NOVA IMS Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Unsupervised Learning Applied to the Segmentation of Users of Online Gambling Platforms in Portugal

The effects of the Covid-19 Pandemic on User Behavior and Segmentation

Leonardo Motta Perazzo Lannes

Project Work presented as partial requirement for obtaining the Master's degree in Advanced Analytics

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

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

UNSUPERVISED LEARNING APPLIED TO THE SEGMENTATION OF USERS OF ONLINE GAMBLING PLATFORMS IN PORTUGAL

The Effects of the Covid-19 Pandemic on User Behavior and Segmentation

by

Leonardo Motta Perazzo Lannes

Project Work presented as partial requirement for obtaining the Master's degree in Advanced Analytics

Advisor: Mauro Castelli, Ph.D.

Co Advisor*:* Fernando Peres

November 2021

DEDICATION

I would like to dedicate this thesis to my wonderful family. To my wife and daughter, who were always so supportive and understanding during the time in which I was committed to my studies. And to my mother and father, who have always encouraged me to pursue my objectives in life. Thank you.

ACKNOWLEDGEMENTS

I would like to thank my thesis advisers, Fernando Peres, and Mauro Castelli. Both were always available when I needed guidance and constantly helped me with interesting insights to improve this work.

ABSTRACT

Online gambling has become an increasingly relevant activity in the last years and is now available through a wide variety of technologies and platforms. This can be seen as an important addition to the entertainment industry since it has the potential of generating great economic impacts. The phenomenon, however, is not free of concerns considering that, like in any other type of gambling activities, online gamblers are susceptible to developing behavioral addiction. This has become a reason of concern to many governmental bodies around the world which are studying this issue due to its social impacts on the population. In this context machine learning algorithms can be applied to understand the behavior of online gamblers and to identify the characteristics of gambling addiction. This work project has the objective of segmentizing users of online gambling platforms in Portugal according to the tendency of these users to have compulsive gambling behavior. It also intends to evaluate the impacts of the Covid-19 pandemic on online gambling addiction by analyzing changes in user segmentation during the initial periods of the pandemic. This will be done by applying unsupervised learning algorithms, specifically K-Means and Self-Organizing Maps and by comparing user clusters from the years 2019 and 2020.

KEYWORDS

Artificial Intelligence; Big Data; Clustering; Data Science; Machine Learning; Segmentation; Unsupervised Learning

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
ANOVA	Analysis of Variance
BMU	Best Matching Unit
CSV	Comma-Separated Values
FCT	Fundação para a Ciência e a Tecnologia
IQR	Interquartile Range
ML	Machine Learning
PAM	Partition Around Medoids
SOM	Self-Organizing Maps
SRIJ	Serviço de Regulação e Inspeção de Jogos

1. INTRODUCTION

Online gambling has become increasingly popular throughout the last years and has become a relevant alternative for those who wish to engage in gambling activities as a form of entertainment. The main factor that has allowed the growth of this activity is the accelerated pace that has marked the development of online technologies that provide these gambling services through diverse distribution channels (European Comission, 2012).

This phenomenon, however, is not free of concerns and currently the most likely candidate to join gambling disorder as a behavioral addiction is internet gambling disorder (Clark, 2014). This has come to the attention of the European Commission, and they have called for specific measures to prevent online gamblers from undertaking addictive behaviors.

In Portugal the job of supervising and controlling online gambling platforms belongs to SRIJ ("Serviço de Regulação e Inspeção de Jogos") which can be described as a Gambling Inspection and Regulation Service. This organization is controlled by "Turismo de Portugal", the national tourism authority in the country which is affiliated with the Ministry of Economy. Having this task, SRIJ receives daily data from all the entities that are authorized to deliver online gambling services to Portuguese citizens (SRIJ, 2018). This data, among other things, contains detailed records of player activities in each of the types of games that are considered legitimate and represent a valuable source of information in the effort of applying responsible and healthy gaming policies.

Given this context, it becomes relevant to analyze this data with these concerns of gambling addiction in mind. And since there is a need to guarantee that this important entertainment industry will comply with government and social demands it becomes imperative to apply machine learning techniques in order to have a better understanding of user behavior. This is necessary due to the amount of data delivered by the gambling entities: this amount of data would be impossible to be processed manually and would not generate great insights with standard data processing tools. Machine Learning (ML) techniques can be of utmost importance when dealing with huge amounts of data and when there is a need to discover patterns that are not obviously apparent (Géron, 2019).

Specifically for the task of understanding user behavior and identifying negative patterns of gambling habits, it seems necessary to apply Unsupervised Machine Learning techniques. In this type of approach, the task of the ML tools is to identify hidden patterns in the data. As a result, the data is divided into groups based on metrics of similarity (Patel, 2019).

As Unsupervised Learning techniques are applied to data and patterns are identified, the data can be divided into groups or clusters. And since these clusters represent records with similar characteristics, in the context of online gambling, they can be helpful to identify different types of user behaviors. Therefore, it becomes less arduous to quantify and describe the aspects of gambling that can determine pathological behaviors or that could lead to them.

1.1. OBJECTIVES

The objective of this thesis is to analyze the behavior of online gambling users in Portugal and the possible impacts that the Covid-19 pandemic may have had in these behaviors. The main focus is to analyze these patterns of behavior with the concerns of pathological gambling in mind.

In the past, it has been reported that gambling disorders may occur in between 0.1 and 5.8% of the population across different countries (Calado & Griffiths, 2016). And since the pandemic has raised many issues related to health, specifically those related to mental health (Håkansson, 2020), it is pertinent to evaluate how much effect, if any, it had on problematic gambling.

This will be done by applying ML and Unsupervised Learning techniques to segmentize user data from 2019 and 2020, specifically during the initial and more severe periods of the pandemic and of lockdown restrictions. It will also be done by comparing player behaviors from one year to the other. As these techniques are applied it will be possible to identify clusters of user behaviors and patterns of gambling practices.

In Portugal, online gambling activities are divided into two groups: Sporting Bets and Games of Chance. The latter is composed of 4 types of games, which are: Slot machines, Poker, Roulette, and French Bank. During the fourth trimester of 2019 the total value of the bets made through games of chance was around 852 million Euros. And of this total, close to 70% refers to Slot Machines (Turismo de Portugal, 2019). Given its relevance, the work of this thesis will be executed with data from Slot Machine gambling delivered by the 11 entities that were authorized to operate during the period of interest.

1.2. RESEARCH QUESTIONS

After performing this analysis, it will be possible to answer the following list of research questions that can help determine and quantify the impacts of the pandemic on negative or excessive gambling habits:

1. **Is it possible to identify specific user behaviors (clusters) based on the data from 2019 that can establish online gambling disorders?**

2. **Also based on the data from 2019, is it possible to identify a cluster of users that could be associated with online gambling problems?**

3. **By comparing the segmentation results from 2019 and 2020 during the lockdown restrictions period, has there been a significant shift in cluster sizes?**

4. **Did players, on average, engage in more intense gambling activities during the lockdown restriction periods?**

5. **During the lockdown restrictions period, is it possible to determine if a certain number of players from other user clusters migrated to clusters of users with potential gambling disorders?**

1.3. STRUCTURE

To achieve the objectives of this thesis and to answer the research questions that have been postulated, this project has been structured into sections that will lead to these goals. These sections are described as follows:

A. Theoretical Background: in this section, the main concepts that will be applied during this thesis will be covered. This includes broad definitions of Artificial Intelligence and Machine Learning and specific definitions of data preprocessing, distance metrics, clustering techniques, and model evaluation.

B. Cluster Analysis: this section will contemplate, initially, all the steps regarding the preparation of the data that has been made available for this project. This includes outlier treatment and feature engineering, among others. Following this step, the selected clustering techniques (those that are considered more appropriate for the problem at hand) will be applied to the data from 2019 in order to identify user clusters. Finally, the same clustering techniques will be applied to the data from 2020 and the results from both years will be displayed.

C. Results and Discussion: the clustering results and appropriate comparisons between user behavior in each year will be made. Other than the differences regarding user segmentation, a few more high-level statistical comparisons will be conducted. As a conclusion to this project, and to achieve the objectives that have been established, the research questions will be answered. This will be done objectively with the results gathered in the previous section.

D. Recommendations for Future Work: in this last section, suggestions of projects related to this thesis and that could build upon the knowledge that has been gained, will be listed. Also, opportunities that have been identified to improve the execution of this project will be mentioned so they can be applied to similar projects.

2. THEORETICAL BACKGROUND

2.1. ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Although it is not possible to find a consensus in a unique definition of Artificial Intelligence, it can be thought of as the capacity of computers to perform tasks associated to intelligent beings. It is generally associated with the development of systems, or simply algorithms that display determined characteristics such as the ability to discover meaning, generalize or learn from past experience (Britannica).

Machine Learning refers to a ramification of the Artificial Intelligence concept that deals with the concept of learning from experience gained from data collected from previous events (Géron, 2019). Machine Learning methods can be applied to different types of databases to learn patterns that are implicit in the data and that can be used to make predictions or segmentize records into groups with similar attributes (Mitchell, 1997). In other terms ML methods invert the traditional order of how algorithms work in which they receive an input and generate an output. With ML the output (training data) is used to learn and generate an algorithm (Domingos, 2015). Additionally, ML algorithms can be divided into supervised and unsupervised, depending on whether they are trained with human supervision, i.e., whether or not the training data available for learning includes the desired results (Géron, 2019).

2.1.1. Supervised Learning

In supervised learning the training data used by the algorithm includes the desired results, which are called labels (Géron, 2019). This means the data used by the algorithm refers to some kind of problem that has been previously mapped or calculated and for which there are predefined classes or values. Based on this data, the algorithm generates a model that is capable of predicting the results of future instances of the same problem. The two most common types of supervised learning applications are classification and regression. In the former, the aim is to assign each record or instance of the data to a finite number of discrete categories. In the latter, the objective consists of predicting one or more continuous variables (Bishop, 2006). An example of a classification problem would be the detection of spam emails. And an example of a regression problem would be the prediction of the sale value of a house.

2.1.2. Unsupervised Learning

This type of machine learning technique is the equivalent of clustering, that is, to group items with similar traits into distinct bins. It is considered unsupervised since the training data available for the learning process is not previously labeled. Thus, it is employed, typically, when the objective is to discover classes within the data. The major issue regarding unsupervised learning is the fact that the learned model cannot give any semantic meaning to the clusters that have been found (Han, Kamber, & Pei, 2012). Once the model has identified the clusters and the values of the variables that define them, the analysis of an expert is necessary for the purposes of interpretation and even to validate if there is any realistic meaning to what has been determined by the model. One of the most common

examples of unsupervised learning applications is the segmentation of customers groups for the purposes of elaborating customized marketing campaigns or to preview customer CHURN rates.

For the objectives of this thesis, unsupervised learning techniques will be implemented in order to identify groups of users with similar attributes. These clusters will then be analyzed with the intention of identifying evident traces of gambling behaviors.

2.1.3. Semi-supervised Learning

Semi-supervised learning algorithms can be applied when the training data is partially labeled, generally with a great deal of unlabeled data and a small amount of labeled data. Ordinarily, semi-supervised techniques are a combination of unsupervised and supervised algorithms with the data being trained sequentially on each of them (Géron, 2019).

2.1.4. Reinforcement Learning

In reinforcement learning the learning algorithm does not have the labels for each of the records in the training data. In contrast to what happens in supervised learning, the algorithm learns through a process of trial and error (Bishop, 2006). It is based on a concept of rewards (positive or negative), in which better rewards are granted when the algorithm generates better results (Domingos, 2015). This "encourages" the algorithm to continue its optimization process until it receives higher rewards.

2.2. BIG DATA

The implementation of ML methods requires the availability of sufficient data for the algorithms to identify patterns. During the greater part of computer science history, the emphasis was to develop sophisticated and optimized algorithms. The major difficulties during this period were to obtain enough computational power to execute such algorithms and the lack of data to train on. Nowadays, computational power has become more accessible, even to the average user. And recent work in AI, indicates that for many problems it is more relevant to have large amounts of data than to worry about the algorithm to be implemented (Russell & Norvig, 2010). The term Big Data has been created exactly for this reason, since we currently live in what is known as the data age. It defines the large volumes of data that are generated daily by a great variety of sources and of different types and formats, such as text files, images, and videos, either structured or unstructured (White, 2015). This has allowed the implementation of machine learning algorithms and the achievement of more conclusive results for a wide variety of problems and fields of study, such as customer segmentation, disease prediction, image classification among many others.

2.3. DATA PREPARATION AND PREPROCESSING

For the execution of any data related projected it is pivotal to perform certain procedures that will guarantee its quality. The quality of data can be measured by numerous factors such as accuracy, completeness, and consistency, all of which can be affected by faulty collection instruments, human or computer errors, technology limitations or unavailability (Han, Kamber, & Pei, 2012). Besides the procedures to improve data quality, other actions might be necessary to ensure data usability such as standardization and dimensionality reduction.

