISEC, Polytechnic Institute of Coimbra
Coimbra, Portugal
e-mail: a21240004@isec.pt

Ana Alves[1,2]
Department of Informatics and Systems Engineering
[1]ISEC, Polytechnic Institute of Coimbra
[2]CISUC, University of Coimbra
e-mail: ana@dei.uc.pt

Abstract— **The on-demand transportation request requires a quick and efficient response to satisfy customers and also make the system a viable option. Clustering techniques were used to group transport requests, i.e., the starting points of vehicles that have been requested to optimize the service with the benefit of reducing the number of vehicles needed and, consequently reduce the amount of pollution produced. The objective is to compare the two main clustering techniques from two distinct categories: partitioned and density-based to evaluate which one is best suited for defining start zones. The quality of the generated clusters is defined by calculating the silhouette related to the generated clusters. Using previous references, the two clustering methods were compared based on the desired characteristics. The analysis demonstrates that DBSCAN is best suited for the problem and is then applied over a sample dataset. The manner in which the DBSCAN algorithm can generate random shapes, which fit well into the geographic distribution of points and how the number of necessary clusters do not need to be defined in advance makes it the ideal choice for defining starting zones.**

Keywords- On-Demand Transport; Transport Requests; Partition-based Clustering; Density-based Clustering; K-Means; DBSCAN

I. INTRODUCTION

Clustering techniques are essential since they allow grouping something, whether they are objects or people, according to their degree of similarity. In this way, clustering will play a key role in this study focused on smart mobility, and the objective is to optimize the collection points of people in a city or on a route to it. It involves aggregating transport requests, grouping people together and simultaneously ensuring route optimization [1]. This article will compare two of the various clustering techniques that exist, more specifically K-Means and Density-Based Spatial Clustering of Applications with Noise (DBSCAN**)**, since they are part of the core of the most popular clustering techniques [2]. Clustering techniques are essential for problems such as people search. Throughout this paper, several key points of each of the mentioned techniques will be described, namely, a description, advantages and limitations, a comparison between the two and a case study that will use the best technique in comparison.

This paper is structured as follows: in section 2, several study scenarios are covered with the clustering techniques that will be compared; in section 3, where various techniques for encouraging the grouping of persons are addressed; in section 4 is made a brief description of Clustering and K-Means and DBSCAN are analyzed and compared. In addition, the silhouette coefficient is also addressed; in section 5, a comparison is made between the two clustering techniques (K-Means and DBSCAN); in section 6, the case study carried out with the application of the best clustering technique resulting from the comparison made in the previous section is described. Finally, in section 7, the conclusions and future work are presented, followed by the thanks and their references.

II. STATE OF THE ART

The importance of clustering applied to mobility has been demonstrated [3] exploring travel patterns and target transit passengers only using smart card data. These smart cards have individual transactions, that is, per user and each working day was used to build travel itineraries. In this research work, DBSCAN and K-Means were used as clustering algorithms. For this, it was necessary to resort to DBSCAN, which was used to group the last landing stops made by users. In addition, the same algorithm also groups the starting stops and the time a passenger normally boards a particular vehicle. Finally, through K-Means, users were classified and divided into several groups, such as “transit passengers”, “regular passengers”, “habitual time passengers” and “irregular passengers”.

Another research work used data-mining tools and presented several measures related to the variability of travel behavior by public transport users. These analyses were carried out using, once again, the smart card and were collected over a period of ten months allowing to understand the difference in terms of boarding per day and new frequent stops with the travel days on the public transport network. For this, the authors used data-mining techniques to build clusters of days that present similar time patterns of boarding on the public transport network in order to understand whether passengers have regular travel behaviors and whether the days differ significantly. The experiments have shown that the behavior of habitual transport users evolves over time, both in terms of frequent transport stops and in relation to boarding

 ISBN: 978-1-61208-798-6

hours, leading to the conclusion that the change in behaviors varies according to the various types of existing users [4].

