User Profiling through Cluster Investigation enriched by a Pre-User-Profiling Method

Eleni Siampali
STUDENT NUMBER: 2032780

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

dr. Giovanni Cassani
dr. Drew Henrickson

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
June 2020

Preface

Four months ago, I was engaged in researching and writing this thesis project. It has been written to fulfill the graduation requirements of the Data Science & Society Program at Tilburg University (TiU). It is the report of a long process. It cannot express the long days spent in front of my computer, the hope for good results, and the sadness with each failed attempt. However, with the right direction and unwavering encouragement, an important goal has been achieved.

I would like to thank my supervisor, dr. Giovanni Cassani, for his excellent ongoing assistance during this process. With his experience and insightful suggestions, he consistently steered me in the right direction. I am extremely grateful to collaborate with such a person. Thanks should also go to Mr. Jacobo Tagliabue for his useful contribution by providing me with the dataset. Besides, I very much appreciate my other colleagues at Tilburg University; I was pleased with our cooperation, and it was always helpful to bat ideas around with them. Finally, I want to express my gratitude to my parents and my friends for providing me with continuous support throughout my studies; their wise counsel and kind words have, as always, served me well.

Eleni Siampali
Tilburg, June 2020

User Profiling through Cluster Investigation enriched by a Pre-User-Profiling Method

Eleni Siampali

Nowadays, e-commerce websites provide tremendous amounts of clickstream data generated by their users on a daily basis. With such enormous amounts of data becoming available, users' browsing behavior, and their corresponding patterns of interest can be discovered. Previous studies have attempted to profile users and identify such patterns using either descriptive analysis or clustering techniques. Works that did utilize clustering, mainly categorized users based on attributes like demographics, page-content, and product-related details, along with the time spent on each product. This research is dedicated to the exploitation of the data stored on Web servers (log-file data) and is restricted in making user profiling based only on sequential click events. In specific, the experiments were conducted using a dataset that contains information about sessions performed by several users on an e-commerce website without any access to users' personal information or product categories. To reinforce categorization, a new approach, called "pre-user-profiling", is proposed to develop an additional feature corresponding to different ratings assigned to each user. A Partitioning, a Probabilistic Model-based, and a Density Model-based algorithm are implemented to check each algorithm's applicability in such user categorization with the findings suggesting the first one as the best, whereas the Density-based model did not serve the desired outcome in terms of identifying common behavior. Another important finding outlines the contribution of the pre-user-profiling process, suggesting its applicability as an extension to similar studies. This may lead to the improvement of existing approaches, and further insights can be gained for effective business strategies.

1. Introduction

1.1 Topic and Context

Over the last decade, an increased interest in the number of online Web services available and the number of their users has been remarked (Kemp 2019). This increase has been a critical driver of e-commerce growth, and buying products from e-commerce websites has become a new trend (Nashar, DP, and Parashakti 2020). As a result, website administrators can track users' interactions with websites in greater detail. Although this is promising for a site's success, the complexity in users' behavior makes it challenging to frame the degree of personalization that a website can offer in presenting its services. Hence, the need for a site that recognizes its users' preferences based on their behavioral navigation is becoming a fundamental issue for online retailers, both for better decision-making strategies and their recommender systems' improvement (Patil 2019). This research aims to make user profiling and identify user segments based on their clickstream activity by implementing unsupervised clustering algorithms. Feature engineering techniques, along with a pre-user-profiling method, are also developed to create features that can be used for making such user segmentation.

1.2 Relevance and Importance

The interest in investigating useful usage patterns from the Web, with the intent of customizing personalized service for a target market, is growing (Anshari et al. 2019). More and more organizations are now focusing on building mechanisms to understand the usual activities of customers or potential customers so as to deliver specific recommendations and convince them to purchase (Chen et al. 2020). Web Usage Mining is "the application to automatically discover and extract useful information from a particular website" (Adeniyi, Wei, and Yongquan 2016, p. 91). Recently, several research activities have investigated Web Usage Mining techniques, and a lot of works have been published on user profiling, with a variety of traditional machine learning methods. The methods vary, from descriptive analysis to clustering and partitioning of similarity graphs. Descriptive research measures user interest mainly by relying on customer ratings (Dollard 2017). Knowledge about the content and the associated advertisements are also available here. Researchers also tried to gain deeper insights into user segmentation. In Wang et al. (2016) and Wang et al. (2017), hierarchical divisive clustering is employed. Although this method is successful, it tends to create a different cluster for each individual user, which is not always accurate in real datasets. In some other cases of research, a more advanced clustering algorithm is investigated by making use of the cosine similarity (Su and Chen 2015).

Among these, unsupervised clickstream clustering is the more promising technique for retrieving insights about user groups with common browsing behavior. However, most of them rely either on demographics and the page content of the website or on the users' common interests, i.e., purchases of the same products (Li et al. 2019). Less emphasis is given on the log-file content. Or, regarding those interested in that, they mainly focus on the visiting paths, the browsing frequency and the time spent on each product category, to reveal user interests and, subsequently, make user profiling.

The data used in this study offers the opportunity to explore more attributes based exclusively on sequential click events and profile users from the extracted patterns. The number of different types of clicks, the dwell time spent on each click as well as the activity ratio; how active the user was during the whole period that is examined, can generate further insights for user profiling on e-commerce websites. Through feature engineering techniques, a pre-user-profiling is also made by creating different user categories, relying on the attributes mentioned above; the different types of clicks, and the time spent on each click. This "pre-user-profiling" leads to the extraction of an extra feature that contributes to the rest, with the intent to make different user segments. Subsequently, valuable insights can be gained from clicks, through which e-commerce companies can make recommendations and precise decision-making.

1.3 Research Questions

Consequently, this thesis focuses on answering the following research questions with the sub-question concerned:

RQ1: Does a pre-user-profiling method contribute to the development of a dataset which leads to user categorization from click events?

RQ2: How do Partitioning, Probabilistic Model-based, and Density-based clustering algorithms perform in making user profiles from click events, including the pre-user-profiling method?

- *Sub-Q1: Are there any differences between these algorithms in making such user categorization?*

To answer these questions, features from sequential click events/actions are extracted, along with an extra feature that represents user profiles and is formulated based on some of the extracted features. Also, the features' dimensions are reduced with the Principal Component Analysis (PCA) (Viswanath et al. 2014), and the data is clustered using the K-means (Kapil, Chawla, and Ansari 2016), the Gaussian Mixture Models - GMMs (Li et al. 2018), and the Density-Based Spatial clustering of Applications with Noise - DBSCAN (Sharma, Gupta, and Tiwari 2016) clustering techniques. Each algorithm is applied twice to our data, to evaluate the contribution of the pre-user-profiling method to the analysis; one including the extra feature and another without that. To compare the algorithms' performance between the two similar processes, the average Silhouette score (Ogbuabor and Ugwoke 2018) and the Standard deviation (Azad et al. 2019) for a specific range of clusters across each algorithm are calculated. The results indicate that better performance is achieved when pre-user-profiling is used, and the method contributes towards forming a set of features that lead to an accurate user segmentation. In particular, K-means and GMMs produce five different groups of users with similar navigation behavior. On the other hand, the users' categorization through DBSCAN into 78 clusters, fails to distinguish representative user groups. Instead, it clusters the majority into two groups, one consisting of users with unusual behavior, and the minority into several different groups. For the evaluation of the different algorithms, including the pre-user-profiling, the average Silhouette score is also calculated as well as the average Davies-Bouldin index (Sitompul, Sitompul, and Sihombing 2019). The outcome marks the K-means as the best algorithm for making user profiling from these data.

1.4 Overview of the Structure

The rest part of this work is structured as follows: Section 2 provides references to earlier similar research approaches and particularly to various user profiling methods not only on e-commerce websites but also on social networks and other applications. Section 3 describes the overall strategy of this study and the context material used by unsupervised clustering algorithms for user profiling. Section 4 offers a comprehensive overview of the experimental set-up. Here, details about feature extraction, the methods used to create different user profiles, as well as the dataset used are included. In Section 5, the results with the important findings are presented, and thereafter, in Section 6, the contribution of the analysis and the efficiency of the proposed methods are assessed. At last, Section 7 summarizes the conclusions of this study based on the research questions and an overview of how the results could help the current research is provided.

2. Related Work

2.1 Web Usage Mining and Clickstream Analysis

With the increase of the Internet's worldwide usage, products and services are used extensively through this distribution channel, and a large number of online applications available ranges from social networking to e-commerce (Souiden, Ladhari, and Chiadmi 2019). Social media features facilitate user interaction and online customer recommendations, as well as dissemination of information about newly released products. At the same time, e-commerce websites offer new opportunities for online purchasing. The compilation of comprehensive information regarding individuals' activities when browsing online has been enabled due to technology advancement (Rao and Arora 2017). Suck kind of information mostly corresponds to clicks, and it is known as clickstream data. Several previous studies have proposed Web Usage Mining or so-called Clickstream Analysis as "a procedure used to discovering important and useful knowledge or information" (Tabassum, Shoaib, and Sarfarz 2016, p. 519) from such data stored in server log files. This is useful not only for people accessing information from the Internet but also for many applications, such as e-commerce websites, to conduct customized marketing e-services. More applications developed are related to government agencies to recognize threats and terrorism, fraud detection to identify fraudulent activities, and businesses to create stronger consumer relationships and enhance their business (Talakokkula 2015).

2.2 User Profiling on the Web

Since Web services are strongly connected to user participation, it would be challenging to understand the complex nature of user behavior. Numerous recent studies have implemented user profiling for two reasons: information security and commercial benefit. Wang et al. (2013), Ruan et al. (2015) and Alswiti et al. (2016) first proposed the idea of making user profiling for compromised account detection in Online Social Networks (OSN), using behavioral features that can adequately characterize users' interactions. Based on a "quantification scheme", these features can precisely identify individual users of Online Services and detect compromised accounts (Cao et al. 2017). A social graph-based approach for finding hidden and suspicious accounts in OSN is proposed by Cao et al. (2017), which introduces a concept called "forwarding message tree". Another study focused specifically on identifying anomalous behavior of Facebook users by observing the change in users' reactions/posts using unsupervised clustering (Savyan and Bhanu 2017).

Similar approaches make user profiling for fraud detection in online advertising. With each successful click on an advertisement, Internet network providers receive a commission and seek to raise their revenue by inflating the number of clicks in these so-called "Pay-Per-Click (PPC)" models. Beránek, Nỳdl, and Remeš3 (2016) presented how to unmask this click fraud and identify those accounts using different time characteristics combined into a "time print". Knowing what common daily activities are, the distinction between strange and ordinary users has been achieved through classification algorithms. In more recent research, fraud and shared user accounts have also been established based on clickstream data. The recommended research, however, effectively introduces its applicability to various domains by the use of a hybrid approach consisting of "subsymbolic and symbolic Artificial Intelligence methods" (Weller 2018, p. 822).