2.3.1. Outlier Treatment

Outliers can be thought of as observations with significantly deviating characteristics (Madsen, 2018). In a ML model, if these observations are very extreme, they might interfere inappropriately with the results generated by the model. In an unsupervised learning model, for instance, outliers can generate individual clusters for only an isolated group of observations. In supervised learning models, outliers can cause severe deviations in the model's predictions. Thus, it is important to perform some sort of outlier treatment to ensure the data is appropriate for training a model.

Since the definition of what is a significant deviation is subjective, there are a few methods to deal with outlier detection. Depending on what the assumptions are, regarding the definition of outliers versus the rest of the data, outlier detection can be classified in two types: proximity-based methods, and statistical methods (Han, Kamber, & Pei, 2012). The former, considers objects as outliers when their locality is sparsely populated (Aggarwal, 2017). The latter, applies simple and powerful statistical techniques for the screening of outliers (Alam, 2020). For the purposes of this project, considering that it will be more relevant to identify extreme values, statistical methods will be discussed at more length and two specific methods will be detailed.

The Z-Score is a parametric outlier detection method that assumes a gaussian distribution of the data and measures how distant the observation is from the mean (measured in standard deviations) (Widmann & Heine, 2018). In this method values that have a Z-Score higher or lower than predetermined limits are considered outliers. The Z-Score is calculated with the following formula:

$$Z\text{-}score = \frac{x - mean}{Standard\ Deviation}$$

The Boxplot method is non-parametric statistical method for identifying outliers, since it does not assume the data has a normal distribution. It explores the concept of interquartile range (IQR) which is the difference between the 1st quartile (25th percentile) and 3rd quartile (75th percentile) of the data (Alam, 2020). Values greater than the upper bound or smaller than the lower bound are considered outliers. These boundaries can be calculated with the following formulas:

upper bound = Q3 + 1,5 * IQR

lower bound = Q1 – 1,5* IQR

Where Q1 and Q3 represent the values of the 1st and 3rd quartiles of the data and IQR represents the difference between Q1 and Q3.

The Boxplot method is considered the default method to identify outliers in univariate data (Madsen, 2018). It has a graphical representation that helps understanding these concepts and its boundaries. This can be visualized in Figure 2.1.

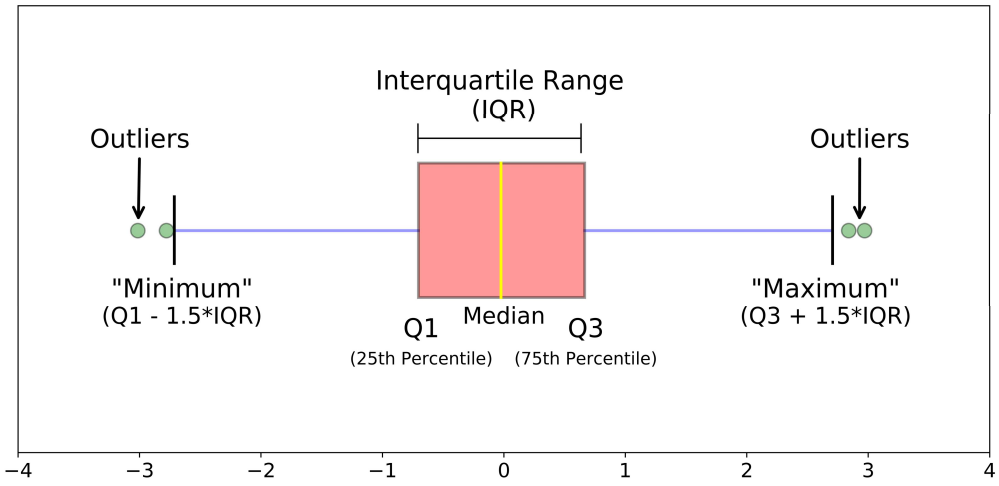

Figure 2. 1: Boxplot
(Galarnyk, 2018)

Identifying the outliers, however, is just the first step in outlier treatment. The next step is deciding what to do with these outliers. And in this case, there are three main possibilities to consider (Birkett, 2019):

1. Keep the outliers.

2. Remove the outliers.

3. Change their values to something that better represents the data, such as trimming extreme values and replacing them by percentiles (Birkett, 2019).

The choice of one of these alternatives may greatly impact a project and the decision must take into consideration the specificities of the data at hand.

2.3.2. Missing Values

Sometimes the data that has been made available for analysis has some missing values and these could be present in just a few observations or, even, in all the observations of the dataset. This anomaly can be present in one or more features and may cause trouble since some machine learning methods do not work with missing features (Géron, 2019). Other methods might generate inconsistent results if the proportion of data with missing values is extremely elevated.

To deal with this problem there are different alternatives:

1. Delete all the records (observations) of the dataset that have missing values in at least one of their attributes.

2. Delete all the attributes in the dataset that have missing values in at least one of the observations.

3. Impute the missing values present in the dataset according to some sort of criterion. In this case the same criterion can be applied to all attributes, or it can be analyzed case by case, in order to apply the appropriate changes for each one.

With respect to this last alternative, there are some possibilities in which missing values can be imputed. Firstly, it is possible to fill in missing values using statistical properties such as mean, median or mode. It is also possible to fill in missing values by applying algorithms that search for the records that are closest to the one that has missing values. And finally, regression can be applied by considering the missing values as dependent variables.

2.3.3. Rescale Data

It is common for the various attributes of a dataset to be in different scales. However, most machine learning algorithms work better if all features are in the same scale. Otherwise, the algorithm could consider a few of the attributes as more relevant to the problem just because it has naturally higher numbers than some of the other features when, in fact, sometimes it could be quite the opposite.

For this reason, there are some alternatives through which data can be rescaled.

1. Normalization: rescales all the attributes to a range between 0 and 1 (Brownlee, 2016). The following formula defines this kind of transformation:

$$X' = \frac{X - X_{min}}{X_{\max} - X_{min}}$$

   Where X_{min} and X_{max} correspond to the minimum and maximum values encountered in the dataset.

2. Standardization: transforms attributes that have a Gaussian distribution into a standard Gaussian Distribution with a mean of 0 and a standard deviation of 1 (Brownlee, 2016). The following formula defines this kind of transformation:

$$X' = \frac{X - \mu}{\sigma}$$

Where μ is the mean of the values and σ is the standard deviation.

The choice of which type of data transformation to apply is not always simple. Normalization is usually better when the data does not follow a Gaussian Distribution and since Standardization does not have boundaries, it will not affect outliers. Anyhow, sometimes it is necessary to apply both transformations to the data and compare the performances (Bhandari, 2020).

2.3.4. Feature Selection

Frequently, the excess of features in a dataset is prejudicial to machine learning models. It is the opposite of what happens with the number of observations, where the algorithms benefit from having more data available. This problem is referred to as the Curse of Dimensionality, meaning that it can be necessary to perform feature selection in order retain only the most relevant features for the purposes of training a model (Géron, 2019).

The curse of dimensionality arises from the fact that irrelevant features can make it more difficult for the algorithms to identify interesting patterns since the data becomes sparser. These patterns would, otherwise, be more transparent in a lower dimensional space (Patel, 2019). Also relevant is the fact that an exaggerated number of features can make the algorithm unbearably computationally expensive.

Specifically for unsupervised learning techniques, the addition of more dimensions to the data can make every observation in the dataset seem equidistant from all the others, since most algorithms use distance measures to identify similarities in the data (Yiu, 2019). This makes it harder to identify any significant clusters in the data.

Figure 2.2 below is a simple example of this since it demonstrates that with only one dimension (one feature) it is very clear that the data can be divided into two clusters. However, when a new feature is considered, each of the data points seem to be isolated from the others and form a cluster of their own.

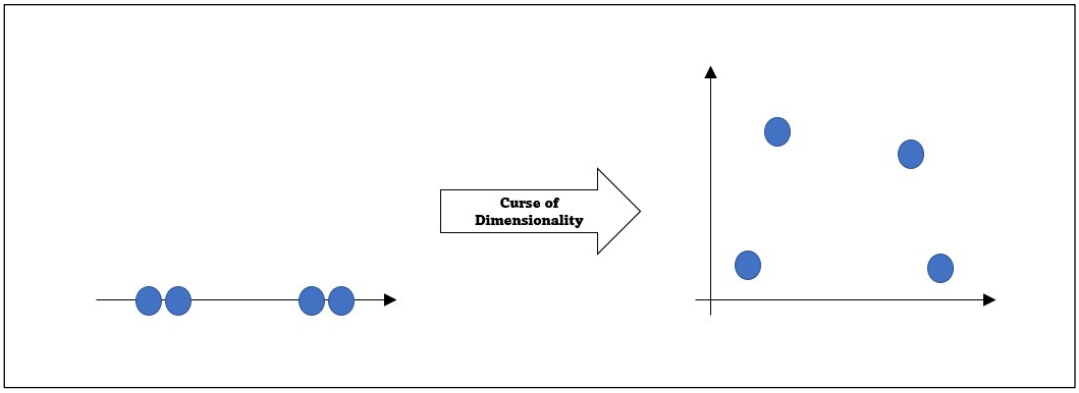

Figure 2. 2: The Curse of Dimensionality

One method of evaluating attributes that can be disregarded is to consider the correlation between each of the features of the dataset. Attributes can be considered redundant if they can be derived from another set of attributes (Han, Kamber, & Pei, 2012). If a dataset has a great number of correlated attributes, ML algorithms that are applied to it might suffer bias towards those attributes and they could possibly end up contributing more to the results than what would be appropriate.

However, in order to perform this type of evaluation it is necessary to quantify the correlations between each pair of variables. This process must take into consideration the types of attributes that are present in the dataset. For the purposes of this project only two types of attributes will be considered: categorical and continuous values. The former, represents values that have a finite number of distinct categories such as education level or gender and can be represented either by words or numbers. The latter, represents values that can have an infinite number of values such as salary or price.

When analyzing correlation between attributes, different approaches must be applied, depending on the types of each one.

- Correlation between two categorical attributes: A statistical test named Chi-squared test is applied to identify similarities or differences (Kumar, 2021)

- Correlation between continuous and categorical attributes: If the categorical attribute is binary, a statistical test named T-test must be applied. If the categorical variable can have more than two possible values, then an ANOVA (Analysis of Variance) test must be applied (Kumar, 2021).

- Correlation between two continuous attributes: In this case, correlation statistical tests can be used. The most common techniques are Pearson Correlation and Spearman Rank Correlation, both of which define correlation values between [-1,1], where proximity to 1 denotes positive correlation and proximity to -1 denotes negative correlation (Kumar, 2021). A positive correlation indicates that when one attribute increases, so does the other. A negative correlation indicates that when one attribute increases, the other decreases. These correlation values can all be viewed simultaneously through what is referred to as a heatmap of correlation matrix, as showed in Figure 2.3, below. This type of visualization indicates

through the intensity of the colors how strong is the correlation between each pair of attributes.

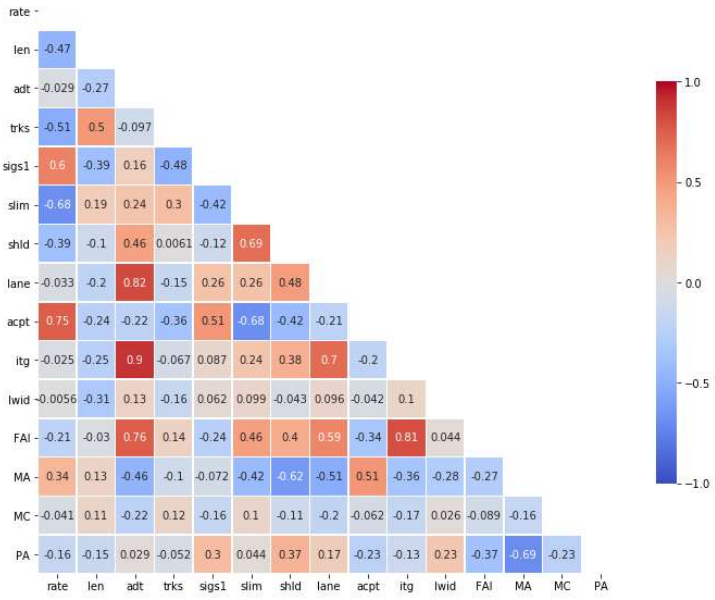

Figure 2. 3: Heatmap of Correlation Matrix
(Kho, 2019)

2.3.5. Feature Engineering

Feature engineering is a mechanism by which new attributes are created based on the ones that are already available. This is something that could be pivotal in some scenarios since machine learning algorithms are only as good as the features that enable it to separate the data points in the feature space (Patel, 2019).

There are many possibilities to consider in the creation of new attributes. Most of these options refer to the generation of categorical variables, such as in the following examples:

- The prefix of a person's name can be utilized to determine the gender of that person.
- City names can be grouped into regions or countries.
- Time of day can be used to segregate the data into periods of the day such as morning, afternoon, and night.
- Dates can be separated into weekdays and weekends.
- Continuous values can be aggregated into groups of delimited bins, each one representing a category.