Another study focuses on taxi transport in Taiwan [5], helping taxi drivers to find high-density locations. This was due to the fact that drivers generally do not know where the passengers are, making them spend a lot of time driving unoccupied vehicles, thus revealing a huge loss to the business. Therefore, it was essential to look for a solution for taxi drivers to find potential customers and the locations of future customers. Based on a given reference position, weather condition, current weather, order history, hotspots have been calculated that can be provided and recommended derived from the use of data-mining techniques. In this case, DBSCAN served to group the coordinates of customer locations according to the spatial distance, leading to each identified cluster, the corresponding roads properly associated. With the result of this analysis, taxi drivers can help their strategies and decided where to go in order to pick up more passengers.

Another research paper published in GeoInfo [6], came up with the development of an application based on the DBSCAN clustering algorithm with the aim of reducing the daily loss of time in moving a large number of people to a common place. Clusters are created based on various attributes, such as the departure time of each person in his home, the final destination and their departure and arrival locations. The people who make up a given cluster are transported in an allocated vehicle according to the size of the same. A case study was then conducted with a certain group of people moving to campus II of the Federal Center for Technological Education of Minas Gerais using a traffic simulator to measure the individual and global time people need to move. Using the concepts of clustering on the latitude, longitude and time of movement of individuals, data were collected and processed in order to group people with similar movement time and location of origin and destination, making it possible to find routes that can transport these groups of people effectively. In order to find a solution that would improve people's daily movement time, the problem was divided into two parts. First, people were brought together in groups, according to the DBSCAN algorithm. For each cluster, a centroid was calculated and defined as a starting point for people in that particular cluster. So, in the first part of the route, each person walks to that central point, spending a certain initial time of the walk, having established that the time of the walk had to be less than twenty minutes. The distance of each individual to the center of the cluster was calculated using the haversine function and the estimated movement time to the centroid was defined as the average walking time of 4.82 kilometers / hour for each person. Subsequently, in the second part of the problem, after all people walked to the centroid of their respective cluster, the route would be established from one or more vehicles. The type of vehicle was chosen taking into account the number of people allocated to each cluster, with up to 5 people being displaced by car, between 6 to 15 people would be in a van, between 16 to 40 would be by bus or more than 40, people would be distributed among several vehicles, depending on their total number. After choosing the most suitable vehicle, the rest of the route is carried out without any kind of stop to the final destination and the movement time of each vehicle to the final destination was calculated using data from Google Maps, thus obtaining the total time that it would take [7].

Another study proposed an on-demand transportation system using clustering, in the book “Data-Driven Solutions to Transportation Problems” [8] had as main objective to help taxi drivers to collect and transport passengers from their departure location to their intended destination location and simultaneously walk the road in order to find the next customer. Data was collected from 1100 drivers in Harbin, a city located in China. This data contains various information, such as: **time,** which indicates the date and time when the records were recorded; **latitude and longitude** that specify the location of the taxi vehicle; -**speed**, which indicates the average speed of the vehicle; **orientation,** which represents the direction in degrees; **travel status,** which is a Boolean value that indicates whether the taxi is occupied by passengers. This study had as main objective to help taxi drivers to collect and transport passengers, from their departure location to their intended destination location and simultaneously walk the road in order to find the next customer. It was then relevant to understand which area of the city can attract a greater number of people and which is the area of the city. Thus, they decided to use DBSCAN as a clustering technique in order to group passengers' embarkation and disembarkation places. Their use has brought several benefits, such as the fact that they are able to classify locations in a cluster with high density, are able to find specific locations on the road network for each cluster and are able to deal well with noise points. Although DBSCAN was already used to group other types of things apart from transport requests, it was found to be the most used for the on-demand transport domain.

III. GROUPING OF PEOPLE

Before moving on to the description of clustering and some of its techniques, it is important to run a little context to understand why this unsupervised machine learning technique will be used. One of the causes of great mobility difficulties, especially in the city, is the volume of private vehicles circulating daily. A high percentage of them travel with only one passenger, which represents reduced efficiency. This inefficient use has a major impact on the flow of traffic routes, difficulties related to parking problems and increased pollution.