2.3 User Profiling in E-commerce Business

In recent years, user profiling in e-commerce seems to be essential, and one of the most critical phases for more personalized recommendations and better decision-making strategies (Zumstein and Kotowski 2020). Thus, academic research has focused on understanding user interest and investigating how user profiling can contribute and improve the websites' success. Parikh and Shah (2015) worked on creating a recommendation system by generating product recommendations that customers may wish to purchase. To achieve this, they made user profiling based on users' transactions, and created rules for the propensity to buy customers; customer's purchased products and customer's profile. However, as association rule mining takes longer to find the rules, the hierarchical clustering technique was used instead, with top-down and bottom-up approaches. Thereby, each unique object is assigned to a cluster along with the corresponding transaction. Data with demographic characteristics (gender, age, place) is also used in another study for creating different user categories. Madke et al. (2018) proposed a model to analyze clickstreams of e-customers, extract information about their shopping behavior, and make predictions. This model predicts the customer's category of most likely purchased products on a digital marketplace and thus gives product recommendations. For analysis and prediction, the Naive Bayes algorithm is used. Also, the dataset includes information about the total time spent on the website, the mouse button movements (categories and products searched), which offer some clues about their purchasing behavior, as well as the number of items and the product item categories in the basket. Both studies aimed to group users based on their actions when browsing on the website using supervised and unsupervised machine learning; however, to identify user interests, they relied on either demographics or page content and item-related features. Besides, they used historical data to make such profiles and a further pattern recognition.

In another work, Zaim, Haddi, and Ramdani (2019) proposed a hybrid user profiling method, using the Decision Tree algorithm, based on users' navigation sessions and reviews. The dataset contains records, from TMALL website, of user-item interactions (user's actions/type of clicks), delivery times, and reviews, considering customer satisfaction as target attribute. With the proposed method, marketing rules can be developed to match customers' satisfaction categories, and accurate customer profiles can be achieved about e-service quality assessment. Besides profiling from clickstream data with supervised learning, approaches that make use of unsupervised clustering have also been undertaken. In Chen and Su (2013) and Su and Chen (2015), a set of three features is analyzed; visiting path, frequency, and time spent on each category for 10,000 users. Based on these behavioral data, a "leader clustering algorithm" is built, which, along with a set of rules, generates patterns of user interests. The user interest here is defined as a set of product categories that the user has visited. Although this approach uses clustering techniques to differentiate the users, it mainly focuses on the algorithm's improvement by relying again on product characteristics and product categories.

To some degree, this thesis builds upon the work of other studies. Unlike these, it aims to focus specifically on the manipulation of the log-file content and decide whether features, i.e., number of different types of clicks, time spent on each click, extracted only from click events (users' behavior on the website), can lead to an adequate distinction and creation of different user profiles. Even with or without demographics,

page-content, item-content, or personal information, there is still much information in the sequential clickstream data that can be extracted using feature engineering techniques. Another aspect that differentiates this work is the development of an additional feature, using a "pre-user-profiling" method, which represents buyers' categories (fast buyer, normal buyer, slow buyer), viewers, and navigators, based on the above-listed attributes. Such an approach seems promising, and the impact of such a feature in the creation of a dataset that can be used for making user profiling is worth-investigating. It would also be interesting to see how various clustering algorithms make user profiling when feeding with all those features and to test if an in-depth understanding of the different user groups' behavior can be made. If so, such groups can help further research to recognize users' common habits based only on clicks.

2.4 Clickstream Clustering

Markov Chain Models and Association Rules are not usually sufficient for capturing the complex users' behavior. Unsupervised clickstream clustering is considered to be the best approach for retrieving interesting user group information, which is also indicated by the previous studies described above. Clustering is a Machine Learning technique, and it is central to many areas of data-driven applications. Existing approaches focus on different types of clustering (Wang et al. 2016, 2017), and it has been studied extensively from various perspectives. What defines a cluster? What is the right distance metric? How to make efficient group instances? Numerous different distance functions and embedding methods have been investigated. The notion behind clustering is to ascribe the objects to clusters in a way that makes the objects in one more homogeneous to others. In fact, it makes groups/clusters of a given dataset using a similarity metric by separating points based on their common characteristics (Kaur, Sahiwal, and Kaur 2012). Generally, clustering analysis is used to gain valuable insights from the data and identify patterns by seeing into what categories the points fall.

Clustering algorithms are split into five categories. This study focuses on the Partitioning, Density-based, and Model-based clustering algorithms to investigate how algorithms of those categories perform in these data. In specific, the K-means, the Gaussian Mixture Models (GMMs), and the Density-Based Spatial clustering of Applications with Noise (DBSCAN) are applied for making user profiling, one from each category.

K-means is a very popular *partitioning method* and has been extensively used. Some of its applications are related to marketing (Wang et al. 2010) and image segmentation (Dhanachandra, Manglem, and Chanu 2015) as well as document grouping (Balabantaray, Sarma, and Jha 2015). GMMs is a *probabilistic model-based algorithm* for representing normally distributed areas within an overall population. This algorithm has been recently used for feature extraction from speech data for speech recognition systems (Yu and Deng 2016). In other studies, the identification of different seismic events in a region (Kuyuk et al. 2012), the quantitative global assessment of brain tissue changes (Moraru et al. 2019), and the extrasolar planets' analysis (Hung and Chang-Chien 2017) are some of its applications. Lastly, DBSCAN is a *density-based non-parametric algorithm* and is excellent at separating objects with complex shapes and homogenous local density. In addition, it performs well with the outliers in a dataset. In many real-world applications, such as social network community identification and satellite image analysis, successful approaches that used DBSCAN to identify regions

that are characterized by locally dense areas, have been proposed (ElBarawy, Mohamed, and Ghali 2014; Harsono, Basuki et al. 2018). Thus, with such a variety of applications, it would be interesting to investigate their applicability to the identification of different user profiles on an e-commerce website.

2.5 Summary

All studies reviewed in this chapter establish that user profiling in e-commerce has been investigated with success. However, such studies have limited focus on the log-file content, dealing mostly with the modeling approach, and rely on the page-content and item-content information. This study aims to assess the extent to which click events, along with a pre-user-profiling process, can lead to the categorization of different user profiles on an e-commerce website, utilizing different clustering techniques.

3. Methods

For a better understanding of this study's structure, a high-level overview of our data is provided with more details to be offered in the next section. The whole analysis is carried out on clickstream data of a fashion e-commerce website and includes clicks that users performed during several sessions. Sessions correspond to visits and clicks to different mouse actions, where each action comes with the date that it happened, the user that made it, the URL, and the product_id. All clicks are presented in sessions (session-level) with different users to perform one or more of them and several clicks to be included during each session.

The structure outlined in this research comprises four main parts. The first one is the data pre-processing, which is performed at session-level and user-level. Specifically, feature engineering techniques are applied to extract information for each session firstly. Here, the most important features for our analysis are kept, which reflect the user's clicking behavior, and new features are extracted from there, including also a pre-user-profiling method. Then, all this information is aggregated per user, and more features are created. The second part is the data preparation, which includes standardization of the feature set and application of the Principal Component Analysis (PCA). PCA is used to extract critical information and to express this information as a collection of summary indices. This will help to observe trends and clusters in our data. The next part is the clustering analysis, which consists of the implementation of one clustering algorithm per category (Partitioning, Density-based, Model-based), with the parameters optimization. The second and third steps are applied twice to evaluate the newly proposed method, one without the extra feature extracted from there and one including it. So, different outcomes are derived, and each algorithm's performance across the two similar processes is assessed using the average silhouette score. After the selection of the best approach for each algorithm, another evaluation is made using both the average silhouette score and the average davies-bouldin index. This evaluation includes a comparison between the different clustering algorithms' performance to provide a context of their results and check their applicability to this kind of data. Finally, a set of rules is made for each cluster of the best algorithm to observe trends and user interests based on their clicking behavior.

3.1 Feature Engineering Approach

The first step, data pre-processing, is the most critical part of this research and is performed in two stages. Initially, the most important features are used; clicks, the time that each click was performed, and the user_id. Based on them, new features are extracted, at session-level, regarding the total number of different types of clicks and the total time spent on each click. Furthermore, a pre-user-profiling method is proposed, which leads to the creation of another feature. This feature corresponds to different user categories and is explained in the next section. After that, this information is aggregated per user (user-level) based on the features' average values across each user's sessions. The last two features that are extracted at user-level are related to the number of users' visits on the website and how active a user was during the whole period that is examined. The Experimental setup section offers more explanations about this step.

3.2 Data Preparation

In this second step, the data is transformed to follow a normal distribution using the *Quantile Transformer* (Bogner et al. 2012). This method "provides non-linear transformations in which distances between marginal outliers and inliers are shrunk" (Bogner et al. 2012, p. 1089). Therefore, due to its tendency to weaken outliers, it is a robust pre-processing technique and is chosen for this research. In this step, Principal Component Analysis (Viswanath et al. 2014) is also applied to retrieve the most important information towards performing user segmentation. PCA is necessary for large datasets, and its main idea is to "reduce the dimensionality of a dataset consisting of many variables correlated with each other, while retaining the variation present in the dataset, up to the maximum extent" (Bro and Smilde 2014, p. 2813). So, in our case, the dataset is transformed into a smaller one with a new set of variables, known as principal components (PCs), ordered in a way that the retention in the variation decreases while moving down in order. Another purpose of using PCA is to discover the important features of our dataset, which reveals relationships by allowing interpretations.

3.3 Clustering Algorithms

After the transformation of our data to the appropriate format, in the third step, the modeling development takes place. This involves unsupervised clustering techniques. Since our data consists of continuous variables, and our goal is to create groups, clustering does a nice job of finding the boundaries between groups, and this is the primary reason for using it. Besides, clustering techniques are suitable for this project, as no indications about user interest are available, and inferences can be made based only on the input features, without referring to known outcomes. Particularly, K-means, GMMs, and DBSCAN are applied to separate the users into different clusters, and we are interested in identifying their applicability to this kind of data.

K-means

It is a partitioning clustering algorithm proposed by Hartigan and Wong (1979). This algorithm attempts to assign each user on k clusters, characterized by their centroids (Kapil, Chawla, and Ansari 2016). k value has to be chosen apriori, and then datapoints are split into clusters of the same area, based on a minimum distance. The Euclidean distance is used here based on an objective function (see below equation) (Jain 2010).

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} (||x_i - u_j||)^2,$$

where $||x_i - u_j||$ is the Euclidean distance between a point, x_i, and a centroid, u_j.

To determine the appropriate number of k value, the average silhouette score from a specific range of clusters is used. K-means generalizes well to clusters of different shapes and sizes, and it is powerful for cluster identification. Therefore, due to its simplicity and performance in other fields of research, it is a suitable point of reference for clustering comparison and was chosen to be investigated.

GMMs

It is a probabilistic model-based algorithm and can be used for finding clusters in the same manner as K-means. Nevertheless, as K-means does not work for round-shaped clusters, because of the distance function that measures from the cluster center, GMMs is used to address such problems. It is also considered to be a powerful tool that tends to group datapoints that belong to a single distribution using the mean and the standard deviation (Biernacki, Celeux, and Govaert 2000). This algorithm is wholly determined by the parameters of its individual components, and to find its parameters for each cluster, an optimization algorithm is used, called Expectation-Maximization (EM). The EM algorithm's function is to find parameters of the maximum likelihood estimates (MLE) in models, where the parameters cannot be identified directly (Li et al. 2018). For the optimization of the components, the average silhouette score of different range of clusters is also used here.