Nonetheless, feature engineering can also be applied to create new continuous variables. One example would be the determination of the age or time of usage of an observation, based on a birthday date or any other raw date.

Finally, feature encoding is a process in which categorical attributes are transformed into numerical variables. This is necessary because most ML algorithms do not work well with categorical variables (Kumar, Feature Engineering — deep dive into Encoding and Binning techniques, 2020). As an example, in a dataset where one of the attributes represents gender, the words Male and Female would be substituted by the numbers 0 and 1.

2.4. CLUSTERING TECHNIQUES

Clustering techniques have the objective of classifying observations or data items into groups (clusters) based on similarities (Jain, Murty, & Flynn, 1999). It is useful in exploratory pattern analysis and ML situations that involve segmentation and pattern classification (Jain, Murty, & Flynn, 1999). As such, these techniques are appropriate for unsupervised learning problems, that is, situations in which the data has not been previously labeled.

There are a great variety of clustering techniques that have been developed and tested. The choice of the best technique for a specific problem depends on the characteristics of the problem and of the data to be used. Three common clustering methods will be discussed in this section: Hierarchical Methods, Partitional Methods and Self-Organizing Maps.

2.4.1. Hierarchical Methods

Hierarchical methods of data clustering separate the data into groups at different levels that represent a hierarchy or a "tree" of clusters. This type of technique is useful for data summarization and visualization since each cluster can be divided into smaller subgroups or grouped up with similar groups (Han, Kamber, & Pei, 2012).

The implementation of hierarchical clustering does not require a previous definition of the number of clusters the data should be divided into. This type of method works by building a dendrogram, an upside-down tree, in which the leaves, at the bottom, represent the individual instances of a dataset. An iterative process joins the leaves as it moves vertically up the tree, based on how similar they are to each other and eventually all the instances will be linked together (Patel, 2019) as can be seen on Figure 2.4 below.

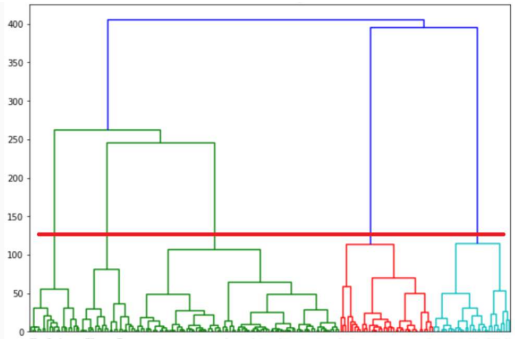

Figure 2. 4: Dendrogram

Since hierarchical clustering does not have a predefined number of clusters, before the implementation of the method, this decision can be made based on the dendrogram. Just by visually inspecting the dendrogram one can have an intuition of how the data is distributed. In the above example, in Figure 2.4, a horizontal red line symbolizes an assumption of how the clusters could be defined. In this case, the red line crosses five vertical lines and each of these vertical lines represent one cluster. As this red line moves higher up the tree this would generate a smaller number of clusters but with more elements is each of those clusters. On the contrary, if the red line moved to lower positions in the tree, it would generate a greater number of clusters but with less elements in each one.

The implementation of hierarchical clustering can be executed in two forms:

- Divisive Clustering: this is a top-down technique in which, initially, all points of the dataset belong to the same cluster. A recursive implementation gradually splits the datapoints as one moves down the hierarchy (Kumar, 2020).
- Agglomerative Clustering: this is a bottom-up technique in which, initially, each data point represents a cluster. The implementation of this technique groups these points into clusters as one moves up the hierarchy (Kumar, 2020).

One important aspect of hierarchical methods is the definition of how to group or separate data points into clusters. Whether using agglomerative or divisive clustering, it is necessary to measure the distance between two clusters (Han, Kamber, & Pei, 2012). This is done based on how close the data points are to each other. Consequently, points that are close to each other tend to be a part of the same cluster and points that are very far apart tend to belong to different clusters. There are a few different metrics that can be applied to evaluate proximity between data points, but the most common are the following four:

- Single-Linkage: this metric uses the distance between the two closest neighbors of different clusters in order to group data points into new clusters. It can generate clusters that are very spread-out and where observations in different clusters are closer to each other than observations within their own cluster (Clements, 2019).
- Complete-Linkage: this metric considers the distance between the two farthest neighbors of each cluster to determine how clusters will be grouped to form new clusters. Usually, the final clusters are very close to each other (Clements, 2019).
- Average-Linkage: this metric uses the average distance between each pair of observations of every combination of two clusters to determine which clusters are closer to one another (Clements, 2019).
- Centroid-Linkage: this metric is based on the distance between the centroids of two clusters (Clements, 2019).

Regardless of which metric is used to evaluate proximity between data points, it is essential to define a distance metric to be applied since this is what will be used to calculate the proximity matrix and define distance between observations (LZP, 2019).

Also relevant is the fact that, with any of the metrics described above, the implementation of hierarchical methods can be terminated when a maximum distance between nearest clusters exceeds a predefined threshold (Han, Kamber, & Pei, 2012).

To summarize, Hierarchical methods have the advantage of being one of the easiest to understand in comparison to other methods. It is also relatively simple to implement and generates an appealing output which is the dendrogram. It does not, however, usually provide the best solutions, given the following factors (Bock):

- It does not work with missing data.
- It does not work well with categorical values.
- It does not work well with large volumes of data. For large datasets the construction of a dendrogram is computationally prohibitive (Jain, Murty, & Flynn, 1999).

2.4.2. Partitional Methods

Partitioning techniques organize the objects of a set into a predefined number of exclusive groups or clusters. These clusters are formed by optimizing a partitioning criterion such as a dissimilarity function based on distance, so that objects within a cluster are "similar" to each other and "dissimilar" to objects of other clusters (Han, Kamber, & Pei, 2012).

The most commonly utilized partitioning method is the K-Means algorithm, which separates the data into clusters of equal variances and minimizes a criterion known as inertia or within-cluster sum-of-squares, as is represented by the formula bellow (sklearn.cluster.Kmeans - scikit-learn 1.0.1 documentation):

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

Where (x_i) represents each of the observations in the cluster and (μ_j) represents the mean of the cluster, which is also known as the centroid.

As the formula shows, the objective of K-Means algorithm is to execute iteratively, by dividing the observations into clusters, until the distances of each observation to its centroid are minimal. It can be intuitive to conclude that a big number of clusters would naturally reduce the intra-cluster distances. However, one of the most important aspects of K-Means is exactly the definition of the number of clusters in which the data will be divided. The number of clusters should ideally illustrate the groups that represent the data and, since this can be a very subjective matter, there are a few tools to help with this judgment.

Basically K-Means algorithm can be implemented in the following five steps:

1. Step 1: Choose the number of clusters (K).
2. Step 2: Select K random points as centroids.
3. Step 3: Assign each observation in the dataset to one of the centroids, the closest one.
4. Step 4: Recompute the centroids of the newly formed clusters
5. Step 5: Repeat steps 3 and 4 until a stopping criterion is achieved.

The computation of the new centroids after each iteration is calculated as the mean (average of all attributes) of all the observations that belong to each cluster, i.e., all the observations that are associated to a centroid.

Regarding step 5, there are basically three stopping criteria that can be applied to interrupt the iteration (Sharma, 2019):

- The centroids of the newly formed clusters do not change from the previous iteration.
- All the data points remain in the same clusters as they were in the previous iteration.
- A maximum predefined number of iterations has been achieved.

Figure 2.5, below, shows the basic evolution of a K-Means algorithm execution.

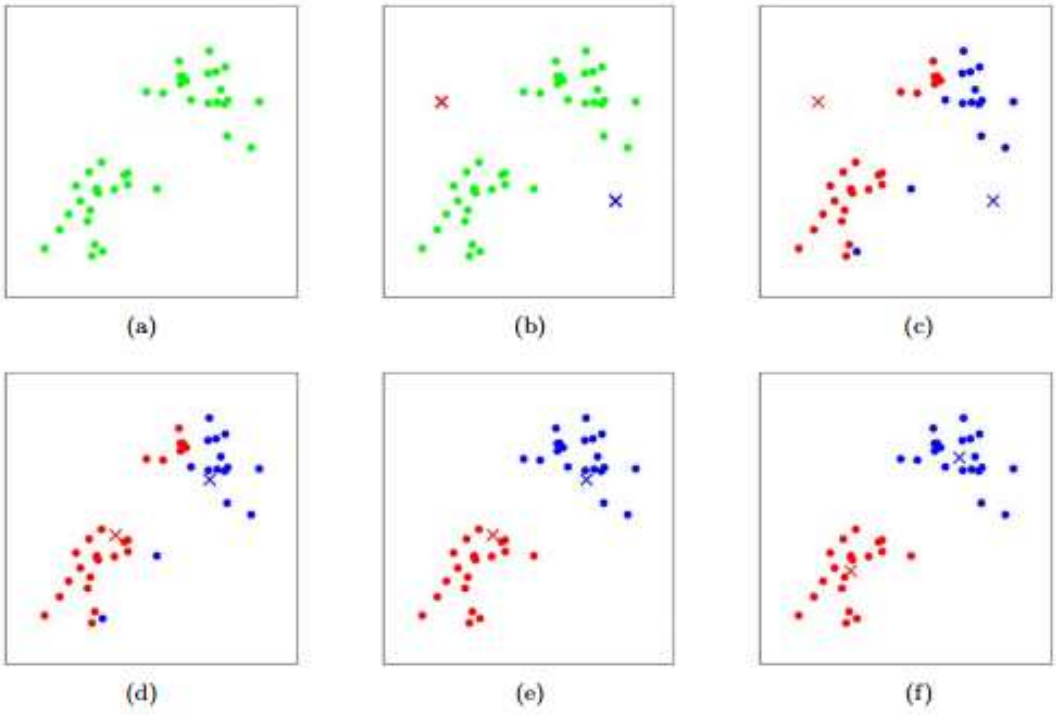

Figure 2. 5: K-Means Algorithm
(Piech, 2013)

In the graphical example above, image (a) shows the original dataset, which will be split in two clusters, i.e., in this case, K=2. Image (b) represents step 2 of the algorithm and shows the two random points (a red and a blue "X") that where selected as the initial centroids. Images (c), (d) and (e) represent steps 3 and 4 of the algorithm and show the data points being assigned to one of the centroids and the centroids being recomputed. Finally, image (f) represents the final status of the algorithm with the data points split into two clusters, each one represented by a color, blue and red.

One aspect of K-Means that needs to be highlighted is that there is a high degree of randomness in this method caused by the fact that the initial centroids are selected randomly. For this reason, it is recommended to perform several executions of the algorithm and add up the total intra-cluster distance for each execution. By the end of this process the execution that has the smallest intra-cluster distance should be selected. It should also be mentioned that the definition of the number of executions is, likewise, a source of randomness. But whatever is the selected number, this strategy will minimize the effects that a very poor initial centroid selection could cause in the algorithm. It just must be done wisely to avoid the waste of computational resources.

Another important aspect of K-Means is the selection of the number of clusters (K) into which the data will be clustered, since this is one of the main inputs of the algorithm. And considering that K is a basic input of the algorithm, the methods that are used to evaluate the most appropriate number of clusters involve multiple executions of the algorithm with different values of K.

There are two very common methods to evaluate the results of different numbers for K and to decide which one is better suited for the data that is being segmentized.

2.4.2.1. Inertia

Inertia measures how far away the points within a cluster are, i.e., it measures the total intra-cluster distance. The objective is to have a low value of inertia, which means most data points are close to their centroids (Amelia, 2018). This can be achieved if the number of clusters is very big, yet this would most likely generate an unreasonable segmentation. A solution that is frequently implemented to evaluate this trade-off is the elbow graph, shown in Figure 2.6 below.

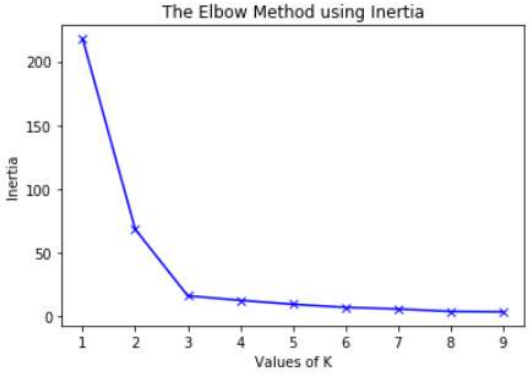

Figure 2. 6: Elbow Graph
(Gupta, 2021)

The elbow graph resembles the arm of a human being, and the elbow point indicates the number of clusters that should be used for the K-Means algorithm. Until that point the decrease in inertia is significant and from that point on the inertia starts decreasing in a linear fashion (Gupta, 2021). In the example above it is clear that the ideal number of clusters is three, since there is a great decrease in inertia until that number. A greater number of clusters will not reduce the inertia significantly and would likely not be beneficial for the segmentation of this data.