Grouping people into vehicles that will make the same trip or similar routes represents one of the ways to reduce the impact mentioned. If only one person is in the vehicle, the cost will be much higher, leading to greater congestion on the roads, whereas there will be a greater number of vehicles on them and a greater impact on environmental pollution in society. In addition, there is a financial impact resulting from the reduction in the cost of travel per element.

There are financial incentives and methods of road organizations to group as many people in a given vehicle as ***High-occupancy vehicle lanes, High-Occupancy toll lanes*** and ***Slugging lines.*** The use of a car has a relatively high cost,

maintenance, consumption, wear at the pneumatic level, among others.

The ***High-occupancy Vehicle lane (HoV)*** method [9] encourages drivers to bring together as many people as possible, offering specific transit routes. On roads with heavy vehicle traffic with a single occupant, this special route facilitates circulation, saving time for those sharing trips. In Portugal, there are traffic corridors for public transport (*buses* and *taxis)*, but in this case, the HoV allow its use by private vehicles.

The ***High-occupancy toll lane*** method [10] encourages users to gather as many people in a given vehicle, reducing the toll price. A smaller number of people in a given vehicle leads to an increase in toll prices.

Another method is ***slugging*** [11], which is an organized system in which people travel to a city to pick up other passengers, who may even be strangers, and share the cost of the trip among them.

IV. CLUSTERING

Clustering or **Data Grouping Analysis** is the set of data-mining techniques that aim to create automatic data groupings according to their degree of similarity. The similarity criterion is part of the problem definition and is dependent on the algorithm that will be used. Each collection resulting from the process is given the name of the group or grouping (cluster) [12].

This data analysis technique is increasingly being used to classify elements into groups, in order to understand which elements within a dataset are similar (are in a cluster*)* and which elements of the set are distinct from each other (they are in distinct clusters*)* [13].

The data grouping algorithms that will be described below are the most popular within the categories of Partitioned (K-Means) and Density-Based (DBSCAN). According to the state of the art, K-Means and DBSCAN are the two techniques that are most used to group people.

A. K-Means

The K-Means algorithm, also known as Hard *C-means* [14], organizes the elements of a dataset in a given number of *clusters* defined by the user. This data should be organized in the different clusters according to the similarity between them. The K-means algorithm organizes the data in k clusters and determines the organization of the clusters through an iterative process. The iterative aspect of the algorithm exists to look for the best result with each iteration with some mechanisms to avoid ending in local optimums.

The operation of the K-means algorithm can be divided into four steps. In the first step, the centers of each cluster are initialized. Typically, these centers are initialized by choosing *k* points randomly in the dataset, with k being the number of clusters defined at the beginning by the user.

After cluster centers have been initialized, each of the dataset elements is assigned to each of these clusters. This assignment is performed based on the shortest distance between an element and the centers of the defined clusters, that is, the cluster to which the element is closest.

In the third step, the value of a cost function is calculated using the equation (1).

$$WCSS_1 := \sum_{c_i \in C} \sum_{j=1 \ldots d} \sum_{x,y \in c_i} (x_{ij} - y_{ij})^2 \qquad (1)$$

The cost **J** of the clustering iteration consists is the sum of the costs of each cluster **C** that consist of the sum of the distances of each element $\mathbf{X_k}$ to the center of the cluster $\mathbf{C_i}$ to which the element was assigned to. In this case, euclidean distance is used although it is possible to use other types of distances between vectors. After calculating the cost function, it is possible to see whether there was an improvement in the result in relation to the value calculated in the previous iteration. If the value has improved, the algorithm moves on to the next iteration. If this value has increased the algorithm ends its execution and returns the results obtained in the previous iteration.