DBSCAN

It is a density-based algorithm where the number of clusters does not need to be defined apriori. This may result to not perform well in high dimensional data. Given a set of points, DBSCAN makes groups of close points, based on a distance metric, labeling as outliers points in low-density regions (Sharma, Gupta, and Tiwari 2016). This algorithm can create clusters of different properties in terms of shape and size, and uses two hyper-parameters; eps and $minPts$. The eps value specifies the maximum distance between points to be considered as part of a cluster, and $minPts$ is the minimum number of points for a cluster's formation. Both need to be carefully tuned as they would affect the number and formations of the final clusters. An important characteristic of this algorithm relates to its tendency to account for noise. Thus, since our data may contain many outliers, it is worth-investigating. Regarding the optimal minPts value, it should be double the dimensions, according to Sander et al. (1998). For the eps value, it can be found through a distance-graph using the K-nearest neighbor method (Satopaa et al. 2011), where the k value is set to be equal to the minPts value (Pirrone et al. 2018).

3.4 Evaluation

The evaluation is performed in two stages; evaluation of the pre-user-profiling method and clustering algorithms evaluation.

3.4.1 Pre-user-profiling Method Evaluation.

The pre-user-profiling method is assessed in order to determine whether it contributes to the creation of different user profiles. For this reason, the algorithms used, including standardization, PCA, and parameters optimization, are applied twice to provide a context of comparison. Firstly, the new features without the one derived from this method are fed to the clustering algorithms, and then the same approach is followed, including it. As evaluation metric, the silhouette score is used, which validates cluster cohesion and cluster separation. Cohesion determines the possibility of a point to belong to a specific cluster, and separation defines the level of clusters' distinction. Given that, it is a helpful metric as the validity of each object's assignment in a specific cluster is evaluated based on them. Silhouette values lie between -1 to +1.

"Negative silhouette values show the incorrect placement of objects, while positive values represent better objects placement".(Bruno et al. 2014, p. 47).

For a given point i in a cluster, its silhouette score, s_i is defined as follows (Kaufman and Rousseeuw 2009, p. 96):

$$s\,(i) = \frac{b(i)-a(i)}{max[a(i),b(i)]}, \text{ where}$$

- "$b(i)$ is the distance between i and its nearest cluster centroid", and
- "$a(i)$ is the average distance between i and all other points of the cluster"

In this study, the average silhouette score for a specific range of clusters is calculated, and it is used to evaluate our approach as well as for the parameters optimization.

3.4.2 Clustering Algorithms Evaluation.

The algorithms' evaluation is also made in the subject of cluster cohesion and cluster separation using the average silhouette score described above. According to Bruno et al. (2014), the calculation of the silhouette coefficients is usually used to determine the appropriate clustering algorithms. Another metric, davies-bouldin index, is also used, which is the ratio of the intra-cluster and inter-cluster distance based on the Euclidean distance (Sitompul, Sitompul, and Sihombing 2019). Its values bound between 0 and 1, with values close to 0 to be good. Both metrics are helpful, and based on them, a comparison is made across all algorithms to choose the most suitable one for this study's purpose.

Finally, a set of rules for each cluster of the best algorithm is built to identify user preferences from clicks. From each cluster, several patterns can be set which will frame their tendency when visiting a website.

4. Experimental Setup

4.1 Dataset Description

For this study, a clickstream dataset was provided by the thesis supervisor in collaboration with Mr. Jacobo Tagliabue (Bigon et al. 2019). The data was collected from a group of 212,036 users who performed 443,662 real sessions (visits on the website) in total. All events were sampled from a period of 18 days in 2018. This dataset provides a significant opportunity to enhance feature representation, and investigate several user patterns, from a different perspective, based on sequential click events. Such feature representation can contribute to the existing research by strengthening current approaches that mostly rely on other attributes for user profiling, i.e., product categories, demographics, and can generate further insights for better and more accurate business strategies.

The dataset includes information about the website's sessions performed by several users. In specific, each row represents a distinct event ("product_action") that was recorded on a specific date-time on the website. The events vary and may be "detail", "add", "remove", "buy" and "click". Besides, an event/action may come with a 'NaN' value, which for this project represents "view". Each action comes with the anonymized information about the user performing it ("client_user_id_hash" for not-logged-in users, "target_user_id_hash" for users after login), standard temporal meta-data ("server_date" or "server_timestamp_epoch_ms"), the "session_id", the "URL" and the list of product identifiers ("product_id") connected to the event, all collected in compliance with the legislation to protect user privacy. It is worth-mentioning that each row has only one event with all the information, while another row may have another or the same event from the same user and the same session. So, we can say that there is a representation of a sequence of consecutive events performed by a user, sorted by order of arrival, with less than 30 minutes between each set (session's duration). The total size of the dataset consists of 5,433,613 rows (events) performed by users in several sessions. **Table 1** summarizes the statistics of the data.

Table 1
Dataset statistics

Number of sessions	443,662
Number of target users	10,184
Number of client users	212,036
Number of product actions	5
Number of different date-times	5,366,134
Number of different dates (days)	18
Number of products	38,345
Number of URLs	256,599
Average number of actions per user	25.63
Average number of sessions per user	2.10

This research makes use of the followings:

- client_user_id_hash
- session_id
- server_timestamp_epoch_ms
- product_action

The session_id field captures all information, including the product actions -with the date and user- performed within that session. A session starts with the first product action performed on the site. The data indicates that the majority performed less than 200 clicks, while the rest, reacted more times with the number to range from 200 to 12,110 clicks (maximum number of clicks). *Figure 1* illustrates the number of clicks made from 100 randomly selected users. For the representation of the actions of all users, another graph (histogram) was created, showing the number of actions in bins (*Figure 2*). Since the data range is wide, the chart is presented on a logarithmic scale (base 10), with the number of bins to be determined according to the Freedman-Diaconis rule (Olson 2018).

Figure 1
Actions performed by user for 100 randomly selected users. The no. of actions varies with some to reach even the 100 clicks, while most of them remain in lower levels.

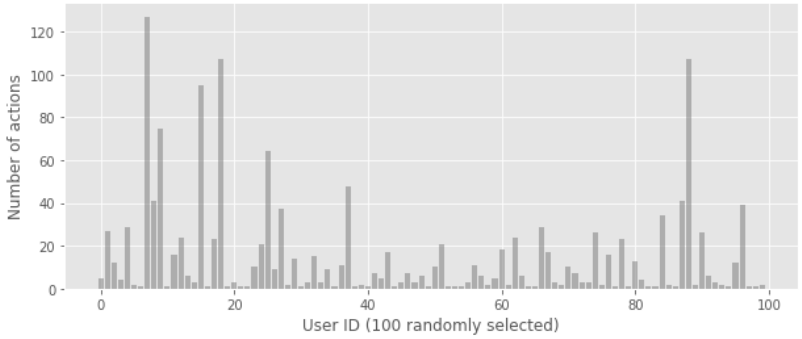
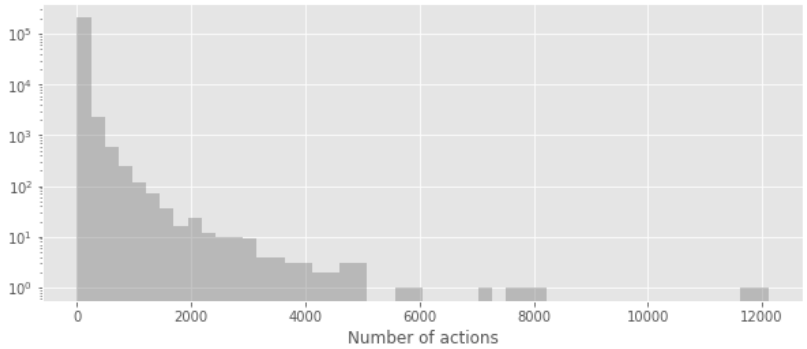


Figure 2
Actions performed by all users presented in a histogram of logarithmic scale. A large number of users made less than 200 clicks (first bin), while a continuous decrease is observed between 200 and 5,000 clicks. A minor percentage is also shown to have acted around 6,000, 8,000, and even 12,000 times.



In *Figure 3*, the sessions of 100 random users are shown. Most of them visited the website once, with some fluctuations between 3 and 9 times to be observed. In *Figure 4*, the distribution of sessions for all users is presented, which confirms that almost all visitors had around 10 sessions, while a minor percentage performed an extraordinary number of visits reaching the 266 sessions.

Figure 3
Sessions performed by user for 100 randomly selected users. The no. of sessions seems to be stable at 1 or 2, whereas a minority of these 100 users visited the website more times.

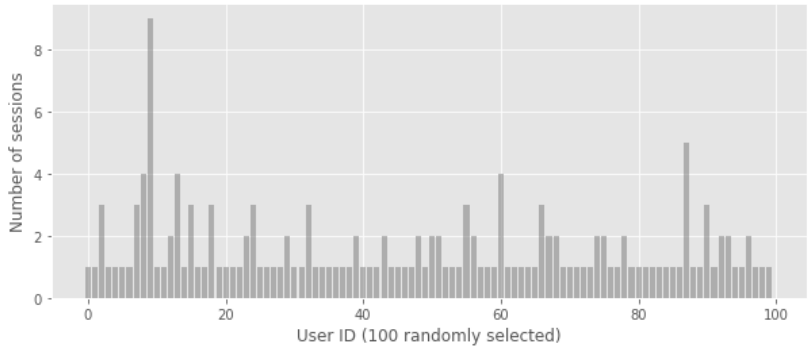
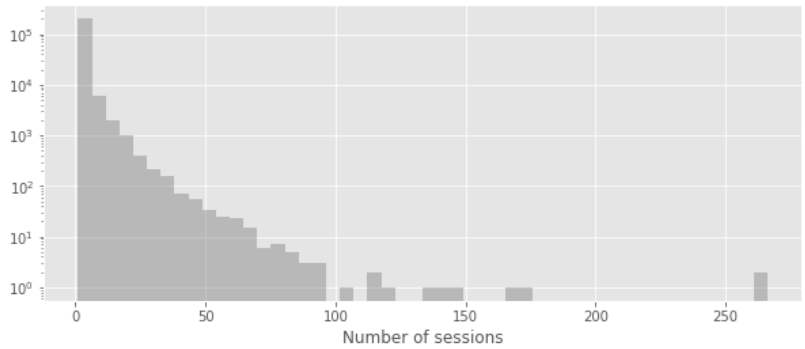


Figure 4
Sessions performed by all users presented in a histogram of logarithmic scale. The majority of users performed 10 visits or less on the website, whereas a small amount had more than 100 sessions.



Note that in the initial data, five different types of actions are present, but in this study, one more type is created from 'NaN' values, as mentioned at the beginning of this section. This will add more information to the analysis by capturing users that visited the website without performing any action. Also, for the rest of this project, sessions with the number of actions being smaller than one, and subsequently, users with only one such session, are excluded.