2.4.2.2. Silhouette Analysis

The silhouette analysis is used to identify the distance between the clusters that were generated after the execution of the partitional algorithm. It works by attributing a silhouette coefficient with a range of [-1,1] to each data point (Selecting the number of clusters with silhouette analysis on KMeans clustering - scikit-learn 1.0.1 documentation). Silhouette scores close to +1 indicate that the observations are far away from other clusters and silhouette scores close do -1 indicate that the observations are nearby other clusters and could have possibly been misclassified.

In order to evaluate the quality of a clustering method one can calculate the average silhouette score of all the datapoints. And if the objective is to compare different implementations of the same algorithm, one option could be to measure the average silhouette score of each implementation. That is why this is one of the recommendations of how to determine the ideal number of clusters of the K-Means algorithm, or to put it another way, how to determine the ideal K in K-Means.

An additional tool that helps in the silhouette Analysis is the silhouette plot, which gives a visual intuition of the proportion of data points that are well classified and that have silhouette scores above average. Figures 2.7 and 2.8 below show an example of this comparison between two K-Means implementations for sample data, where the first has K=2 and the second has K=6.

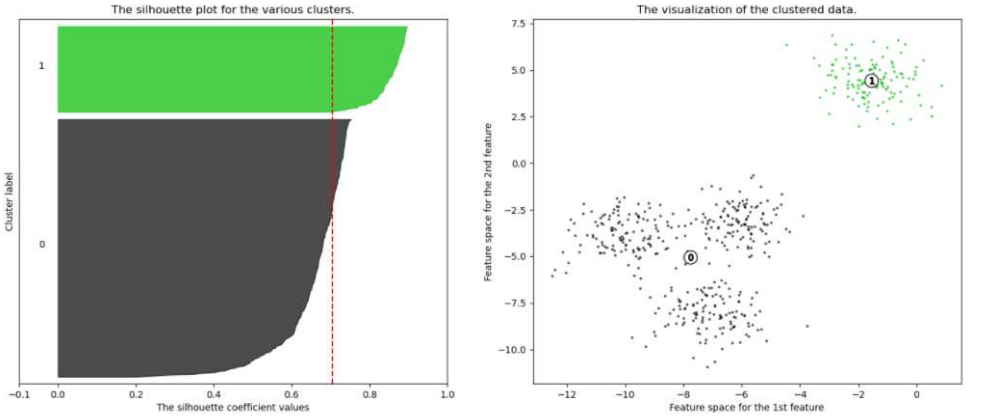

Figure 2. 7: Silhouette Analysis for K-Means on sample data with K=2
(Selecting the number of clusters with silhouette analysis on KMeans clustering - scikit-learn 1.0.1 documentation)

Figure 2.7 shows the silhouette analysis regarding the clustering of the sample data into two clusters. The image on the right of this figure shows the distribution of the data and how each data point was classified, either in cluster 0 (grey) or in cluster 1 (green). The image on the left shows the silhouette plot of this clustering option and a few conclusions can be inferred.

- The thickness of the silhouette plot indicates how large each cluster is, hence, cluster 0 has more observations than cluster 1.

- No observations have a negative silhouette score meaning probably no observations were incorrectly classified.

- The dashed red line indicates the average silhouette score of the algorithm which, in this case, is 0,70.

- Most data points in cluster 1 have a silhouette score above the average, meaning they are very distant from other clusters.

- Cluster 0 has some points that have silhouette scores below the average, but most have a silhouette score greater than 0,6 meaning they were probably well classified.

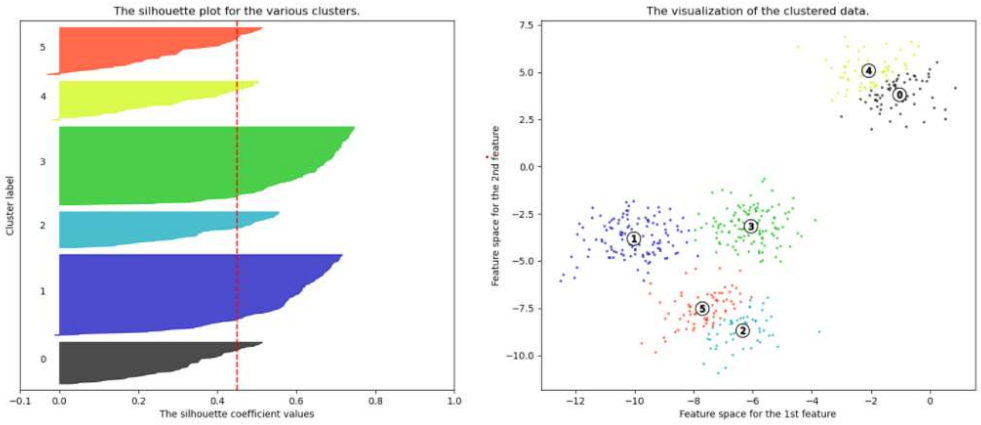

Figure 2. 8: Silhouette Analysis for K-Means on sample data with K=6
(Selecting the number of clusters with silhouette analysis on KMeans clustering - scikit-learn 1.0.1 documentation)

Figure 2.8 shows the silhouette analysis regarding the clustering of the sample data into six clusters. The image on the right of the figure shows the distribution of the data and how each data point was classified, either in cluster 0 (grey), cluster 1 (dark blue), cluster 2 (light blue), cluster 3 (green), cluster 4 (yellow) or in cluster 5 (red). The image on the left shows the silhouette plot of this clustering option and the following conclusions can be reached.

- The thickness of clusters 1 and 3 indicate they are the larger clusters. The other four clusters have about the same size.

- Clusters 1, 4 and 5 have some negative silhouette scores which indicates that a portion of the data points might have been incorrectly classified.

- The dashed red line indicates the average silhouette score of the algorithm which, in this case, is 0,45.

- Most data points in clusters 1 and 3 have a silhouette score above the average, meaning they are distant from other clusters.

- Clusters 0, 2, 4 and 5 have many data points that have silhouette scores below the average, meaning the classification is not very reliable.

By comparing these two implementations through the silhouette analysis it becomes clear that the algorithm generated better clusters when k=2, basically because it generated clusters that are further apart and that can be clearly identified and differentiated.

2.4.3. Variations of K-Means

K-Means is one the most well-known partitional method algorithms for clustering. It is also the most commonly used. It has, however, been modified throughout the years in order to adapt to some of its weaknesses and in order to generate better results is some specific situations. Three of these variations will be presented in this work.

2.4.3.1. K-Medians

K-Medians is a variation of K-Means that utilizes the median of each variable, instead of the mean, to determine the central point in the cluster. This is a form of avoiding the vulnerability that the mean has towards outliers, since even one drastic outlier can pull the value of the mean away from the majority and create inadequate clusters (Whelan, Harrell, & Wang, 2015).

2.4.3.2. K-Medoids

K-Medoids is also known as the PAM (partition around medoids) algorithm. It differs from the previous versions of the partitioning algorithms in the sense that each cluster is represented by actual data points, which are called medoids. The medoids are objects within a cluster for which the average dissimilarity between it and all the other members of the cluster is minimal, i.e., it corresponds to the most centrally located point in the cluster (Kassambara). It is very similar in concept to K-Medians and in the case of a univariate situation the median is the same as the medoid. However, in the case of a multivariate problem, which is the most common, the median of each variable will probably generate a data point that does not belong to the dataset.

2.4.3.3. K-Modes

K-Modes is a partitioning method that can be applied to categorical variables within a dataset. In this case the central point of the cluster will be represented by the mode of each one of the categorical variables, that is, the value that appears more frequently in the dataset.

2.4.4. Self-Organizing Maps (SOM)

Self-organizing maps is a form of unsupervised learning that is based on the concept of neural networks. The idea of SOM is to map high-dimensional data into a lower dimension feature map of units (neurons) in a way that data points that are close to each other in the input space will also be close to each other in the output space (feature map) (Henriques, Lobo, & Bação, 2012). In an analogy to the previous clustering methods that have been presented, each unit corresponds to a cluster and every data point must be assigned to one of these units based on the distances between them. In order for this process to work, each unit is represented by a vector of the same dimension (number of variables) as the dataset that needs to be clustered. Figure 2.9 below shows a 2D representation of a SOM in which data points are mapped into an output layer feature map.

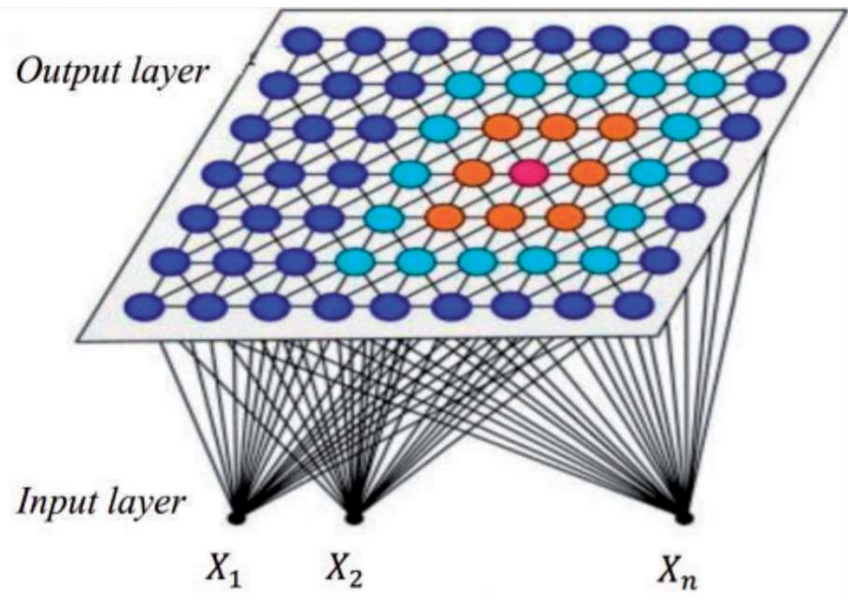

Figure 2. 9: Structural model of SOM
(Zair, Rahmoune, & Benazzouz, 2018)

The SOM training process can be divided in three steps (KHAZRI, 2019):

- A competition step in which each data point is assigned to a unit to which it is closest to. This unit is known as the Best Matching Unit (BMU).
- A cooperation step comes next, in which not only the BMU is identified but also its closest neighbors.

- The last step is the adaptation step, in which the BMU and the closest neighbors are updated to move closer to the data point that is being analyzed. All of these units are updated according to a learning rate that declines based on the distance between the data point and the units, that is, units that are closer to the data point will be updated at a higher rate.

This process is executed for every observation in the dataset and is repeated for a predefined number of iterations, also known as epochs. At the end of each epoch the learning rate is decreased, and this means that during the later epochs the units will move at slower rates towards the datapoints.

Figure 2.10 below shows an example of how the units in the feature map move closer to the data point. In the image the green circles represent the units, and the red X represents the data point that is being classified. As the image shows, the winning unit (BMU) moves at a faster rate closer to the data point. Other units in the neighborhood also move towards the data point but at a slower rate. And the units that are far away from the data point do not move at all.

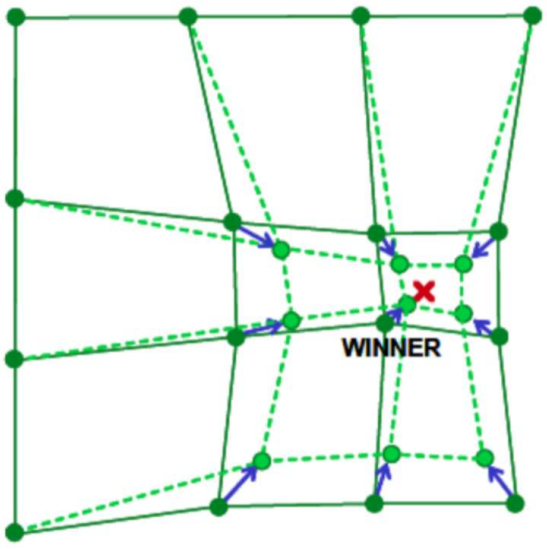

Figure 2. 10: Neurons of the Output Layer Space
(KHAZRI, 2019)

At the end of the implementation of SOM it might be necessary to apply some other type of clustering algorithm in order to create adequate clusters. This happens due to the fact that the output feature map of SOM is usually formed by a high number of units. K-Means, for instance, can be applied to identify portions of the feature map where there are larger concentrations of data and that, consequently, indicate the actual clusters of data.

As it has been mentioned in the description of the SOM algorithm, the BMU is not the only unit to be updated and to move closer to the data point. This also happens to the closest neighbors of the BMU, which are calculated based on a neighborhood function. If it were not for this, the SOM algorithm

would be practically identical to the K-Means algorithm (Bação, Lobo, & Painho, 2005). In this case, if we consider each unit as a centroid and if the radius of the neighborhood function were equal to zero, the centroid would only be influenced by the data points that are related to it.

2.4.5. Combination Methods

An alternative to utilizing just one of the clustering techniques described above is to implement a combination of two of these methods. Two combinations can be particularly useful: K-Means with Hierarchical and SOM with K-Means.

2.4.5.1. K-Means with Hierarchical

This is a strategy that can be applied in order to identify the appropriate number of clusters in a dataset before applying K-Means. This is done by firstly applying a Hierarchical clustering algorithm to define what would be the ideal number of clusters and then applying the result of this as the K in a K-Means implementation.