In case the algorithm does not finish its execution in the third step, there is a fourth and final step consisting of the renewal of the centers of all clusters using equation (2). This equation calculates the new centers $\mathbf{C_i}$ of each cluster by averaging between the elements $\boldsymbol{X_k}$ of that cluster.

$$c_i = \frac{1}{|G_i|} \sum_{k, x_k \in G_i} x_k \qquad (2)$$

Once the new centers have been calculated, the algorithm returns to the second step and continues to iterate until the stop criterion is met. Many implementations of this algorithm also present a variant that consists of its repetition, choosing at the end the best result among all repetitions, that is, the one that has a lower cost function value. This variant was introduced due to two factors. The first because the quality of the *K-means* algorithm is dependent on the initialization of cluster centers, because if the startup was always performed with the same centers the result obtained for a dataset would always be the same. Another factor is the possibility of the algorithm reaching quite easily local optimums. Thus repeating the algorithm, a certain number of times and using different centers at the beginning of the algorithm, the probability of achieving better results increases [15].

B. DBSCAN

The **DBSCAN** [16] algorithm is based on the density at which clusters grow according to a given density [17] threshold, where the number of clusters is decided depending on the data that is provided. Since it is a density-based clustering algorithm, some points in the data may not belong to any cluster. Again, this is quite different from the algorithm that was mentioned earlier (K-means), where all points of the data are considered and necessarily belong to some cluster. This can be explained with the help of two parameters ***epsilon*** and ***min_points*** that are used in the algorithm. Two points are considered neighbors if the distance between them is below the ***epsilon*** threshold. The minimum number of neighbors that a given point must have to be classified as a main point is defined by ***min_points***. This type of algorithm is widely used at the level of population density and there are some aspects that are relevant and worth mentioning:

- DBSCAN is a flexible algorithm, since it is dynamic relative to the data;
- The parameters required to execute the algorithm can be obtained from the data themselves, using **adaptive DBSCAN;**
- It provides a more intuitive grouping as it is based on density that leaves out points that do not belong anywhere (outliers).
- DBSCAN has $O(n^2)$ Complexity [18] where *n* is the number of data points, and so it is much faster when compared to traditional clustering techniques such as K-means that has a complexity of O(IKn), where *n* is the number of data points, *k* is the number of clusters and *I* is the number of iterations [19].

C. Silhouette Coefficient

The silhouette coefficient is a metric used to calculate the quality of a clustering technique. This will be used to evaluate the quality of the DBSCAN algorithm that uses the average distance between points of the same cluster and the average distance between nearby clusters. The value of the silhouette varies between -1 and 1, where -1 is the worst possible score and 1 the best. The value of 1 represents high density clusters (internal points very close to each other) and clusters far between each other. A value of 0 suggests overlapping clusters and a value of -1 means that clusters are assigned in the wrong way [20].

V. COMPARISON OF THE TWO CLUSTERING TECHNIQUES

In [21], the author made a comparison of several clustering techniques where the metrics analyzed were as follows: **Feature A-** ability to identify clusters with random shapes; **Feature B-** ability to identify clusters in datasets with high data volume; **Feature C-** good performance in obtaining results, mainly also in datasets with considerable volume of data; **Feature D-** ability to deal with noise and achieve that its presence has no impact on the results obtained; **Feature E-** parameterization/initial configuration of the algorithm (the non-obligation to indicate the number of final clusters that will have to be generated and estimation of the initial parameters); **Feature F:** handle numeric values.

For the current case under study we do not know the number of *clusters* that should be generated, which makes Characteristic E relevant. The ability of the algorithm to be able to identify *clusters* with multiple random shapes is also a characteristic that is intended, given the fact that we do not know what type of form the *clusters* will have though we predict that the adopted forms will correlate with geographic planning. It will also be interesting to analyze whether the algorithm to choose can handle noise and whether it does not affect the results. The ability to handle numeric values is key, as the *clustering* of transport requests will be done and a grouping of a radius will have to be done. Another metric that will also be important to analyze is a good performance in achieving results and the ability to identify *clusters,* as it will use datasets with a large volume of data.

TABLE I. COMPARISON BETWEEN THE TWO CLUSTERING TECHNIQUES [21]

Algorithm	A	B	C	D	E
K-Means	x	x	x	x	✔
DBSCAN	✔	✔	✔	✔	✔

Through the Table I, we can perceive that *K-means,* which belongs to partitioned methods, is sensitive to noise which influences the final results and it cannot identify clusters with random shapes, being only able to identify circular *clusters.* One of the main limitations of the algorithm mentioned above is that it cannot find the best partitions for clusters with different sizes and with different densities, and for these reasons and those mentioned above, a decision was made to not use it. Therefore, the algorithm to be used for this paper will be DBSCAN, since it meets all the requirements necessary, namely the fact that it is able to deal with *clusters* with different sizes and shape and the fact that it is needed to group *n* transport requests around a geographic coordinate, which are based on numerical values.