4.2 Feature Engineering - Preprocessing

4.2.1 Session-level.

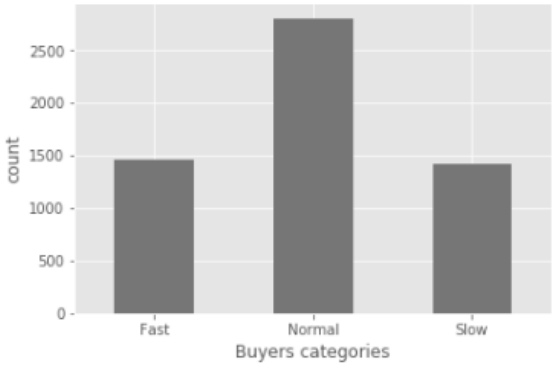
Each session is handled as a sequence of consecutive actions that comes with a user_id and the corresponding information. Firstly, 14 features are extracted here; the total number of different types of actions performed in a session, the dwell time spent on each action as well as the "Start Time" (when the session started), and the "End Time" (when the session ended). For the first, as six different actions are present, six features are created and represent the *"Number of Adds"*, *"Number of Removes"*, *"Number of Details"*, *"Number of Clicks"*, *"Number of Views"*, *"Number of Purchases"*. For the dwell time, six more features are extracted; *"Add Time"*, *"Remove Time"*, *"Detail Time"*, *"Click Time"*, *"View Time"*, *"Purchase Time"*. The dwell time for each action is calculated as the time that elapsed from one action until the next one, having as a reference starting point the time that the action was performed and ending point the time that the following action happened. Given that, for each unique session, the total dwell time of each type of action is extracted. Since the dwell time for the last action cannot be calculated, it is not taken into account. After the creation of the above-listed attributes, all are normalized based on the length (number of the performed actions) of the session. Normalization is vital here and a crucial pre-processing part of features' counterpoising effects in the hope of achieving objectivity and more robust results. The last two features are the *"Start Time"* and *"End Time"*, and both are extracted based on the time that the first action and the time that the last action took place respectively in a session.

Pre-user-profiling method

At session-level, one more feature is extracted, "User Profile", through a pre-user-profiling method. Buyers and non-buyers are categorized into five different groups, based on their clicking behavior in a session. For the buyers, if at least one purchase was made, three different categories are created; "fast-buyers", "normal-buyers", "slow-buyers", by taking into account only the first purchase. On the other hand, for the non-buyers, two categories are created considering now the whole sequence of events/clicks; "viewers" that performed not at all any action rather than "view", and "navigators" if not only "view" but also at least one of the other actions were made. The importance of this categorization aims to provide more realistic results and to contribute towards the identification of useful patterns.

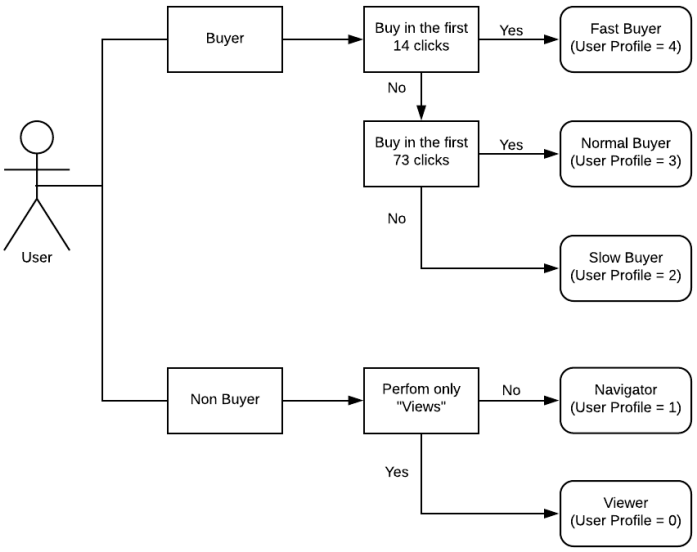
According to the above-mentioned criterion, buyers capture only 3% of all users. To define the thresholds of clicks, the *Quantile-based discretization function* (Kazachuk et al. 2016) was used, and the buyers were divided into bins. "The function defines the bins using percentiles based on the data distribution, not the actual numeric edges of the bins" (Kazachuk et al. 2016, p. 420). So, if someone purchased in the first 14 clicks, he/she is a fast buyer, in the next 59 clicks normal buyer, otherwise, after 73 clicks, slow buyer. An overview of the buyers' categories is presented in *Figure 5*, while in *Figure 6*, an indication of the pre-user-profiling process through a flow-chart, including the click thresholds, is set up.

Figure 5
Buyers' categories based on their purchasing activity in terms of clicks. Buyers represent the minority of the whole population. From them, half are normal buyers, while the rest are equally distributed as fast and slow buyers.



Buyers categories	count
Fast	1,458
Normal	2,804
Slow	1,414

Figure 6
Flow-chart of the pre-user-profiling method. Buyer if at least one purchase took place; if yes, purchase in 14 clicks means fast buyer, in 73 clicks normal buyer, in more clicks slow buyer. If no purchase; only view means viewer, in case another type of action (not only view) was performed, then navigator.



4.2.2 User-level.

After the feature extraction at session-level, then all this information is aggregated per user. Specifically, the values regarding each user's sessions are compressed using the average value of them. Hence, the average number of each type of action, the average dwell time of them as well as the average value of user categories are extracted. The

other two attributes, "Start Time" and "End Time" are not compressed. Instead, both are used for the extraction of the "Activity Ratio" feature, which is explained in the next paragraph. To make the transition from session-level to user-level more clear, an example has been illustrated, shown in *Figure 7*. For the "User Profile", the values serve as ratings, and they range from 0 to 4, at session-level. For example, a value of 1.52 at user-level indicates that the user is classified somewhere between navigator and slow buyer. This results in a further scaling of our features, which provides us with more flexibility.

Figure 7
Example of the transition from session-level (table above) to user-level (table below). All sessions, along with the included information, are compressed into the corresponding users by means of their average value.

Session ID	User ID	Number of Purchases	Purchase Time (mins)	...	Start Time (timestamp)	End Time (timestamp)	User Profile
14	1	5	10	...	153789	153811	3
33	1	3	4	...	153935	153942	2
105	1	0	0	...	154241	154248	1
68	2	3	20	...	153625	153648	2
92	2	0	0	...	154430	154439	0

User ID	Number of Purchases	Purchase Time (mins)	...	User Profile	Number of Visits	Activity Ratio
1	2.6	4.6	...	2	3	0.85
2	1.5	10	...	1	2	0.42

In the second table, two more attributes are created for each user; *"Number of Visits"* and *"Activity Ratio"*.

- **Number of Visits** is determined by the number of different sessions performed on the website by each user.
- **Activity Ratio** is defined as the total time (TT) a user spent on the website divided by the total reported time. The first one is the total time a user spent in all sessions, while the total reported time is the time elapsed from the first action of the first session (FFS) to the last action of the last session (LLS). In other words, the observant time of operation about the period investigated (18 days), which remains constant. The equation below defines how the activity ratio is calculated:

$$ActivityRatio = \frac{TotalTime}{TotalReportedTime} = \frac{\sum_{i=1}^{n} TT_(i)}{|FFS-LLS|}, \text{ for } i \text{ referring to a session.}$$

Now that all features have been extracted, the rest of this study, including the next two main steps, data preparation and algorithms implementation, is applied twice, as explained in the previous section. This will provide an indication of the contribution of the proposed pre-user-profiling method for making user profiling, delivering an answer to the RQ1.

4.3 Data Preparation

Some preparation steps are required in order to have the desired feature format for modeling development.

4.3.1 Standardization.

Standardization refers to the re-scaling process for the values of the data variables as they share a similar scale. Since some of our features have different units, and the scales vary, this is a crucial step to take. This importance also stems from the role of cluster analysis, since the groups created are specified based on the distance between datapoints. The *Quantile Transformer* (Bogner et al. 2012), which follows a normal distribution, is chosen as the most suitable method for these data, which outperforms other scaling techniques, such as the *Standard Scaler* (AlDaweesh 2019). This method of transformation weakens the outliers. Because our data is not balanced enough, it is a robust scheme, which will boost the models' efficiency.

4.3.2 PCA.

Since all features are of equal weight, PCA is used to determine the main components. The selection is made dependent on the explained variance (Cruz et al. 2017). Below, two figures are presented delivering the same information; one without the "User Profile" and another with it. On the left graph of *Figure 8*, the cumulative explained variance from the principal components is shown, which accounts for 88% of the total variance for the first four components. The graph on the right serves the same result by presenting the contribution of each component separately. It is clear that after the fourth, the increase is minor for each additional component. Four components out of fifteen is a good indicator, as both goals are achieved; the feature space is significantly reduced while including most dataset information. In *Figure 9*, the same is observed with no significant increase after the fourth component, and the explained variance at that point to be also 88%. Therefore this number is selected for both approaches. We could say that the only difference between them is until the third component, where the contribution in the second figure is slightly bigger.

Figure 8
PCA without the "User Profile". Cumulative explained variance - Explained variance per PC. On the left, there is a steady increase for each additional component. The sign at four components marks the point at which the increase stops to be sharp. On the right, the drop outlines that after the 4th component, less than 5% of the variance is added.

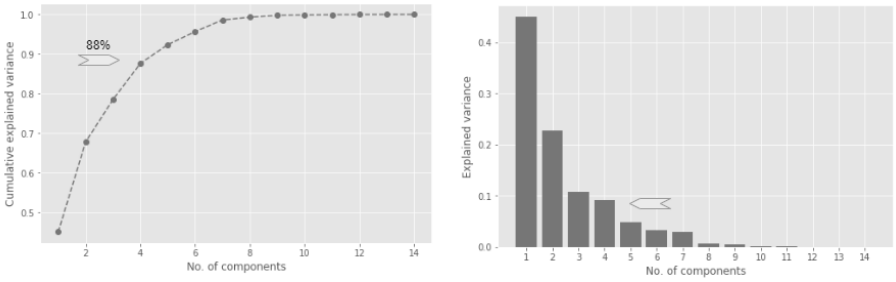
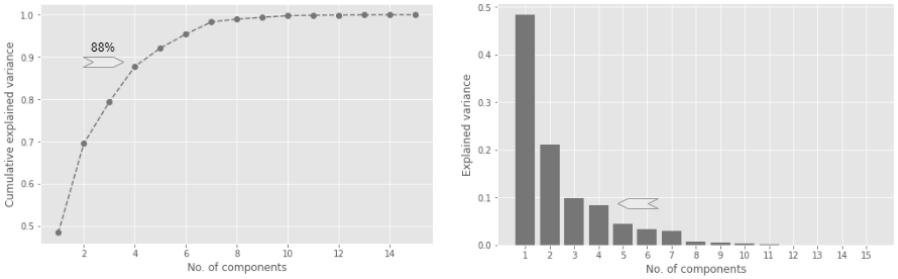


Figure 9
PCA with the "User Profile". Cumulative explained variance - Explained variance per PC. On the left, a steady increase for each additional component is also seen with the sign at four components to mark the point at which the increase stops to be sharp. On the right, the drop also outlines that after the 4th component, less than 5% of the variance is added, with the only difference to be observed until the first three components with the variance here to be slightly bigger.



4.4 Clustering Algorithms Implementation

The algorithms are implemented on the new feature set, and the parameters optimization is made using the average silhouette score.