Another option is to do the inverse by, initially, implementing K-Means with a large number of clusters and then applying a hierarchical method in order to determine the appropriate number of clusters.

2.4.5.2. K-Means with SOM

The implementation of SOM is done based on a n-dimensional grid of neurons which represents the output space. The problem is that usually this grid is formed by a large number of neurons and, consequently, a large number of clusters. So, in most cases, after applying a SOM algorithm it is necessary to apply another clustering algorithm to the results of SOM. One of the preferred choices for this is the K-Means algorithm since it will cluster the grid of neurons instead of the initial data. By doing this, the algorithm will try to identify clusters of data based on the proximity of the neurons, since during the execution of the first algorithm each of them will move throughout the output space according to the number of data points that are close by.

By using this strategy, one will be taking advantage of both clustering techniques. First, the SOM algorithm will make it possible to analyze the distribution of the data through the output space. And, later, K-Means will translate this into a reasonable number of clusters that can be clearly identified and represented.

2.5. PROFILING

One of the last steps, but not less important, during the process of implementing clustering techniques is to perform cluster profiling. This is done by analyzing the results that were achieved and trying to make sense of them. This process usually involves the participation of business experts that are more accustomed to the nuances of the field of study.

The objective of cluster profiling is to verify if the clusters that were generated by the algorithm translate into real world clusters for the problem at hand and if they clearly identify and separate the data points in a way that makes sense for any future analysis and decision making.

2.5.1. Radar Chart

A very useful tool to compare different clusters and profiles is the radar chart. With this tool it is possible to compare the behavior of numerical variables amongst different observations. Since clusters can be represented by a centroid or an average value, radar charts can be applied to compare these values to perform graphical analysis of their characteristics.

Radar charts are built by attributing each variable to an individual axis, which are spaced equally, and arranging them around the center. Each observation is then plotted along each axis like a scatter plot and the points are connected to form a polygon (Radečić, 2021). Afterwards, each observation can be plotted into the same plot. Alternatively, individual plots can be generated for each observation.

Figure 2.11 below shows an example of a radar chart that was built to compare variables of three different observations.

Figure 2. 11: Example of a Radar Chart
(Radečić, 2021)

This example compares 5 variables regarding the qualities of three restaurants. The radar chart helps in identifying the strengths of each one of these restaurants. One performs better in terms of affordability while another performs better in terms of food variety, quality, and ambience. However, the important aspect of this analysis is that it could be applied for a group of observations (clusters) such as different restaurant chains.

3. CLUSTER ANALYSIS

In this section the gambling data regarding user behavior on online platforms in Portugal will be analyzed. Initially, the data from 2019 will be examined in order to identify user clusters. Afterwards, the data from 2020 will also be analyzed and the results will be compared to those of 2019. All the data analysis will be performed in Python through the use of Jupyter Notebooks and during the presentation of the results some of these steps will be shown.

First it is important to highlight that a decision was made in terms of the scope of this work. The platforms that have authorization to promote online gambling in Portugal offer two types of activities: sports bets and games of chance. In 2019 sports bets generated a gross income close to 107 million euros and games of chance generated a gross income close to 109 million euros, indicating both had almost equal relevance in the business. In terms of betting volumes sports bets totaled something around 543 million euros while games of chance reached values of 2,9 billion euros (Turismo de Portugal, 2019). With that in mind, and considering that sports bets probably suffered impacts in 2020 caused by the Covid-19 pandemic that would complicate any comparison with data from 2019, a decision was made to limit the scope of this work to data regarding games of chance.

Regarding the scope of games of chance, the following four games were made available for playing during the fourth trimester of 2019: slot machines, roulette, blackjack, and poker (two types). The distribution of the betting volumes between these games is shown in Figure 3.1 below.

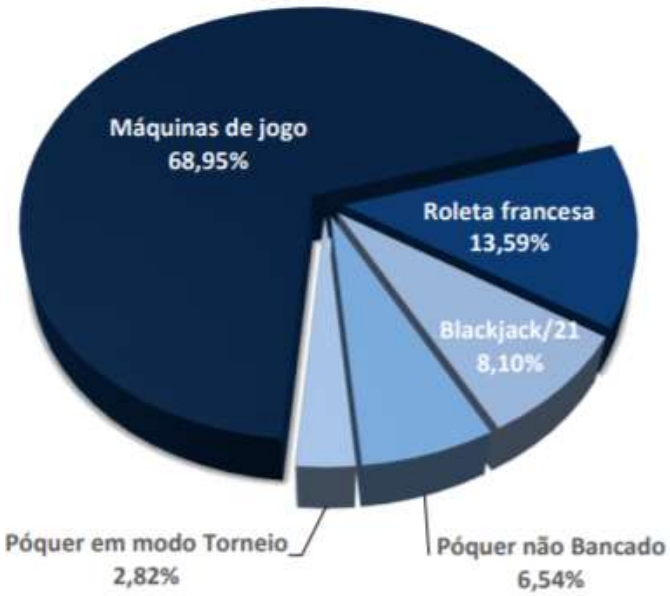

Figure 3. 1: Distribution of Betting Volume in 2019
(Turismo de Portugal, 2019)

As the image shows, slot machines represented almost 69% of betting volumes in the fourth trimester of 2019. Since this indicates that slot machines are very significant to this business and that it is a form of game that attracts intense level of gaming, it has been chosen for the analysis that has been proposed in this work. Consequently, the scope of the work will be limited to the data regarding user activity while playing online slot machines.

3.1. CLUSTER ANALYSIS FOR 2019 DATA

The first portion of this work will be dedicated to the evaluation of data from 2019.

3.1.1. 2019 Data Loading and Initial Analysis

The original and raw data regarding user activity within the ten entities that had permission to provide online gambling in 2019 were delivered to NOVA IMS by SRIJ. The raw data contains specific details of each transaction executed in the platforms throughout the year.

This data was consolidated by the staff of NOVA IMS into the form of daily records, which will be the starting point of all analysis performed in this work. The consolidated file was delivered in the form of a CSV file with 3.512.157 entries and with no missing values.

The file contains the following variables:

Variable	Description
operation	Variable created during the consolidation of the data
entity_id	Id of the Entity (1,2,3,4,5,6,7,9,10,11)
player_id	Id of the Player in that Entity
day_dt	Date in the format 'YYYYMMDD'
day_num	Day of Month
day_of_week	Day of Week (0,1,2,3,4,5,6)
amount	Total Amount played during the day
amount_var	Variation of the Amount during the day
amount_std	Standard Deviation of the Amount during the day
hours_played	Hours played during the day
count	Number of bets placed during the day
wins	Number of wins during the day
balance	Finantial Balance at the end of the day
dawn	Number of bets placed during the dawn period (00:00 to 5:59)
morning	Number of bets placed during the morning period (06:00 to 11:59)
afternoon	Number of bets placed during the afternoon period (12:00 to 17:59)
night	Number of bets placed during the night period (18:00 to 23:59)
amount_mean	Avarage Amount for each bet
wins_perc	Percentage of wins during the day
dawn_perc	Percentage of bets placed during dawn
morning_perc	Percentage of bets placed during the morning
afternoon_perc	Percentage of bets placed during the afternoon
night_perc	Percentage of bets placed during the night
balance_perc	Ratio of balance and amount
player_uid	Unique User Id
self_excluded	Indication if the user has ever been self excluded

Table 3. 1: List of Variables in 2019 CSV File

In summary, each record contains the description of user activity during a specific day in which that player was active in one of the ten online platforms.

To better understand this information a few of the variables listed on Table 3.1 must be better explained:

- PLAYER_ID: this variable refers to the identification of a user in a specific platform. This means that the same person can be registered in two or more platforms and will consequently have a different PLAYER_ID in each one of them

- PLAYER_UID: this refers to a unique user identification and indicates that a person will have the same ID throughout all the records in the dataset.

- WINS: since the concept of victory in gambling can be considered subjective, for this study only situations in which the player finishes the bet with more money than what was invested initially will be considered a win. So, for instance, if a player invests 10 euros in a bet and ends up with 8 euros that will not be considered a win, since the player's balance will be of minus 2 euros.

- SELF_EXCLUDED: all online platforms must provide an alternative for users that feel they do not have a healthy gambling behavior to exclude themselves from the platforms. This binary variable indicates if the user has, at any time, been self-excluded in that specific platform. It is important to mention that this variable is just an indication if the player has ever been self-excluded with no indication of the period in which this happened.

An important aspect to consider in this initial analysis of the raw data is that the records are not evenly distributed among the 10 platforms. Table 3.2 below shows the number of records of each entity and, also, the percentage of records of each entity in relation to the total number of records of the dataset (3.512.157 entries).

Entitity ID	Number of Records	Percentage
1	635.367,00	18,1%
2	479.268,00	13,6%
3	872.422,00	24,8%
4	624.067,00	17,8%
5	184.364,00	5,2%
6	327.880,00	9,3%
7	115.731,00	3,3%
9	15.626,00	0,4%
10	121.856,00	3,5%
11	135.576,00	3,9%
Total	3.512.157,00	100%

Table 3. 2: Number of records of each Entity

The information in the table shows that while some entities have a very significant participation in online slot machine gambling others have very little representation. This could be caused by the fact that some entities are newer to the business and have not yet achieved a relevant user base. This also indicates that the results achieved during this work could have variations based on the business strategies implemented by each entity since some of them might be willing to accept smaller profits than others.

Another relevant matter that can be analyzed in the raw data is the variation of the attributes during the year since some specific user behaviors can be influenced by seasonality. Figure 3.2 below shows the graphic representations of the variation of four major attributes in this dataset based on the average values of each month.

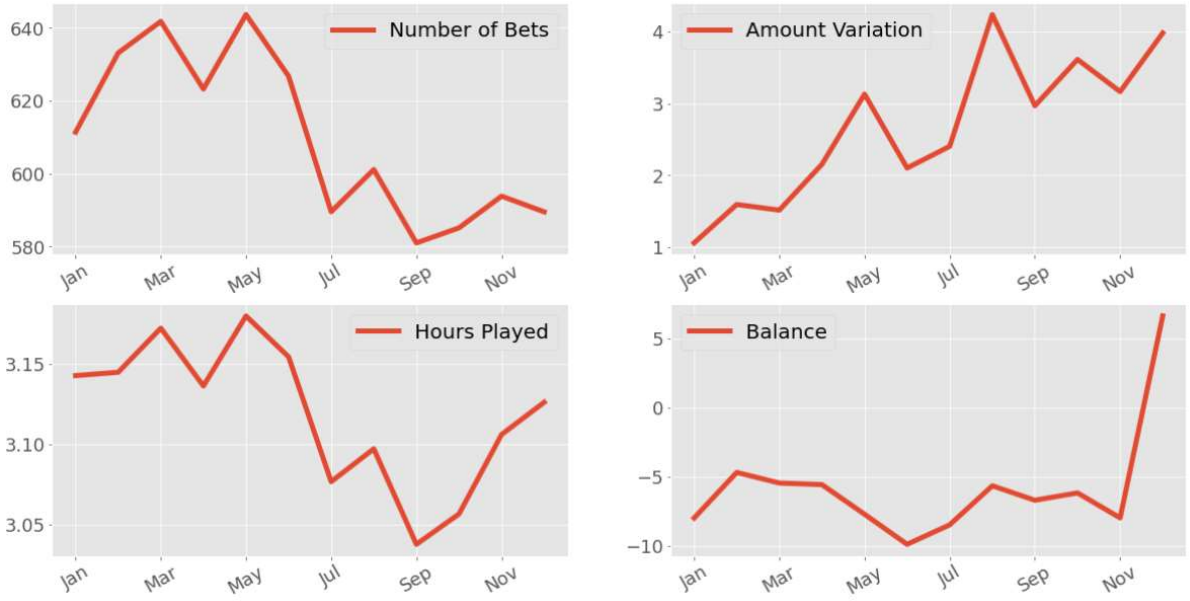

Figure 3. 2: Variation of attributes during 2019

This image demonstrates that the number of bets and the total number of hours played during the day did not change very much during the year. It is noticeable, however, that the amount variation and balance increased towards the end of the year. Given that the increase is relatively significant for both variables it is conceivable that this is caused by outliers.

The increase in balance is especially intriguing since it has a negative value throughout the year and jumps to a positive value in the month of December. Buy looking at the average values for each month it becomes clear that in fact the values that represent the balance are unexpected. Between the months of January and November the highest value for balance was -4,69 and the lowest value was -9,90. In December the average balance becomes 6,63 and this would indicate that on average the players would have had a profit over the online platforms during this month. Since this wouldn't be reasonable from a business standpoint it very likely that these numbers are being inflated due to

outliers and that some sort of treatment will be necessary to avoid that these outliers pollute the models that will be created.