To our knowledge, DBSCAN has not yet been used to group transport orders, although it has already been used to group other very similar types of data, as already mentioned in the state of the art.

VI. CASE OF STUDY

In view of the result of the comparison made earlier, it was concluded that the best technique to use would be DBSCAN. A dataset [22] consisting of a set of taxi trips to the city of Chicago was used. This dataset contains several columns that were essential to the study, such as the "*Pickup Centroid Latitude*" and the "*Pickup Centroid Longitude*" that refer to the location of a taxi's starting point, as well as the "*Dropoff Centroid Latitude*" and the "*Dropoff Centroid" Longitude*" that refer to the location of a destination point of a taxi.

Thus, to perform the case study, a Jupiter notebook was produced importing the ***scikit-learn*** library. First, for the case of starting points, the dataset was loaded containing 278000 points. A sample with first 1000 points of the full dataset was used for the remaining steps to simplify visualization. The data relating to the "Pickup Centroid Longitude" and "Pickup Centroid Latitude" were extracted, having drawn them on a map taking into account the "corners" of the map so that the dots fit inside.

DBSCAN execution requires *epsilon* and *min_points* to be defined. The value for *epsilon* is obtained by executing the *nearest neighbors algorithm* with the dataset. The value for *epsilon* is the *y* coordinate of the point on the generated graph that presents the greatest inflection. The choice for the *min_points* value was made by comparing the results of the DBSCAN algorithm and choosing the value that produce the most satisfying results. Based on the data from the pickups, the Nearest Neighbors algorithm was executed to obtain an epsilon value to use in the DBSCAN algorithm. Then, the DBSCAN algorithm and the removal of points with noise were executed. Thus, as optimal values for *epsilon,* the value of 0.0000001 was obtained and for *min_points* the value of 3

was obtained. DBSCAN's results were 56 clusters with a silhouette coefficient of 0.907. This coefficient corresponds to a metric for measuring DBSCAN performance and is calculated using the average intracluster mean distance [15], and a value close to 1 is desired that indicates a high density of the points within each cluster and far from the remaining clusters. The case of study is stored on GitHub in a public way and can be accessed through the following url: https://github.com/pedroafonsoo/clustering_case_study_industrial_seminars/blob/master/dbscan_case_study_dbscan.ipynb .

The image in Figure 1 shows the locations of the mapped dataset, the image in Figure 2 demonstrates the same but colored locations representing the clusters and the Figure 3 demonstrates the representation of the Figure 2 but in detail with the largest area populated by points. The images contain two axes: x- longitude; y- latitude. Each of the clusters created, on average, have 18 points, and the cluster that presented the highest number of points was the cluster with label 3 with 109 points. To limit the number of points per cluster, it should be necessary to apply in a second phase a second clustering technique with this capacity as it will be presented in the future work.

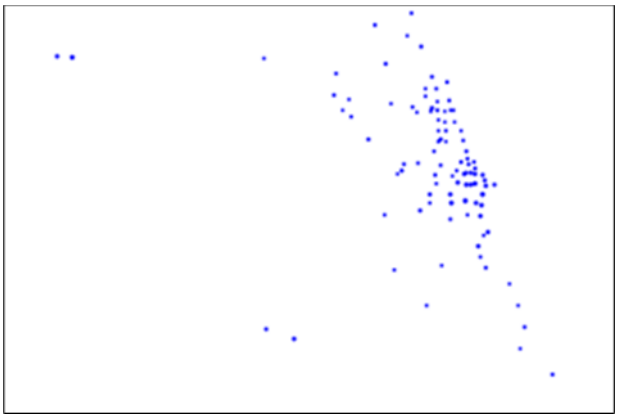

Figure 1. Spatial representation of Pickup points.