K-means algorithm

Firstly, K-means clustering is applied to make different clusters for our data. For the optimal value of clusters (k), the average silhouette scores are calculated with the model to be executed ten times for each k. The standard deviation of the silhouette scores for each k is also calculated, which offers an estimation of the difference between each observant value from the mean. So, more robust results are taken, which explains the reason for executing the model ten times, and the one for which the model is stable with fewer fluctuations is chosen. Based on that, for k=5, we have the highest scores for both when the pre-user-profiling method is not included (*Avg. Silhouette coef.=0.6517, Std.Dev=0*) and also when we include that (*Avg. Silhouette coef.=0.6825, Std.Dev=0*). *Figure 10* and *Figure 11* show different values of k tested with the corresponding scores and the standard deviation for both approaches. The dots on the line indicate the scores, without any error to be observed for those tested. More specifically, a zero value of standard deviation shows that the silhouette value is fixed. This means that for each cluster, even when the algorithm is executed ten or fewer times, and the average is extracted, this value is always the same.

Figure 10
K-means without the "User Profile". Line-graph (on the right) of the average silhouette scores for each value of k, along with the potential errors. The algorithm has optimal clusters for k=5.

K-means tuning (without the "User Profile" feature)

Clusters	Avg. Silhouette Coef	Std.Dev
2	0.5437	0.0
3	0.6166	0.0
4	0.6188	0.0
5	**0.6517**	**0.0**
6	0.6483	0.0
7	0.6461	0.0
8	0.6417	0.0
9	0.6361	0.0
10	0.6229	0.0

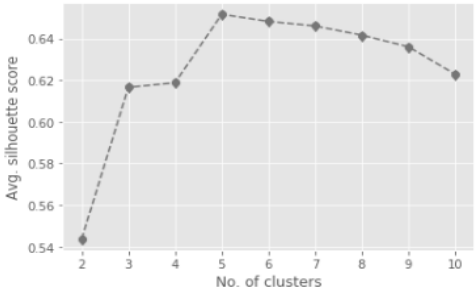
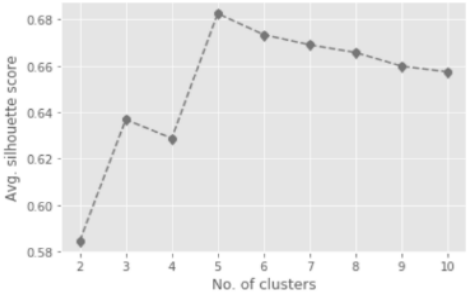


Figure 11
K-means with the "User Profile". Line-graph of the average silhouette scores for each value of k, along with the potential errors. The algorithm for this approach has also optimal clusters when k=5.

K-means tuning (with the "User Profile" feature)

Clusters	Avg. Silhouette Coef	Std.Dev
2	0.5846	0.0
3	0.6368	0.0
4	0.6288	0.0
5	**0.6825**	**0.0**
6	0.6733	0.0
7	0.6691	0.0
8	0.6656	0.0
9	0.6598	0.0
10	0.6574	0.0



GMMs algorithm

The GMMs algorithm is also applied and is determined by the parameters of its components. For the optimization of the components/clusters, the average silhouette score is also used following the same procedure with the K-means; ten times for each number of clusters along with the standard deviation. *Figure 12* and *Figure 13* show the different number of clusters tested. As can be seen, for *n_clusters=5*, we have the highest average silhouette scores for both approaches. Specifically, when the "User Profile" is excluded, we have *Avg. Silhouette coef.=0.6425* and minor error, *Std.Dev=0.0183*, while with the "User Profile", *Avg. Silhouette coef.=0.6576* and *Std.Dev=0.0182*.

Figure 12
GMMs without the "User Profile". Line-graph of the average scores for each value of n_clusters, along with the potential errors. This algorithm has optimal clusters for n_clusters=5.

GMMs tuning (without the "User Profile" feature)

Clusters	Avg. Silhouette Coef.	Std.Dev
2	0.5312	0.0339
3	0.5925	0.0344
4	0.6136	0.0165
5	**0.6425**	**0.0183**
6	0.6403	0.0187
7	0.6326	0.0240
8	0.6319	0.0263
9	0.6257	0.0189
10	0.6243	0.0341

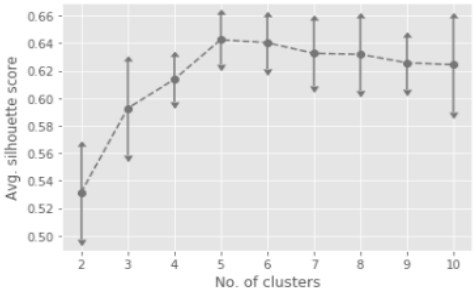
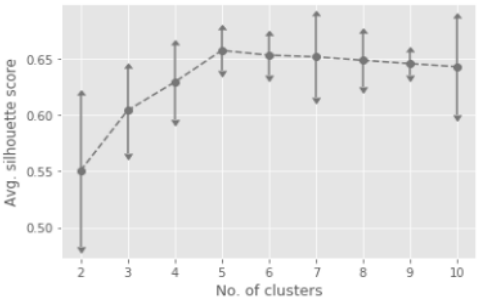


Figure 13
GMMs with the "User Profile". Line-graph of the average scores for each value of n_clusters, along with the potential errors. The optimal value here is the same, n_clusters=5.

GMMs tuning (with the "User Profile" feature)

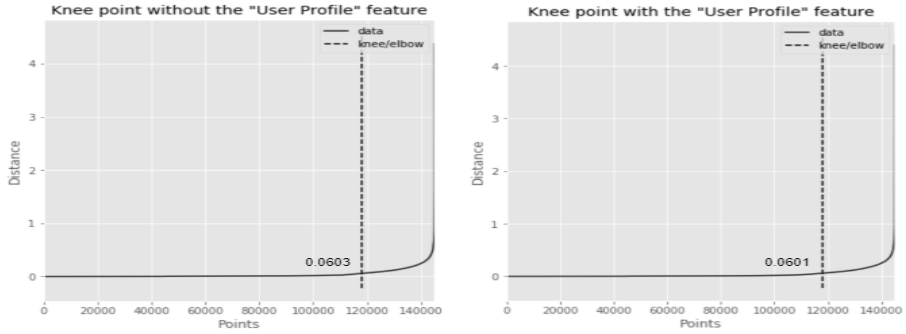
Clusters	Avg. Silhouette Coef.	Std.Dev
2	0.5505	0.0675
3	0.6042	0.0379
4	0.6294	0.0336
5	**0.6576**	**0.0182**
6	0.6534	0.0179
7	0.6519	0.0365
8	0.6488	0.0241
9	0.6459	0.0103
10	0.6431	0.0431



DBSCAN algorithm

Lastly, DBSCAN is applied, which is also tuned. Regarding its hyper-parameters, the *minPts* value is set to be 8, double the number of principal components used (Sander et al. 1998) for both cases as the same number has been selected. For the *eps* value, a systematic method for tuning is used; the k-nearest neighbor. Since the *k* value should be equal to the *minPts* value (Pirrone et al. 2018), we apply the method for *k=8*. The *KneeLocator*, proposed by Satopaa et al. (2011), is implemented to detect the Elbow point, which is used to be the *eps* value. *Figure 14* confirms that the detected knee point is at distance 0.0603 and 0.0601, for without and with the pre-user-profiling, respectively. Finally, for these parameters, the algorithm's performance is executed ten times again, and the average silhouette score and the standard deviation are calculated.

Figure 14
KneeLocator without the "User Profile" and with it, in order to find the optimal eps value. It can be seen that the detected knee point is at distance 0.0603 and 0.0601 respectively.



4.5 Software

The implementation of this project was performed using the Python programming language (version 3.7.6) in Jupyter notebook (version 6.0.2). For the feature engineering and modeling parts, the following packages were used: *pandas* (McKinney 2019), *numpy* (Oliphant et al. 2019), *scikit-learn* (Pedregosa et al. 2019), and *time* (Roberts et al. 2018). The visualizations were created through *matplotlib* (Nelli 2018) and *seaborn* (Ostrowski and Menyhárt 2020).

5. Results

In this section, the clustering algorithms performance for the approaches mentioned in section 4 on our clickstream dataset is presented. The contribution of the pre-user-profiling method on each algorithm separately and a comparison among the different algorithms implemented are illustrated.

5.1 Pre-user-profiling Method

A pre-user-profiling method was introduced, which, along with the extracted features, would result in user profiling from click events. To determine its contribution, the algorithms were implemented twice (without applying this method and with applying it), and each algorithm's performance was assessed using the average silhouette score and standard deviation. The standard deviation was used so that the approach with the lowest standard deviation value is chosen if the difference between two scores is minor. The results show that higher scores were achieved for all algorithms when the feature "User Profile" was included. In particular, for the second approach, the average silhouette coefficients for K-means, GMMs, and DBSCAN, were 0.6825, 0.6576, and 0.6644, respectively. **Table 2** presents an overview of the models' performance for both approaches.

Table 2
Clustering algorithms' performance without the pre-user-profiling and with it. Performance is measured using the average silhouette score and the standard deviation. The algorithms' implementation with the pre-user-profiling yields the best performance for all cases. The Clusters variable here indicates the number of clusters formed.

Method	Clusters	Avg. Silhouette coef.	Std. Dev.
K-means (without "pre-user-profiling")	5	0.6517	0.0
K-means (with "pre-user-profiling")	**5**	**0.6825**	**0.0**
GMMs (without "pre-user-profiling")	5	0.6425	0.0183
GMMs (with "pre-user-profiling")	**5**	**0.6576**	**0.0182**
DBSCAN (without "pre-user-profiling")	75	0.6521	0.0074
DBSCAN (with "pre-user-profiling")	**78**	**0.6644**	**0.0097**

The table above shows that the introduced pre-user-profiling method contributes to the analysis with all models to outperform when the method is used. The integration of the extra feature has the highest impact on K-means performance resulting in a 4.73% increase. GMMs and DBSCAN were also improved by 2.35% and 1.89%. The standard deviation is the same or smaller in almost all cases of the second approach, which confirms that this approach is better and robust outcomes can be drawn from there. In the third case that DBSCAN is applied, although the standard deviation is slightly bigger, this value is minor, which could not reject DBSCAN being better with pre-user-profiling.

Another important finding is that highest scores were achieved for clusters=5 (for K-means and GMMs). Although this number is the same as user categories extracted from the pre-user-profiling, the results are not biased. The reason is that these categories were

aggregated at user-lever, as previously explained, and were handled as continuous values with most numbers to range between 0 and 1.8. The results above also confirm that even with or without pre-user-profiling, best scores were achieved for clusters=5.

So, we can say that regardless of the slight difference, pre-user-profiling does contribute to the current analysis. The extracted variable enriched the data with more valuable information towards separating the users. It is also noteworthy that even without the utilization of the proposed method, good results are obtained, which means that informative features can be created from click events to make user profiling.

5.2 Clustering Algorithms

As the deployment of pre-user-profiling led to the best performance, the rest of this study continues with the approach that includes it. So, a comparison is made across the three algorithms to find out how they perform towards our objective. For evaluation, the average silhouette coefficient and the average davies-bouldin index were used, with the scores to be presented in *Figure 15*. The silhouette scores were 0.68, 0.66, and 0.66 for K-means, GMMs, and DBSCAN. The second metric was 0.62 for K-means, 0.73 for GMMs, and 0.87 for DBSCAN. Since a silhouette value close to 1 is a good score, as described in Section 3, it seems that all algorithms had relatively good results. Good values for the davies-bouldin, on the other hand, are close to 0. The results here are not that good as all algorithms had a score above 0.5. Out of them, however, the K-means yielded the best performance with the highest silhouette and the lowest davies-bouldin values.