Outliers can represent actual values that indicate extreme values can occur in the dataset. They can also, however, be generated by data entry errors and interface problems while data is being transferred from one system to another. In the case of the variable BALANCE there are records that indicate daily balances of over 200.000 euros, which seems extremely unreasonable. So, as a way of analyzing the effects of extreme values in this dataset, the monthly average of the variable BALANCE will be calculated, firstly with all the records and secondly with a limitation of the values, considering the number that represents the 0,99 percentile, i.e., all values that are higher than the 0,99 percentile will be changed to this maximum value. Tables 3.3 and 3.4 show the results of this comparison.

Month	Average Balance (€)
January	-7,99
February	-4,69
March	-5,46
April	-5,58
May	-7,72
June	-9,90
July	-8,49
August	-5,66
September	-6,71
October	-6,18
November	-7,99
December	6,63

Table 3. 3: Average Balance with Outliers

Month	Average Balance (€)
January	-23,92
February	-26,39
March	-25,76
April	-27,00
May	-29,58
June	-29,42
July	-27,58
August	-25,28
September	-26,54
October	-26,93
November	-25,23
December	-20,86

Table 3. 4: Average Balance without outliers

First, it is noticeable how these outliers affect the average monthly values. But more importantly is the fact that specifically for the month of December the outliers caused the average value to be totally incoherent with what had happened in previous months and even with what would be expected as a normal balance value from a business point of view.

3.1.2. 2019 Data Preparation

As it has been demonstrated previously, it is important to perform some level of preparation of the dataset. This involves things like feature engineering, variable selection, exclusion of outliers, among others, and will be addressed in the following sections.

3.1.2.1. Feature Engineering

One aspect of this dataset that stands out is the fact that it does not contain many categorical variables that would help describe the users such as age, gender, and location. The only categorical variables that are present in the dataset are ENTITY_ID and SELF_EXCLUDED which indicate the platform entity identification and if the player has ever been self-excluded from that platform respectively.

Given this fact and considering that feature engineering can generate important information for machine learning models, a few attributes were created to enhance these models. Table 3.5 below shows a list and description of these new features.

Variable	Description
day_type	Categorical variable to determine if the day related to the record refers to a weekday or a weekend. For this purpose, Monday, Tuesday, Wednesday and Thursday will be considered weekdays and Friday, Saturday and Sunday will be considered weekends. Weekdays will have value 0 and weekends will have value 1.
user_id	An identification of the player in each entity. This attribute will be a concatenation of the entity_id and player_id variables.
avg_play_per_hour	This will be a calculated attribute that will show how many bets the user made on average each hour of the day. It will be calculated by dividing the hours_played variable by the count variable.
automated_play	Binary categorical variable to determine if the player used any form of automated tool to play during the day. Whenever the avg_play_per_hour is greater than 360 this will be considered true and receive value 1. Otherwise, it will have value 0.
winning_day	Categorical variable that indicates if the player had a "winning" day. This attribute will receive value 1 if the player balance is positive and will receive 0 otherwise.

Table 3. 5: New Features

3.1.2.2. Data Aggregation

Another relevant aspect of this dataset is that the data is grouped in the form of daily records. This means that the information regarding user activity is organized in a way that shows the totality of the interactions the user had in each day of the year. As a result, each user will have as many records in the dataset as the number of days in which he or she was active in online gambling platforms. For instance, if a player was active on 10 different days of the year, that player will have 10 records in the dataset.

Since the objective of this work is to understand player behavior and to analyze how the average behavior might have changed from 2019 to 2020, a decision was made to group the data considering user identifications. In this manner a dataset will be generated in which each record will show the "average" behavior of one specific user during the year of 2019.

For the purposes of aggregation, the variable USER_ID that represents the identification of users in each entity has been chosen instead of the variable PLAYER_UID that represents a unique identification of users across all platforms. This is due to the fact that the PLAYER_UID attribute is not considered completely reliable and could generate inconsistencies in the result of the aggregation.

The table below shows the list of the attributes of the dataset that were generated to aggregate data of users into individual records.

Variable	Description
user_id (index)	User Id that combines original Id and Entity Number
days_played	Number of days of the year in which the user played slot machine
avg_count	How many time the user played on average each day
avg_win_perc	How many wins the player had on average each day
avg_amount	The finantial amount the player bet on average each day
avg_amount_mean	The average of the mean amount per bet
avg_amount_var	The average of the amount variation
avg_amount_std	The average of the amount standard deviation
avg_balance	The average of user balance per day
avg_auto_play	The percentage of days in which the player used auto play mecanism
avg_day_type	The pecentage of days that were weekends (Friday-Sunday)
avg_hours_played	How many hours the user played on average per day
avg_dawn_perc	Average percentage of dawn play
avg_morning_perc	Average percentage of morning play
avg_afternoon_perc	Average percentage of afternoon play
avg_night_perc	Average percentage of night play
self_excluded	Indicates if player was ever self exluded
entity_id	Entity Id
winning_day_avg	Percentage of days in which the player had a winning day

Table 3. 6: Attributes of aggregated dataset

The decision to aggregate the dataset based on the users may cause the loss of specific daily information which will have to be analyzed as an average throughout the year. It will, on the other hand, allow for a better understanding of general characteristics of each user. And by doing this it will be possible to separate these users in clusters based on their behavior during the year.

The dataset that was generated with information aggregated by users has 253.826 records, meaning this is the number of users that played online slot machine at least once in 2019.

3.1.2.3. Scope Limitation

Following this decision to group the dataset into records that identify each user, comes the decision of which records should be considered for this analysis. Since the idea is to understand the typical user behavior and later to compare this with data from 2020 to identify any behavior shifts, it would not make sense to keep in the dataset records related to users that have had limited amount of interaction with the online gambling platforms. Table 3.7 below shows the distribution of users according to the frequency in which they played. In this case the frequency is calculated based on the number of days of the year in which the player placed at least one bet.

Days Played	Percentage
(0,1]	37%
(1,5]	30%
(5,10]	10%
(10,50]	16%
(50,100]	4%
(100,300]	3%
(300,365]	0%

Table 3. 7: Distribution of users according to frequency of play

As Table 3.7 shows, 67% of users were only active in online gambling in five days or less during 2019. For the purposes of understanding user profiles and identifying specific user behaviors these users can be considered irrelevant and could actually harm the analysis. In a more direct conclusion, and for the objectives of this work, these users can be classified as “non-players”.

With this evaluation it was decided to exclude these users from the dataset and to proceed in the analysis only with users that were active in at least 6 days of the year. This represents around 33% of the data and reduces the dataset to 92.021 records.

3.1.2.4. Feature Selection

There are many motives to exclude variables from a dataset. Some are clearly irrelevant to the analysis that will be performed. In the case of this work variables from the original dataset such as OPERATION and PLAYER_UID would be of no help. The former is just a system generated variable and the latter simply will not be used since it was decided to use the variable PLAYER_ID. Other than that, since it was decided to aggregate the dataset based on users, some variables would just be meaningless such as DAY_NUM and DAY_OF_WEEK.

Finally, another important way of identifying relevant attributes and excluding some of the variables in the dataset is to analyze the correlation between them in order to exclude highly correlated variables that would end up polluting the clustering models. Figure 3.3 below shows a Pearson correlation matrix of the variables in this dataset.

Figure 3. 3: Pearson Correlation Matrix of Dataset Variables

By looking at the correlation matrix it is possible to identify correlation between the following variables:

- AVG_AMOUNT, AVG_AMOUNT_MEAN, AVG_AMOUNT_VAR and AVG_AMOUNT_STD (from now on these will be referred to as AMOUNT VARIABLES)
- AVG_HOURS_PLAYED, DAYS_PLAYED, AVG_COUNT and AVG_AUTO_PLAY (from now on these will be referred to as FREQUENCY VARIABLES)

- WINNING_DAY_AVG and AVG_WIN_PERC (from now on these will be referred to as WINNING VARIABLES)
- AVG_DAWN_PERC, AVG_MORNING_PERC, AVG_AFTERNOON_PERC and AVG_NIGHT_PERC (from now on these will be referred to as PERIOD OF DAY VARIABLES)

These correlation values can be used as input for the selection of variables that will be used in the clustering models and can help in avoiding redundant variables.

3.1.2.5. Outlier Treatment

As it was observed in the evaluation of the original dataset, outliers might be present in the data and could cause inconsistencies in the models that will be generated.

The DESCRIBE method present in the Pandas Library shows a general overview of the distribution of the data according to certain percentiles. Table 3.8 below shows the result of this method applied to the AMOUNT and BALANCE variables of the aggregated dataset using the following percentiles: 0,25, 0,50, 0,75, 0,90 and 0,99.

Percentile/Variable	AVG_AMOUNT	AVG_AMOUNT_MEAN	AVG_AMOUNT_VAR	AVG_AMOUNT_STD	AVG_BALANCE
Mean	392,23	0,62	2,41	0,29	-1,94
Minimum Value	0,01	0,00	0,00	0,00	-4.365,11
25th Percentile	44,20	0,21	0,01	0,06	-11,92
50th Percentile	115,35	0,34	0,03	0,12	-3,54
75th Percentile	297,60	0,61	0,12	0,25	3,04
90th Percentile	756,07	1,13	0,64	0,53	24,85
99th Percentile	4.492,78	4,79	21,72	2,90	209,88
Maximum Value	272.597,95	135,36	15.388,41	67,74	25.887,55

Table 3. 8: Percentiles of AMOUNT and BALANCE Variables

Through this table it very clear that the dataset has very extreme values that are even much higher than the 99th percentile. Some of these values seem unreasonable and were probably generated due to errors, either in the data collection or in some of the interfaces through which the data was transmitted.

To help visualize these outliers and to have some sense of their representation, boxplots of the following four variables were plotted: AVG_AMOUNT, AVG_AMOUNT_MEAN, AVG_AMOUNT_VAR and AVG_BALANCE. Figure 3.4 below shows these boxplots.

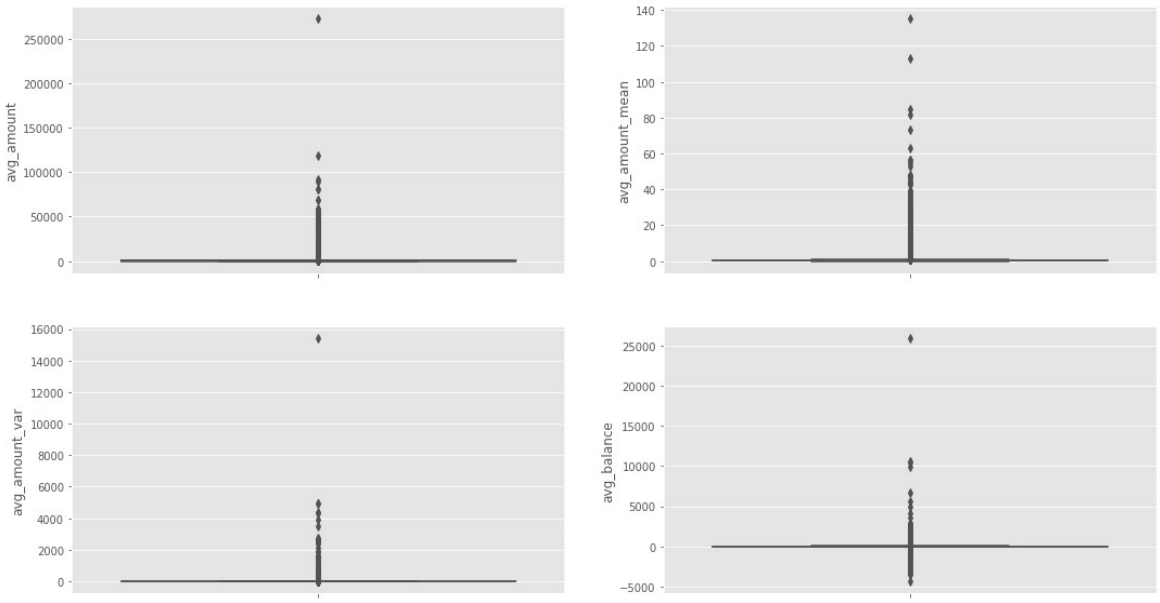

Figure 3. 4: Boxplots of Selected Variables

By looking at the boxplots of these four variables it is visible that there are many outliers. In the boxplots the lines that represent the upper and lower limits (IQR * 1,5) are barely visible and the number of records beyond the lines is very significant. By considering the traditional values for upper and lower limits there are a total of 26.063 records that have outliers in at least one of the five variables related to AMOUNT or BALANCE. This represents almost 30% of the dataset and simply excluding these records would cause great losses to the models since this would mean losing relevant information of many frequent users.