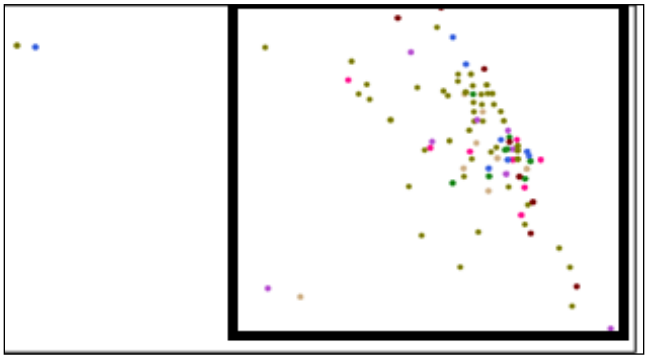

Figure 2. Spatial representation of Pickup points and clusters to which they belong (Each cluster is represented by a color which could be repeated across the 56 clusters generated)

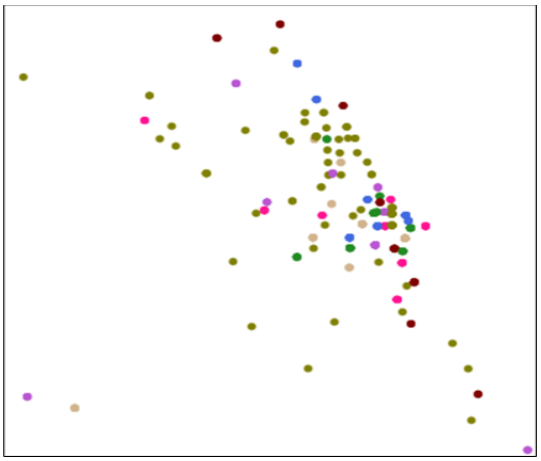

Figure 3. Representation of the Figure 2 but in detail with the largest area populated by points

VII. CONCLUSION AND FUTURE WORK

For the case of on demand transport study, the DBSCAN algorithm is the most appropriate compared to K-means, since it complies with all evaluated characteristics. By applying the algorithm with a real data subset, we obtained the set of associated clusters that define pickup points, with a strong silhouette value, indicating quality in the result. The generated clusters confirmed that there are a set of common pickup zones that can be explored in future steps of the study, which is publicly available. As a future work, it may be to combine the DBSCAN clustering technique with Constrained K-Means [23]. Constrained K-Means implementation modifies the cluster assignment step by formulating it as a Minimum Cost Flow (MCF) linear network optimisation problem. This is then solved using a cost-scaling push-relabel algorithm and uses Google's Operation Research tools's SimpleMinCostFlow which is a fast C++ implementation. With the application of a second phase of Constrained K-Means it will be possible to restrict the capacity of the minimum and maximum number of points for each cluster and at the same time guarantee the optimization of the distance between the points. In this way it will be possible to group a minimum and a maximum number of passengers, depending on the capacity of a given vehicle. In this way, Constrained K-means would be applied to each cluster resulting from DBSCAN.

ACKNOWLEDGMENT

I thank Ubiwhere, primarily to André Duarte for the opportunity, help and suggestions he gave for the preparation of this paper. I thank my advisor for ISEC, teacher Ana Alves, for her professionalism, both as a teacher – where in classes she excelled in pedagogical and didactic excellence- as a counsellor, and exemplary follow-up, which she has been performing during the development of the internship and finally to teacher João Cunha, not only for everything he taught in Industrial Seminar classes as well as the preparation he demanded with several presentations and debates throughout the semester.

 ISBN: 978-1-61208-798-6

REFERENCES

[1] Cordeau, J. F., Laporte, G., Potvin, J. Y., & Savelsbergh, M. W. (2007). Transportation on demand. *Handbooks in operations research and management science*, *14*, 429-466.

[2] Seif, G. Towards Data Science, 5 February 2018. [Online]. Available: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68. [Accessed November 2019].