Figure 15
Comparison of the performance between the three clustering algorithms tested. Performance is measured using the average silhouette coefficient and the average davies-bouldin index. K-means achieved the best performance for both metrics.

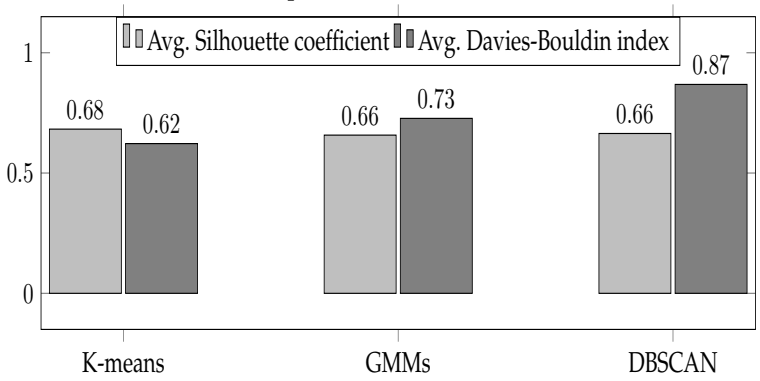


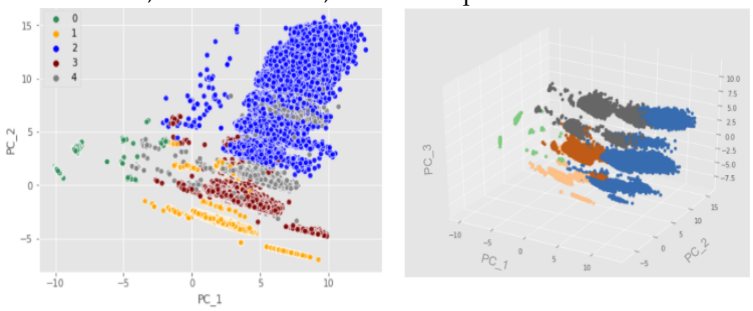
Table 3 provides the same information, as well as the clusters that were formed. The optimal solution for each algorithm produced 5 clusters, apart from DBSCAN, which separated the users into 78 different groups. It is apparent that all of them can be characterized by relatively strong cluster cohesion and cluster separation (high silhouette).

Table 3
Comparison of the performance between the three clustering algorithms tested with the corresponding no. of clusters created.

Method	Clusters	Avg. Silhouette coef.	Avg. Davies-Bouldin index
K-means	**5**	**0.6825**	**0.6224**
GMMs	5	0.6576	0.7272
DBSCAN	78	0.6644	0.8683

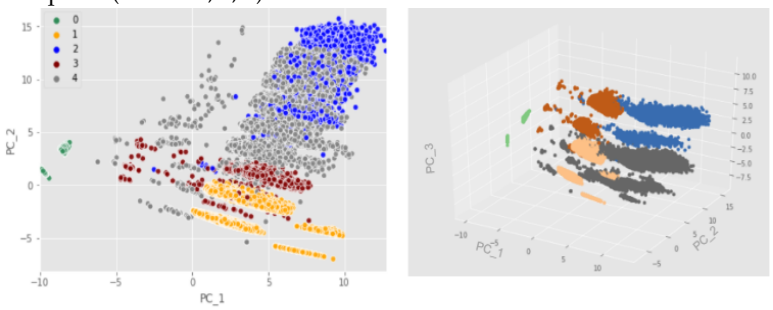
For a clearer understanding of the results and the clusters generated from each algorithm, some visualizations are provided. K-means user classes are displayed in *Figure 16*, plotted in the first two (left graph), and first three main components (right graph). Five clusters were formed, most belonging to Cluster 1. Some clusters do not seem to be well separated. The right chart provides a better view of their separation, with some to be densely connected around of their centroids (clusters 2, 3, 4), while others are more spread out (cluster 0 and cluster 1).

Figure 16
K-means clusters - 2PCs (left), 3PCs (right). Clusters 2, 3, 4 are densely connected, around of their centroids, while clusters 0, 1 are more spread out.



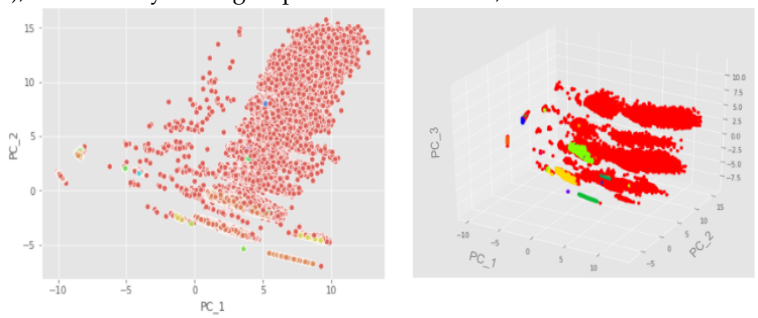
For GMMs, users were plotted in a similar manner as K-means in terms of the number of groups created. *Figure 17* shows the categorization in the first two (left graph), and first three components (right graph). Because of this algorithm's ability to find probabilistic cluster assignments, users were categorized slightly differently. Even though most users still belong to Cluster 1, in Cluster 2, we have fewer points now. However, some points continue to be around their centroids, while others are more sparse.

Figure 17
GMMs clusters - 2PCs (left), 3PCs (right). Clusters 2, 3 are densely connected, while the others are sparse (cluster 0, 1, 4).



Finally, two graphs are presented in *Figure 18*, showing the clusters that were created from DBSCAN. It is evident from both graphs that a different user categorization was made, and the vast majority was grouped into one large cluster, with 76 tiny clusters (each containing about 1% of the entire dataset) to be formed. Another big cluster was also created, including -outliers- points that do not fall into any of the other major groups.

Figure 18
DBSCAN clusters - 2PCs (left), 3PCs (right). The majority was grouped into one cluster (cluster 0), with 76 very small groups to be also created, and one more for outliers (cluster -1).



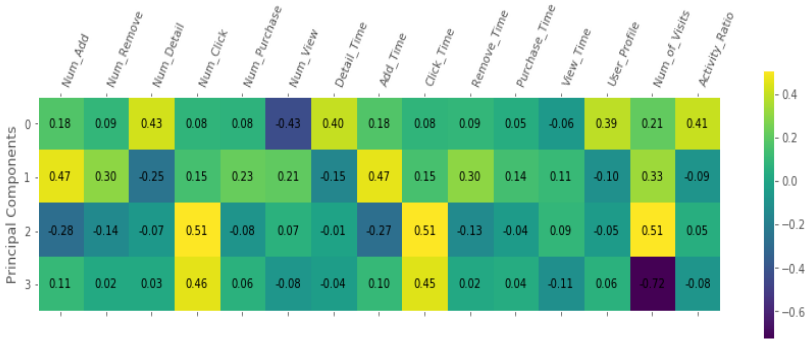
5.3 Interpretation of the K-means Clusters

Since K-means had the best performance, some interpretation of the clusters created and further investigation of the patterns that can be revealed follow here. To do this, the centroids of each cluster on the first two principal components axis in *Figure 16* were identified. **Table 4** presents the centroids. Based on them, the feature correlation on these components is investigated separately using *Figure 19*, which leads to the observation of different user profiles. Specifically, five clusters were created with most users being in the first two clusters, while clusters 2, 3, and 4 had fewer users.

Table 4
Kmeans clusters' centroids

	Centroids	
Clusters	PC_0	PC_1
0	-9.61	1.39
1	2.28	-3.56
2	6.41	7.52
3	3.07	-1.15
4	3.99	1.53

Figure 19
Features correlation per principal component.



Thus, Cluster 0 usually contains users that made only views, with that being indicated mostly from the x-axis value, while from PC_1 nothing clear is provided. Similarly, in Cluster 2, there are very active users, as from PC_0 is observed that these users had many visits on the website, and they also performed many details. However, PC_1 shows that they also added a product in their basket, or they removed an added product as well as that they purchased. For the other three clusters, however, not a thorough interpretation can be made from here with the clearest performed action to be detail. Some minor differences are related to the most time that users of Cluster 1 spent in each session and the more times that those of Cluster 4 visited the website.

To get a better understanding, a more thorough investigation was made. Different users from each cluster were checked to identify their main characteristics. **Table 5** shows such an illustration where one user per cluster is shown.

Table 5
Example of randomly selected users per cluster generated from the K-means algorithm.

	Users				
Features	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Number of Views	1.0	0.823	0.779	0.563	0.352
View Time	192.3 (s)	85.1 (s)	88.7 (s)	24.3 (s)	13.5 (s)
Number of Details	0.0	0.211	0.182	0.355	0.647
Detail Time	0.0	14.4 (s)	21.9 (s)	16.5 (s)	28.3 (s)
Number of Clicks	0.0	0.0	0.002	0.064	0.0
Click Time	0.0	0.0	1.3 (s)	1.8 (s)	0.0
Number of Adds	0.0	0.0	0.017	0.016	0.0
Add Time	0.0	0.0	2.8 (s)	2.6 (s)	0.0
Number of Removes	0.0	0.0	0.014	0.0	0.0
Remove Time	0.0	0.0	0.8 (s)	0.0	0.0
Number of Purchases	0.0	0.0	0.023	0.0	0.0
Purchase Time	0.0	0.0	2.3 (s)	0.0	0.0
User Profile	0.0	1.0	1.6	1.0	1.0
Number of Visits	1.0	1.0	15.0	4.0	3.0
Activity Ratio	0.64e-9	0.48e-5	0.23e-3	0.24e-4	0.36e-4

The same was done for more users in order to obtain a more comprehensive view, and the following observations were made.

Cluster 4 contains mainly navigators; users who performed only details and views, with the frequency of visits being more than one session. Because these users had sessions that either views or views and details were performed, their user profile ratings are between 0 and 1.
In *Cluster 0*, we have only viewers, users that made only views. What is remarkable here is that those users had only one session and, most times, only one action that was view. This explains their low activity ratio as they visited the site only once.
Cluster 1 consists of navigators; users that made details, and views. What differentiates users here from those in Cluster 4 is that they had only one session. Therefore, their user profile is always one, and their activity ratio is lower compared to those in Cluster 4.
Cluster 2 seems to be the most interesting cluster, including very active users; users who performed most of the different types of clicks. Thus, buyers are also included here. Except for buyers, there are users who did not make any purchases but were actively engaged with the website; for example, they clicked on items, added them to the basket, and then removed them. Their user profile ranges from 1 to 4, but most of them have a value of around 1.3. This means that although they were active, they did not buy very often. For example, the user from Cluster 2 in Table 4 visited the site 15 times, and its user profile is 1.6. Therefore, this user made purchases but not in every session.
Lastly, in *Cluster 3*, there are users who made many clicks but fewer than those performed by users in Cluster 2. It is worth-mentioning that none of them bought, which confirms that all buyers are classified into Cluster 2. Besides, their number of visits is lower compared to Cluster 2 users but typically varies between 1 and 10.