Since the number of outliers is very significant and since the values above the 99th percentile are very extreme, one strategy that can be applied is the limitation of extreme values to the value of the 99th percentile. In this approach, instead of excluding outliers, their values will be replaced by an upper limit, in this case the value of the 99th percentile. This will be done for the variables related to AMOUNT and BALANCE as shown on Figure 3.5 below that contains the Python code that executes this transformation.

```python
# Outlier Treatment

max_value = frequent_user_df['avg_amount'].quantile(0.99)
frequent_user_no_outlier.loc[:,['avg_amount']] = \
frequent_user_no_outlier['avg_amount'].apply(lambda x: max_value if x>max_value else x)

max_value = frequent_user_df['avg_amount_mean'].quantile(0.99)
frequent_user_no_outlier.loc[:,['avg_amount_mean']] = \
frequent_user_no_outlier['avg_amount_mean'].apply(lambda x: max_value if x>max_value else x)

max_value = frequent_user_df['avg_amount_var'].quantile(0.99)
frequent_user_no_outlier.loc[:,['avg_amount_var']] = \
frequent_user_no_outlier['avg_amount_var'].apply(lambda x: max_value if x>max_value else x)

max_value = frequent_user_df['avg_amount_std'].quantile(0.99)
frequent_user_no_outlier.loc[:,['avg_amount_std']] = \
frequent_user_no_outlier['avg_amount_std'].apply(lambda x: max_value if x>max_value else x)

max_value = frequent_user_df['avg_balance'].quantile(0.99)
frequent_user_no_outlier.loc[:,['avg_balance']] = \
frequent_user_no_outlier['avg_balance'].apply(lambda x: max_value if x>max_value else x)
```

Figure 3. 5: Python code for Outlier Transformation

For the purposes of the models that will be generated in the following sections of this work, a variety of alternatives will be evaluated, including options regarding outliers. Some models will be implemented with the original outlier values and others will be implemented with the limitation of the 99th percentile described above.

3.1.3. Implementation of Clustering Techniques to 2019 Data

In this section, some alternatives will be considered with the objective of identifying the most adequate implementation of clustering algorithms for this data, in particular. The following items will be evaluated:

- Data Rescaling: Standardization and Normalization.

- Variable Selection: A variety of combinations of variables will be tested, mainly according to their correlation.

- Number of Clusters: This will be evaluated with the help of Elbow Graphs and inter-cluster distances.

- Outliers: The best models will be evaluated with outliers and with outlier treatment.

- Algorithms: For this dataset the K-Means algorithm and SOM will be evaluated. Hierarchical methods are not appropriate in this case due to the large number of records.

In addition to the alternatives described above, silhouette graphs of some of the most adequate models will be generated to help analyze the results. Finally, after identifying the best model, some specific analysis will be made to better understand the final results and cluster profiles. It is also

relevant to mention that the model that is considered the most appropriate for the 2019 data will also be applied in the analysis of the 2020 data.

3.1.3.1. Model 1

This model will be implemented with the following configuration:

- Algorithm: K-Means
- Variables: All the variables in the dataset
- Scaler: Standard Scaler
- Outliers: Outliers will not be altered

The Elbow Graph of this model is displayed in Figure 3.6 below.

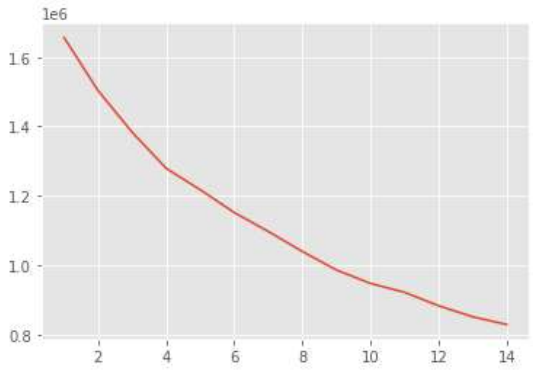

Figure 3. 6: Elbow Graph of Model 1

The elbow graph does not give a clear indication for the ideal number of clusters in this configuration. Since it has a slight change of inclination around number 4, the model will be implemented with four clusters. Table 3.9 below shows the values of the centroids of each of the four clusters that were generated by the algorithm.

Cluster	DAYS_PLAYED	AVG_COUNT	AVG_WIN_PERC	AVG_AMOUNT	AVG_AMOUNT_MEAN	AVG_AMOUNT_VAR
0	75.84	1257.90	0.20	1257.60	0.90	3.48
1	28.13	299.90	0.17	182.24	0.52	0.62
2	47.01	920.08	0.20	27320.54	27.80	1051.59
3	26.46	305.90	0.20	195.41	0.53	0.80

Cluster	AVG_AMOUNT_STD	AVG_BALANCE	AVG_AUTO_PLAY	AVG_DAY_TYPE	AVG_HOURS_PLAYED	AVG_DAWN_PERC
0	0.50	-6.86	0.22	0.42	4.64	0.24
1	0.22	-3.63	0.03	0.43	2.15	0.30
2	17.27	275.01	0.15	0.38	3.57	0.28
3	0.24	1.50	0.02	0.40	2.34	0.10

Cluster	AVG_MORNING_PERC	AVG_AFTERNOON_PERC	AVG_NIGHT_PERC	SELF_EXCLUDED	ENTITY_ID	WINNING_DAY_AVG
0	0.14	0.30	0.35	0.24	4.15	0.31
1	0.07	0.21	0.43	0.09	3.97	0.22
2	0.13	0.29	0.33	0.27	4.50	0.27
3	0.24	0.47	0.25	0.11	4.46	0.25

Table 3. 9: Centroids of Model 1

As the table shows, the centroids are not well defined and are not interpretable, thus not generating reasonable clusters. Additionally, the clusters are not well distributed and cluster 2 contains only 111 records. This is probably due to the fact that there is an excess of variables and some of them are highly correlated. In the following sections, models with alternative variable selections and parameter configurations will be implemented.

3.1.3.2. Model 2

This model will be implemented with the following configuration:

- Algorithm: K-Means
- Variables: Excluding ENTITY_ID and SELF_EXLCUDED; keeping only one AMOUNT variable; keeping two FREQUENCY variables
- Scaler: Standard Scaler
- Outliers: Outliers will not be altered

The Elbow Graph of this model is displayed in Figure 3.7 below.

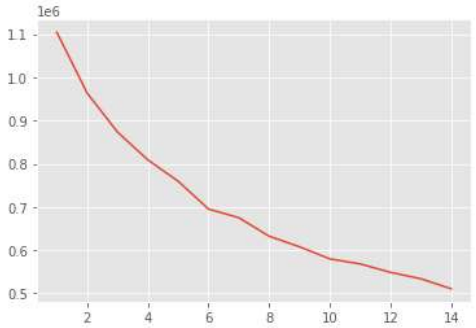

Figure 3. 7: Elbow Graph of Model 2

The intra-cluster distances (inertia) of this model are slightly better than of the previous model and the elbow graph suggests the model should have six clusters. Table 3.10 shows the centroids of this model.

Cluster	AVG_WIN_PERC	AVG_AMOUNT_VAR	AVG_BALANCE	AVG_HOURS_PLAYED	AVG_AUTO_PLAY	AVG_DAY_TYPE
0	0.17	1.81	-10.09	2.47	0.02	0.40
1	0.32	7142.50	15401.52	5.25	0.35	0.44
2	0.30	3.07	43.44	2.78	0.04	0.42
3	0.16	1.28	-9.14	2.20	0.03	0.44
4	0.16	2.89	-10.63	2.25	0.04	0.42
5	0.19	3.00	-21.14	4.76	0.32	0.43

Cluster	WINNING_DAY_AVG	AVG_DAWN_PERC	AVG_MORNING_PERC	AVG_AFTERNOON_PERC	AVG_NIGHT_PERC	WINNING_DAY_AVG
0	0.19	0.10	0.26	0.48	0.23	0.19
1	0.32	0.27	0.14	0.35	0.32	0.32
2	0.49	0.18	0.15	0.35	0.36	0.49
3	0.19	0.15	0.08	0.25	0.54	0.19
4	0.20	0.50	0.08	0.18	0.26	0.20
5	0.29	0.25	0.13	0.28	0.36	0.29

Table 3. 10: Centroids of Model 2

This model also does not generate interpretable clusters. Some variables seem irrelevant in terms of cluster differentiation. Cluster distribution also was not ideal with cluster 1 having only 3 records.

3.1.3.3. Model 3

This model will be implemented with the following configuration:

- Algorithm: K-Means
- Variables: Excluding ENTITY_ID and SELF_EXCLUDED; keeping only one AMOUNT variable; keeping two FREQUENCY variables; keeping one WINNING variable; excluding all PERIOD OF DAY variables
- Scaler: Standard Scaler
- Outliers: Outliers will not be altered

The elbow graph for this model is shown on Figure 3.8.

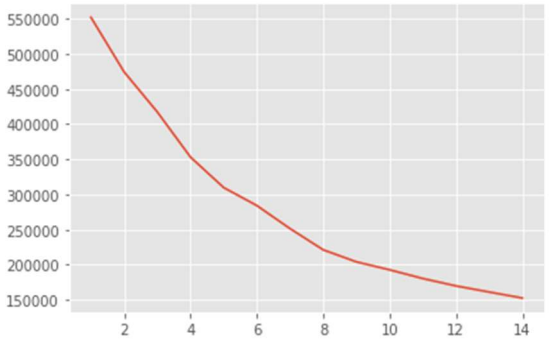

Figure 3. 8: Elbow Graph for Model 3

The intra-cluster distances seem to have improved in this model and the graph suggests five clusters. Which are represented by its centroids on Table 3.11 below.

Cluster	DAYS_PLAYED	AVG_HOURS_PLAYED	AVG_WIN_PERC	AVG_AMOUNT_VAR	AVG_BALANCE	AVG_DAY_TYPE
0	17.83	2.58	0.35	4.49	53.33	0.40
1	19.94	2.20	0.17	1.01	-7.29	0.57
2	110.18	4.40	0.18	3.81	-13.33	0.42
3	15.00	5.25	0.32	7142.50	15401.52	0.44
4	21.00	2.21	0.16	1.71	-10.88	0.32

Table 3. 11: Centroids of Model 3

Model 3 is the best model until now, since it seems to generate reasonably interpretable clusters, even though some variables remain irrelevant to the model such as AVG_WIN_PERC and AVG_DAY_TYPE. Additionally, cluster 3 contains only three records, probably due to outliers.

3.1.3.4. Model 4

This model will be implemented with the following configuration:

- Algorithm: K-Means
- Variables: Excluding ENTITY_ID and SELF_EXCLUDED; keeping only one AMOUNT variable; keeping two FREQUENCY variables; keeping one WINNING variable; excluding all PERIOD OF DAY variables
- Scaler: Normalization (Minmax)
- Outliers: Outliers will not be altered

The Elbow Graph of this model is shown on Figure 3.9.

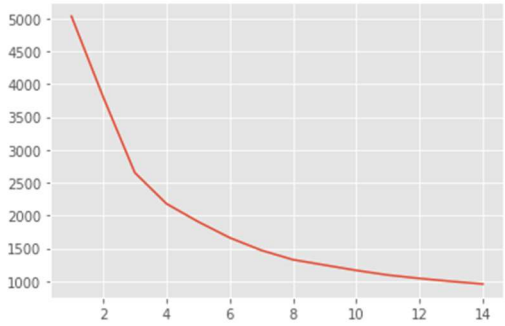

Figure 3. 9: Elbow Graph of Model 4

The elbow graph suggests four clusters, which are represented by its centroids on Table 3.12.

Cluster	DAYS_PLAYED	AVG_HOURS_PLAYED	AVG_WIN_PERC	AVG_AMOUNT_VAR	AVG_BALANCE	AVG_DAY_TYPE
0	29.61	2.76	0.19	2.07	-2.62	0.43
1	13.75	2.17	0.20	3.64	0.62	0.23
2	13.60	2.21	0.19	1.34	0.64	0.64
3	166.23	3.78	0.18	3.35	-9.83	0.42

Table 3. 12: Centroids of Model 4

This model generates clearly identifiable clusters. Cluster 3, for example, represents high intensity gambling with greater losses. Clusters 1 and 2 represent users that make profits but have specific gambling habits.

3.1.3.5. Model 5

This model will be implemented with the following configuration:

- Algorithm: K-Means
- Variables: Excluding ENTITY_ID and SELF_EXCLUDED; keeping only one AMOUNT variable; keeping two FREQUENCY variables; keeping one WINNING variable; excluding all PERIOD OF DAY variables
- Scaler: Standard Scaler
- Outliers: Outliers will be limited to the 99th Percentile

The elbow graph for this model is show on Figure 3.10.

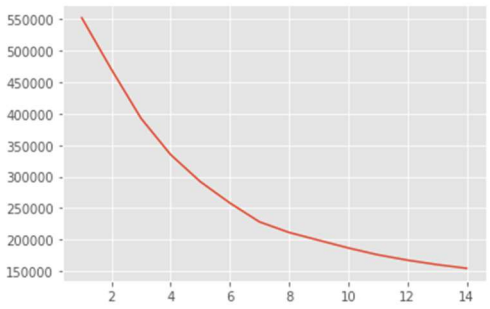

Figure 3. 10: Elbow Graph for Model 5

The elbow graph indicates that the ideal number of clusters is seven. Since this seems exaggerated for the problem at hand the model will be implemented with six clusters. They are represented by the centroids in Table 3.13.