[3] Bhaskar, A., & Chung, E. (2014). Passenger segmentation using smart card data. *IEEE Transactions on intelligent transportation systems*, *16*(3), 1537-1548.

[4] Morency, C., Trépanier, M., & Agard, B. (2006, September). Analysing the variability of transit users behaviour with smart card data. In *2006 IEEE Intelligent Transportation Systems Conference* (pp. 44-49). IEEE.

[5] Chang, H. W., Tai, Y. C., Chen, H. W., Hsu, J. Y. J., & Kuo, C. P. (2008). iTaxi: Context-aware taxi demand hotspots prediction using ontology and data mining approaches. *Proc. of TAAI*.

[6] "GEOINFO," [Online]. Available: http://www.geoinfo.info/geoinfo2020/. [Accessed July 2020].

[7] Andrade, T. C., de Arruda Pereira, M., & Wanner, E. F. (2014). Development of an Application Using a Clustering Algorithm for Definition of Collective Transportation Routes and Times. In *GeoInfo* (pp. 13-24).

[8] Tang, J. (2019). Urban Travel Mobility Exploring With Large-Scale Trajectory Data. In *Data-Driven Solutions to Transportation Problems* (pp. 137-174). Elsevier.

[9] High Occupancy Vehicle (HOV) Lanes, [Online]. Available: http://www.its.leeds.ac.uk/projects/konsult/private/level2/instruments/instrument029/l2_029summ.htm. [Accessed April 2020]. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[10] High-Occupancy Toll Lanes (Partial Facility Princing), 2 March 2020. [Online]. Available: https://ops.fhwa.dot.gov/congestionpricing/strategies/involving_tolls/hot_lanes.htm. [Accessed April 2020].

[11] About Slugging, 31 January 2020. [Online]. Available: http://slug-lines.com/Slugging/About_slugging.asp. [Accessed April 2020].

[12] Clustering in machine learning, [Online]. Available: https://www.geeksforgeeks.org/clustering-in-machine-learning/. [Accessed April 2020].

[13] Oliveira, B. *O que é a análise de cluster?*, 2 October 2019. [Online]. Available: https://operdata.com.br/blog/analise-de-cluster/. [Accessed December 2019].

[14] Goktepe, A. B., Altun, S., & Sezer, A. (2005). Soil clustering by fuzzy c-means algorithm. *Advances in Engineering Software*, *36*(10), 691-698.

[15] Oliveira, J. *Classificação de Literatura Biomédica*, December 2014. [Online]. Available: http://files.isec.pt/DOCUMENTOS/SERVICOS/BIBLIO/Tees/Tese_Mest_Joao-Santos-Oliveira.pdf. [Accessed December 2019].

[16] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, No. 34, pp. 226-231).

[17] Castro, M. *Agrupamento-Clustering*, July 2003. [Online]. Available: http://www.dei.isep.ipp.pt/~paf/proj/Julho2003/Clustering.pdf. [Accessed December 2019].

[18] Agrawal, S. Machine Learning - DBSCAN, 29 May 2013. [Online]. Available: https://algorithmicthoughts.wordpress.com/2013/05/29/machine-learning-dbscan/. [Accessed December 2019].

[19] Clustering Algorithms: K-means, 2006. [Online]. Available: https://www.cs.princeton.edu/courses/archive/spring08/cos435/Class_notes/clustering2_toPost.pdf. [Accessed December 2019].

[20] Lutins, E. DBSCAN: What is it? When to Use it? How to use it., 6 September 2017. [Online]. Available: https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818. [Accessed December 2019].

[21] Simões, J. *Repositório Comum- Análise de dados e Machine Learning na Mobilidade Urbana*, April 2019. [Online]. Available: https://comum.rcaap.pt/bitstream/10400.26/29858/1/Joao-Pedro-Fernandes-simoes.pdf. [Accessed December 2019].

[22] Taxi Trips-Chicago Data Portal, 11 December 2019. [Online]. Available: https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew. [Accessed December 2019].

[23] Bradley, P. S., Bennett, K. P., & Demiriz, A. (2000). Constrained k-means clustering. *Microsoft Research, Redmond*, *20*(0), 0.