6. Discussion

6.1 General Discussion

This study aimed to make user profiling and identify user segments from clickstream data based on their clicks. In particular, the focus was on exploiting the log-file content through feature engineering techniques and determining whether a pre-user-profiling approach contributes towards creating a dataset for making user categorization. The purpose was also to investigate how different kinds of traditional algorithms perform in user profiling from such data. To achieve this, new features were extracted, at session-level first and user-level later, and the data was scaled with the Quantile Transformer. Then, PCA was applied, and the first four principal components were used for the clustering algorithms. After tuning the algorithms' hyper-parameters and choosing the optimal ones, the results showed that almost all algorithms had good scores managing to create different clusters. In particular, K-means performed the best with DBSCAN and GMMs to follow.

For the scaling, since our data consists of continuous values with highly skewed distribution and many outliers, the Quantile Transformer was the most suitable method as the gaps between marginal outliers and inliers were shrunk. The Standard Scaler was initially tested and resulted in poor clustering results. This is explained by the degree to which outliers affect each feature and, since the data distribution on each feature varies, the Standard Scaler cannot balance the scale, and was rejected. PCA was also crucial to our study, as the feature space was reduced, leading to uncorrelated features being developed. The first four were picked from these to be a fair representation of the initial dataset. The main aim of using this approach was not only to reduce the dimensionality but also to investigate the patterns that can be revealed from feature correlation on the first two principal components. What was observed is that, depending solely on users' clicking behavior is appropriate for user profiling, and insightful results can be obtained.

It seems that user interests when visiting a website is not only dependent on the products they buy but also on their overall behavior and clicks. Therefore, other suggestions may be made to users who visit the website very often, but, for some reason, never buy (they maybe expect for a discount?), others to loyal customers, or even be aware of those with odd habits. Another positive result addressing the *RQ1* is related to the "pre-user-profiling" method. The attribute "User Profile" generated by this and represents user categories from clicks, contributes to the analysis. It is an important feature and therefore, the method was deemed successful. What we could also say is that this research makes user profiling using a combination of descriptive analysis and unsupervised clustering, including it. There is no similar approach for clickstream data in the current research that makes a pre-user-categorization. Researchers focused mainly on doing either descriptive analysis and association rules or clustering for user profiling. More explanations about such studies are offered in *Section 6.4*.

6.2 Algorithms' Comparison

In this research, one clustering algorithm from three different categories (Partitioning, Model-based, and Density-based) was used (*Section 2.4*). In particular, K-means,

GMMs, and DBSCAN were tuned and implemented to make user profiling from user actions in clickstream data. Out of them, K-means achieved the best results, with an average silhouette score of 0.6825 and an average davies-bouldin index of 0.6224. Although its drawback in defining the optimal number of clusters to use, K-means is considered to be a strong clustering algorithm with the appropriate optimization methods. Furthermore, since our data consists only of continuous values, and there are no limitations on datapoints separation, it works effectively. Besides, partitioning algorithms can efficiently manage large datasets, as in our case, which leads to outperform the other two algorithms. GMMs achieved the lowest silhouette score (0.6576), and its davies-bouldin index was 0.7272. Although user segments were identified, and it is considered to be a powerful algorithm, its tendency to group probability-based datapoints did not yield the best performance. A possible reason stems from the fact that the general shape of the curve followed by the clusters is not known, and points are highly skewed in some areas. Finally, DBSCAN also achieved a good score, with an average silhouette of 0.6644. Its davies-bouldin index, however, was the worst (0.8683). This algorithm did not serve the desired results for this study's purpose. Many different clusters were developed, which makes it difficult to recognize user trends. Nonetheless, it successfully identified outliers and users with unusual behavior. This is very important and promising since it could be used for other purposes in a similar dataset, such as for anomaly detection analysis.

All types of algorithms achieved good silhouette scores. Their davies-bouldin scores though were not that good, with the DBSCAN to have the worst. A possible reason is the normalization, as datapoints from different clusters are very close to each other. So, since this metric measures inter-cluster separation, and the distance between clusters is not big, the scores are poor. However, all algorithms managed to form different user segments. Out of them, the Partitioning algorithm (K-means) performed the best. The others also had good silhouette scores, but DBSCAN did not meet this study's goal, which was to identify user categories with common behavior, with mostly outliers to be captured.

6.3 Patterns of User Interest

A thorough interpretation of K-means clusters was also made, with several patterns to be identified. In particular, a user profiling identification was developed employing PCA and the feature correlation matrix. What was observed is that almost all buyers and very active users that performed several different types of clicks belonged to the same group. Also, users who visited the site only once and made none click were grouped together. This is remarkable, as these users capture nearly the largest percentage of the data, and were successfully identified. For the other three groups, there is not such a clear view, though. All of their users clicked on detail with the only different characteristic to be that some had more sessions, while others spent more time during their sessions, and they were not such frequent visitors.

To get a better view of them, we extracted a set of individual users. From there, it is confirmed that buyers and very active users are in one group, and users with one session and without any action in another group. Regarding the other groups, in one of them, we have mainly navigators -they made only views and details- with one visit, while in another, navigators again but they visited the site more times. The last group consists of users with not so many sessions, but they made any type of clicks except for

purchase.

It seems that different user profiles were created, which would be interesting for a retail company. For example, special focus could be given on the group with the active users, where both buyers and potential buyers belong. Several solutions could be developed from there to encourage them to purchase (e.g., tailored recommendation systems, promotions). This would lead to more sales and growth of the company's revenue. Moreover, the group of users who did not make any click is worth-investigating, to check for potential bots. Users of this group may be responsible for an abnormally high bounce rate on the site and should be excluded.

6.4 Comparing User Profiling Methods

In the current research on user profiling in e-commerce, several methods using either supervised or unsupervised machine learning have been investigated. The experiments were carried out on different datasets, with others to contain similar information with even more attributes (click events and product categories and/or demographics), while others included customers' reviews.

Parikh and Shah (2015) created a content-based recommendation system by making user profiles from predefined user groups. A transaction data set containing 10,000 transactions was used, and they created groups based on rules of purchased products and customer's profile. They accomplished this by using hierarchical clustering, along with Association Rule Mining, which resulted in a successful user profiling from customer profile data, like age, gender, location, and transactions of other customers. Such information is not available in our case, however, and that method does not fit. A representative amount of information is required to make a decision, and creating rules based on thresholds generated from numeric data of click events is not possible.

Madke et al. (2018) analyzed user behavior using supervised machine learning techniques. They analyzed clickstreams of e-customers, extracted information about their shopping behavior, and made predictions. The dataset had a server-side system and was used to collect data from servers of organizations or businesses. To define customers' behavior and predict customers' categories of most likely purchased products, they relied on attributes like the total time spent on the website, mouse button movements (product categories, products in the basket), and demographics. Thus, the user interest identification and the grouping were made using different data by relying on demographics, page-content, and item-related features.

Zaim, Haddi, and Ramdani (2019) worked on a different manner in terms of attributes used to create marketing rules, and to make accurate user profiles. Reviews and clickstream data from TMALL, an important business unit of Alibaba Group in China, was used. The users' behavior of browsing TMALL reflects their preference for items. The data set contained 25,432,915,908 records of user-item interactions; users' actions/types of clicks, delivery times, and 241,919,749 reviews. A hybrid user profiling approach based on them was generated, and specific customer profiles were built by generating rules from the Decision Tree algorithm. Although it is about a promising study, it requires the discovery of social information and business knowledge to make the categorization.

The closest attempt to this study's approach was made from Chen and Su (2013) and Su and Chen (2015). They implemented and leveraged a "leader clustering algorithm" with a set of theory to generate patterns of user interest. User interest here is defined as the time spent on each set of product categories. Additional attributes were the visiting path and the frequency of visits. The dataset was obtained from a real website, and 10,000 user records were chosen randomly from server logs covering the entire day of May 6, 2013. Findings from the experiment showed that the newly proposed algorithm could outperform traditional clustering algorithms as well as the K-medoids, both in efficiency and effectiveness. However, it should be pointed out that products are also included here, which are essential to forming user interests. This partitioning algorithm, though, seems to be fairly stable and would offer promising results, if it was used to our features with the goal of identifying user interest from clicks.

All studies mentioned above seem promising but used demographics, page-content, and item-content features to reveal user interest, and made profiling for a different objective. They categorized the users mainly based on rules, and they used supervised or unsupervised techniques. None of them relied solely on sequential click events extracted from clickstream data to define segments using these methods. So, we may assume there is no comparison metric to check the applicability of our study to their approaches.

7. Conclusion and Future Research

In conclusion, the purpose of this work was to make user profiling on an e-commerce website based only on users' clicking behavior. This was achieved through clustering techniques, and user trends were recognized. The results suggest that interesting patterns and different user categories can be created from clicks, with the newly proposed "pre-user-profiling" method to contribute to the approach.

As for the second part of this work, it was about checking how different clustering categories identify user profiles from this data. Specifically, a partitioning (K-means), a probabilistic model-based (GMMs), and a density-based (DBSCAN) algorithm were implemented. The results showed that K-means and GMMs worked better for this study's objective, and both were able to distinguish different user segments with common behavior. The partitioning algorithm, however, performed the best (Avg. Silhouette coef.=0.6825, Avg. Davies-Bouldin index=0.6224). In contrast, the density-based algorithm did not serve the desired goal despite its high silhouette score (Avg. Silhouette coef.= 0.6644), which was still lower than K-means. It identified several small, distinct groups as well as outliers. From that, we concluded that this type of algorithm could work well for other approaches that focus more on detecting strange clicks or strange users.

Such findings illustrate the valuable knowledge that can be collected when making user profiling from sequential click events, along with a pre-user categorization for achieving better results. Another important aspect that this research accomplishes is the creation of user profiles when products, product categories, or users' personal information are not available. As long as significant progress has been made in algorithm's modeling in this area with the development of state-of-the-art models, it would be good to see how these models could produce better results when feeding these data. For example, "how a hybrid model can make user profiling based on click sequences", and similar questions could be triggered. Also, the contribution of the "pre-user-profiling" method when more attributes are available, i.e., product information, is worth-investigating. From all these, better service customization could be made through which both marketers and customers would benefit.

References

Adeniyi, David Adedayo, Zhaoqiang Wei, and Y Yongquan. 2016. Automated web usage data mining and recommendation system using k-nearest neighbor (knn) classification method. *Applied Computing and Informatics*, 12(1):90–108.

AlDaweesh, Sarah A. 2019. Predicting hourly particulate matter (pm 2.5) concentrations using meteorological data. In *2019 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, pages 136–140, IEEE.

Alswiti, Wedyan, Ja'far Alqatawna, Bashar Al-Shboul, Hossam Faris, and Heba Hakh. 2016. Users profiling using clickstream data analysis and classification. In *2016 Cybersecurity and Cyberforensics Conference (CCC)*, pages 96–99, IEEE.

Anshari, Muhammad, Mohammad Nabil Almunawar, Syamimi Ariff Lim, and Abdullah Al-Mudimigh. 2019. Customer relationship management and big data enabled: Personalization & customization of services. *Applied Computing and Informatics*, 15(2):94–101.

Azad, Armin, Mehran Manoochehri, Hamed Kashi, Saeed Farzin, Hojat Karami, Vahid Nourani, and Jalal Shiri. 2019. Comparative evaluation of intelligent algorithms to improve adaptive neuro-fuzzy inference system performance in precipitation modelling. *Journal of hydrology*, 571:214–224.

Balabantaray, Rakesh Chandra, Chandrali Sarma, and Monica Jha. 2015. Document clustering using k-means and k-medoids. *arXiv preprint arXiv:1502.07938*.

Beránek, Ladislav, Václav Nỳdl, and Radim Remeš3. 2016. Click stream data analysis for online fraud detection in e-commerce. In *The International Scientific Conference INPROFORUM*.

Biernacki, Christophe, Gilles Celeux, and Gérard Govaert. 2000. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*, 22(7):719–725.

Bigon, Luca, Giovanni Cassani, Ciro Greco, Lucas Lacasa, Mattia Pavoni, Andrea Polonioli, and Jacopo Tagliabue. 2019. Prediction is very hard, especially about conversion. predicting user purchases from clickstream data in fashion e-commerce. *arXiv preprint arXiv:1907.00400*.

Bogner, K, F Pappenberger, HL Cloke, and C de Michele. 2012. The normal quantile transformation and its application in a flood forecasting system. *Hydrology & Earth System Sciences*, 16(4).

Bro, Rasmus and Age K Smilde. 2014. Principal component analysis. *Analytical Methods*, 6(9):2812–2831.

Bruno, Giulia, Tania Cerquitelli, Silvia Chiusano, and Xin Xiao. 2014. A clustering-based approach to analyse examinations for diabetic patients. In *2014 IEEE International Conference on Healthcare Informatics*, pages 45–50, IEEE.

Cao, Jian, Qiang Fu, Qiang Li, and Dong Guo. 2017. Discovering hidden suspicious accounts in online social networks. *Information Sciences*, 394:123–140.

Chen, Long, Ziyu Guan, Qibin Xu, Qiong Zhang, Huan Sun, Guangyue Lu, and Deng Cai. 2020. Question-driven purchasing propensity analysis for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 35–42.

Chen, Lu and Qiang Su. 2013. Discovering user's interest at e-commerce site using clickstream data. In *2013 10th International Conference on Service Systems and Service Management*, pages 124–129, IEEE.

Cruz, FC, EF Simas Filho, MCS Albuquerque, IC Silva, CTT Farias, and LL Gouvêa. 2017. Efficient feature selection for neural network based detection of flaws in steel welded joints using ultrasound testing. *Ultrasonics*, 73:1–8.

Dhanachandra, Nameirakpam, Khumanthem Manglem, and Yambem Jina Chanu. 2015. Image segmentation using k-means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54:764–771.

Dollard, Morgan Francois Stephan. 2017. Media content advertisement system based on a ranking of a segment of the media content and user interest. US Patent App. 13/342,518.

ElBarawy, Yomna M, Ramadan F Mohamed, and Neveen I Ghali. 2014. Improving social network community detection using dbscan algorithm. In *2014 World Symposium on Computer Applications & Research (WSCAR)*, pages 1–6, IEEE.

Harsono, Tri, Achmad Basuki, et al. 2018. Cloud satellite image segmentation using meng hee heng k-means and dbscan clustering. In *2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, pages 367–371, IEEE.

Hartigan, John A and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

Hung, Wen-Liang and Shou-Jen Chang-Chien. 2017. Learning-based em algorithm for normal-inverse gaussian mixture model with application to extrasolar planets. *Journal of Applied Statistics*, 44(6):978–999.

Jain, Anil K. 2010. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.

Kapil, Shruti, Meenu Chawla, and Mohd Dilshad Ansari. 2016. On k-means data clustering algorithm with genetic algorithm. In *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pages 202–206, IEEE.

Kaufman, Leonard and Peter J Rousseeuw. 2009. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.

Kaur, Navjot, Jaspreet Kaur Sahiwal, and Navneet Kaur. 2012. Efficient k-means clustering algorithm using ranking method in data mining. *International Journal of Advanced Research in Computer Engineering & Technology*, 1(3):85–91.

Kazachuk, Maria, Alexander Kovalchuk, Igor Mashechkin, Igor Orpanen, Mikhail Petrovskiy, Ivan Popov, and Roman Zakliakov. 2016. One-class models for continuous authentication based on keystroke dynamics. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 416–425, Springer.

Kemp, S. 2019. Digital trends 2019: Every single stat you need to know about the internet. *Retrieved February*, 11:2019.

Kuyuk, HS, E Yildirim, E Dogan, and Gündüz Horasan. 2012. Application of k-means and gaussian mixture model for classification of seismic activities in istanbul. *Nonlinear Processes in Geophysics*, 19(4).

Li, Jing, Xinwei Zhang, Keqin Wang, Chen Zheng, Shurong Tong, and Benoit Eynard. 2019. A personalized requirement identifying model for design improvement based on user profiling. *AI EDAM*, pages 1–13.

Li, Kehua, Zhenjun Ma, Duane Robinson, and Jun Ma. 2018. Identification of typical building daily electricity usage profiles using gaussian mixture model-based clustering and hierarchical clustering. *Applied Energy*, 231:331–342.

Madke, Nilesh B, Mr Yogesh Kulkarni, Mr Rushikesh Mule, Mr Akash Lakade, and Mr Subodh Kulkarni. 2018. User profile based behavior identificaton using data mining technique.

McKinney, Wes. 2019. Pydata development team. pandas: powerful python data analysis toolkit.

Moraru, Luminita, Simona Moldovanu, Lucian Traian Dimitrievici, Nilanjan Dey, Amira S Ashour, Fuqian Shi, Simon James Fong, Salam Khan, and Anjan Biswas. 2019. Gaussian mixture model for texture characterization with application to brain dti images. *Journal of advanced research*, 16:15–23.

Nashar, Muhammad, Agustinus Hariadi DP, and Ryani D Parashakti. 2020. E commerce, economic growth and gross domestic product to the demand for delivery goods (case study on pt. pos indonesia tangerang). *International Journal of Organizational Innovation*, 12(3).

Nelli, Fabio. 2018. *Python data analytics: with pandas, numpy, and matplotlib*. Apress.

Ogbuabor, Godwin and FN Ugwoke. 2018. Clustering algorithm for a healthcare dataset using silhouette score value. *International Journal of Computer Science & Information Technology*, 10(2):27–37.

Oliphant, Travis E et al. 2019. Numpy: N-dimensional array package for python.

Olson, Eric. 2018. A study of the effects of histogram binning on the accuracy and precision of particle sizing measurements. *Pharmaceutical Technology*, page 29.

Ostrowski, Joao Gabriel and József Menyhárt. 2020. Statistical analysis of machinery variance by python. *Acta Polytechnica Hungarica*, 17(5).

Parikh, Vishal and Parth Shah. 2015. E-commerce recommendation system using association rule mining and clustering. *International Journal of Innovations & Advancement in Computer Science*, 4:148–155.

Patil, Preeti. 2019. Recent trends in consumer market and retail. *research journal of social sciences*, 10(7).

Pedregosa, F, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, et al. 2019. Scikit-learn: machine learning in python. 2011. *Mach. Learn. Res*, 12.

Pirrone, Roberto, Vincenzo Cannella, Sergio Monteleone, and Gabriella Giordano. 2018. Linear density-based clustering with a discrete density model. *arXiv preprint arXiv:1807.08158*.

Rao, Rajinder Singh and Jyoti Arora. 2017. A survey on methods used in web usage mining. *International Research Journal of Engineering and Technology (IRJET)*, 4(5):2627–2631.

Roberts, Wade, Gustavious P Williams, Elise Jackson, E James Nelson, and Daniel P Ames. 2018. Hydrostats: A python package for characterizing errors between observed and predicted time series. *Hydrology*, 5(4):66.

Ruan, Xin, Zhenyu Wu, Haining Wang, and Sushil Jajodia. 2015. Profiling online social behaviors for compromised account detection. *IEEE transactions on information forensics and security*, 11(1):176–187.

Sander, Jörg, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. 1998. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data mining and knowledge discovery*, 2(2):169–194.

Satopaa, Ville, Jeannie Albrecht, David Irwin, and Barath Raghavan. 2011. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pages 166–171, IEEE.

Savyan, PV and S Mary Saira Bhanu. 2017. Behaviour profiling of reactions in facebook posts for anomaly detection. In *2017 Ninth International Conference on Advanced Computing (ICoAC)*, pages 220–226, IEEE.

Sharma, Arvind, RK Gupta, and Akhilesh Tiwari. 2016. Improved density based spatial clustering of applications of noise clustering algorithm for knowledge discovery in spatial data. *Mathematical Problems in Engineering*, 2016.

Sitompul, Bernad Jumadi Dehotman, Opim Salim Sitompul, and Poltak Sihombing. 2019. Enhancement clustering evaluation result of davies-bouldin index with determining initial centroid of k-means algorithm. In *Journal of Physics: Conference Series*, volume 1235, page 012015, IOP Publishing.

Souiden, Nizar, Riadh Ladhari, and Nour-Eddine Chiadmi. 2019. New trends in retailing and services.

Su, Qiang and Lu Chen. 2015. A method for discovering clusters of e-commerce interest patterns using click-stream data. *electronic commerce research and applications*, 14(1):1–13.

Tabassum, Hira, Umar Shoaib, and M Shahzad Sarfarz. 2016. Tools and techniques for predicting user browsing behavior by using web usage mining. *International Journal of Computer Science and Information Security*, 14(11):519.

Talakokkula, Anitha. 2015. A survey on web usage mining, applications and tools. *Computer Engineering and Intelligent Systems*, 6(2):22–29.

Viswanath, Bimal, M Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. 2014. Towards detecting anomalous user behavior in online social networks. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 223–238.

Wang, Gang, Tristan Konolige, Christo Wilson, Xiao Wang, Haitao Zheng, and Ben Y Zhao. 2013. You are how you click: Clickstream analysis for sybil detection. In *Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13)*, pages 241–256.

Wang, Gang, Xinyi Zhang, Shiliang Tang, Christo Wilson, Haitao Zheng, and Ben Y Zhao. 2017. Clickstream user behavior models. *ACM Transactions on the Web (TWEB)*, 11(4):1–37.

Wang, Gang, Xinyi Zhang, Shiliang Tang, Haitao Zheng, and Ben Y Zhao. 2016. Unsupervised clickstream clustering for user behavior analysis. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 225–236.

Wang, Haibo, Da Huo, Jun Huang, Yaquan Xu, Lixia Yan, Wei Sun, and Xianglu Li. 2010. An approach for improving k-means algorithm on market segmentation. In *2010 International Conference on System Science and Engineering*, pages 368–372, IEEE.

Weller, Tobias. 2018. Compromised account detection based on clickstream data. In *Companion Proceedings of the The Web Conference 2018*, pages 819–823.

Yu, Dong and Li Deng. 2016. *AUTOMATIC SPEECH RECOGNITION*. Springer.

Zaim, Houda, Adil Haddi, and Mohammed Ramdani. 2019. A novel approach to dynamic profiling of e-customers considering clickstream data and online reviews. *International Journal of Electrical & Computer Engineering (2088-8708)*, 9(1).

Zumstein, Darius and Wolfgang Kotowski. 2020. Success factors of e-commerce–drivers of the conversion rate and basket value. In *Proceedings of the 18th International Conference e-Society 2020*, pages 43–50, IADIS Press.