Cluster	DAYS_PLAYED	AVG_HOURS_PLAYED	AVG_WIN_PERC	AVG_AMOUNT_VAR	AVG_BALANCE	AVG_DAY_TYPE
0	17.67	2.44	0.36	0.34	37.49	0.40
1	169.61	3.64	0.17	0.42	-7.47	0.42
2	46.46	3.19	0.20	18.35	-154.28	0.40
3	19.54	2.04	0.17	0.20	-5.68	0.58
4	35.07	4.76	0.18	0.36	-11.54	0.42
5	22.20	2.03	0.16	0.24	-7.53	0.32

Table 3. 13: Centroids for Model 5

This model minimizes the negative effects that outliers had caused in previous models that used Standard Scaler. Clusters can be visualized mainly due to the FREQUENCY variables and the monetary values (AMOUNT and BALANCE). The clusters are well distributed and clusters numbers 1 and, 2 that represent intensive gamblers (possibly addictive behavior), represent around 22% of users.

3.1.3.6. Model 6

This model will be implemented with the following configuration:

- Algorithm: K-Means
- Variables: Excluding ENTITY_ID and SELF_EXCLUDED; keeping only one AMOUNT variable; keeping two FREQUENCY variables; keeping one WINNING variable; excluding all PERIOD OF DAY variables
- Scaler: Normalization (Minmax)
- Outliers: Outliers will be limited to the 99th Percentile

The elbow graph for this model is show on Figure 3.11.

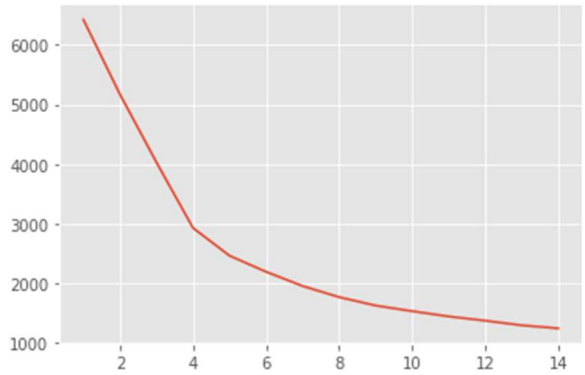

Figure 3. 11: Elbow Graph for Model 6

The elbow graph indicates that the ideal number of clusters could be either four or five. Since in the previous model six clusters were utilized, this model will also be tested with six clusters. The clusters for each of these three options are represented in the following tables:

Cluster	DAYS_PLAYED	AVG_HOURS_PLAYED	AVG_WIN_PERC	AVG_AMOUNT_VAR	AVG_BALANCE	AVG_DAY_TYPE
0	21.38	2.49	0.19	0.27	-2.73	0.32
1	20.30	2.44	0.19	0.23	-2.13	0.56
2	47.01	3.16	0.20	18.41	-125.68	0.40
3	151.51	3.73	0.17	0.40	-8.08	0.42

Table 3. 14: Centroids for Model 6 with 4 Clusters

Cluster	DAYS_PLAYED	AVG_HOURS_PLAYED	AVG_WIN_PERC	AVG_AMOUNT_VAR	AVG_BALANCE	AVG_DAY_TYPE
0	13.66	2.17	0.20	0.25	-1.95	0.23
1	29.51	2.74	0.19	0.27	-3.44	0.43
2	46.97	3.16	0.20	18.43	-125.96	0.40
3	166.30	3.78	0.18	0.41	-7.90	0.42
4	13.58	2.20	0.19	0.21	-1.15	0.64

Table 3. 15: Centroids for Model 6 with 5 Clusters

Cluster	DAYS_PLAYED	AVG_HOURS_PLAYED	AVG_WIN_PERC	AVG_AMOUNT_VAR	AVG_BALANCE	AVG_DAY_TYPE
0	31.26	2.75	0.17	0.27	-7.54	0.43
1	13.56	2.15	0.17	0.24	-6.70	0.23
2	47.08	3.17	0.20	18.46	-126.69	0.40
3	168.32	3.80	0.18	0.42	-7.87	0.42
4	17.01	2.58	0.37	0.30	33.25	0.39
5	13.55	2.18	0.18	0.21	-3.54	0.65

Table 3. 16: Centroids for Model 6 with 6 Clusters

All three alternatives generate reasonable clusters that can be clearly defined. The alternative with 6 clusters, however, seems to generate a more interesting division since cluster four represents players that have a winning pattern of gambling. One thing that stands out is that the variables AVG_WIN_PERC and AVG_DAY_TYPE do not seem relevant to separate users. In the case of the variable AVG_DAY_TYPE, it only serves to separate clusters 1 and 5 indicating that players on cluster 5 have a tendency to play more frequently on weekends. This, however, does not seem to improve user segmentation.

3.1.3.7. Model 7

This model will be implemented with the following configuration:

- Algorithm: K-Means
- Variables: Excluding ENTITY_ID, SELF_EXCLUDED and AVG_DAY_TYPE; keeping only one AMOUNT variable; keeping two FREQUENCY variables; keeping one WINNING variable; excluding all PERIOD OF DAY variables
- Scaler: Normalization (Minmax)
- Outliers: Outliers will be limited to the 99th Percentile

The elbow graph for this model is show on Figure 3.12.

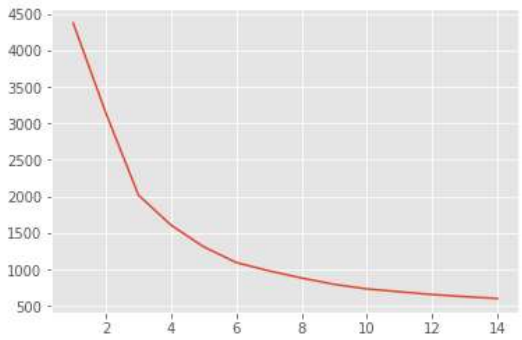

Figure 3. 12: Elbow Graph for Model 7

The elbow graph indicates that the number of clusters could be between 3 and 6, so the model will be implemented with 5 clusters to evaluate if there can be any reduction in comparison to the last model. The centroids for these five clusters are represented in the following table.

Cluster	DAYS_PLAYED	AVG_HOURS_PLAYED	AVG_WIN_PERC	AVG_AMOUNT_VAR	AVG_BALANCE
0	15.05	2.47	0.34	0.28	26.94
1	56.30	3.86	0.18	0.37	-9.21
2	15.28	2.02	0.16	0.20	-7.55
3	191.86	3.87	0.17	0.44	-8.24
4	46.50	3.15	0.20	18.33	-125.04

Table 3. 17: Centroids for Model 7

This model also generated reasonable clusters that can be clearly identified and that are reasonably distributed.

3.1.3.8. Model 8

This model will be implemented with the following configuration:

- Algorithm: SOM
- Variables: Excluding ENTITY_ID, SELF_EXCLUDED and AVG_DAY_TYPE; keeping only one AMOUNT variable; keeping two FREQUENCY variables; keeping one WINNING variable; excluding all PERIOD OF DAY variables
- Scaler: Normalization (Minmax)
- Outliers: Outliers will be limited to the 99th Percentile

The implementation of SOM will be experimented in a 10x10 feature map that will be executed for 100 epochs. Afterwards, K-Means will be applied to the results of SOM in order to obtain six clusters of data.

The BMU Hits View displayed on Figure 3.13 below indicates the number of records associated to each neuron (unit) in the feature map.

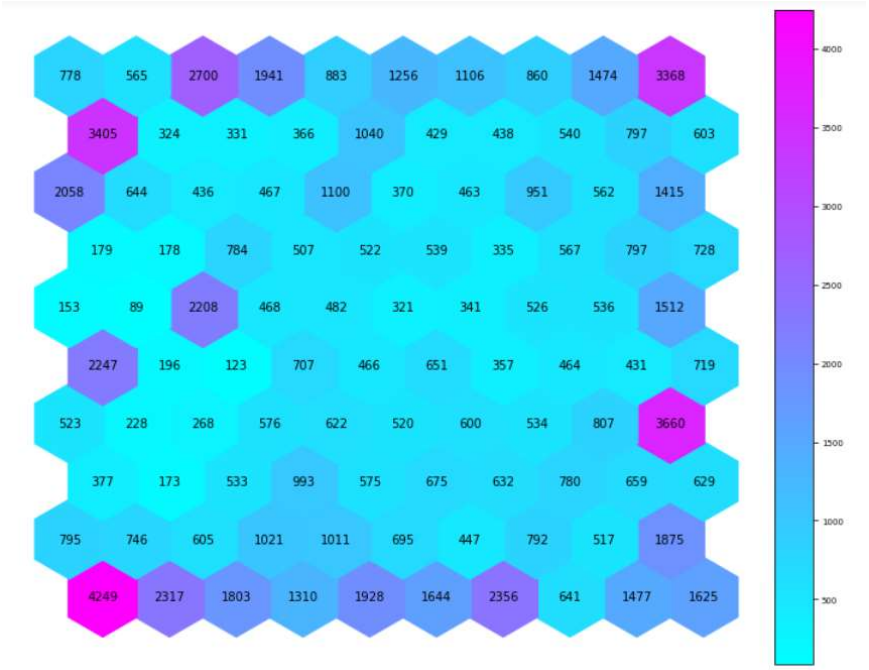

Figure 3. 13: BMU Hits View for Model 8

And after applying K-Means the Hits Map on Figure 3.14 indicates which neurons are associated to each cluster.

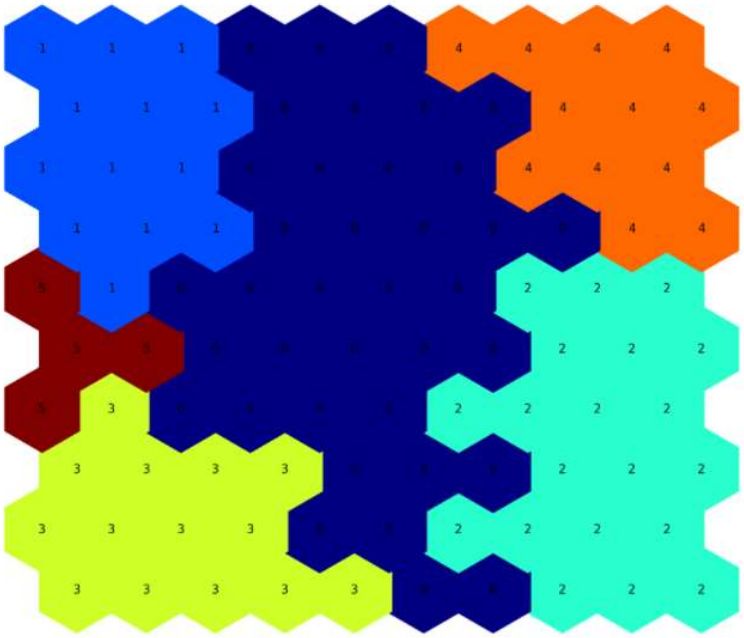

Figure 3. 14: Hits Maps for Model 8

The hits map gives an idea of the distribution of the records between the clusters with Cluster 0 having the greater number of records and Cluster 5 being the smallest cluster.

By calculating the average values of the variables in each cluster it is possible to identify values that can be considered as the centroids of each cluster. These centroids are represented in the following table.

Cluster	DAYS_PLAYED	AVG_HOURS_PLAYED	AVG_WIN_PERC	AVG_AMOUNT_VAR	AVG_BALANCE	AVG_DAY_TYPE
0	29.19	2.80	0.16	0.14	-4.25	0.42
1	124.41	4.35	0.18	0.34	-9.62	0.42
2	15.34	1.94	0,16	0.14	-5.44	0.60
3	13.61	1.95	0.16	0.18	-9.26	0.24
4	18.71	2.36	0.33	0.25	30.29	0.40
5	40.70	3.05	0.19	12.28	-125.96	0.40

Table 3. 18: Centroids for Model 8

The centroids for this model are very similar to those of model 6 that used K-Means and it also generates reasonable clusters that are well distributed.

3.1.4. Model Selection for 2019 Data

After having implemented a series of different models, one of them must be chosen as the primary model for modeling this data. One way of evaluating all these models is to check their silhouette scores. The following table shows the value for each model.

Model	Silhoutte Score
Model 1	0.11
Model 2	0.13
Model 3	0.02
Model 4	0.01
Model 5	0.24
Model 6 (4 clusters)	0.30
Model 6 (5 clusters)	0.27
Model 6 (6 clusters)	0.28
Model 7	0.38
Model 8	0.23

Table 3. 19: Silhouette Scores of the Models

One thing that is clear, by looking at these scores, is that models in which outliers were treated had a much better performance. This is the case for models 5 through 8.

It is clear that Model 7 had the best performance, followed by Model 6. And Model 8, which implemented SOM, did not perform very well in comparison to these other models.

Another form of evaluating the quality of the models is to plot their silhouette coefficients. Since, based on the silhouette average score, it is already possible to have an idea of this quality, only the coefficients for models 6 (with 6 clusters) and 7 will be plotted. These were two of the models with best silhouette average scores.

Figure 3.15 displays the silhouette plot for Model 6 (with 6 clusters).

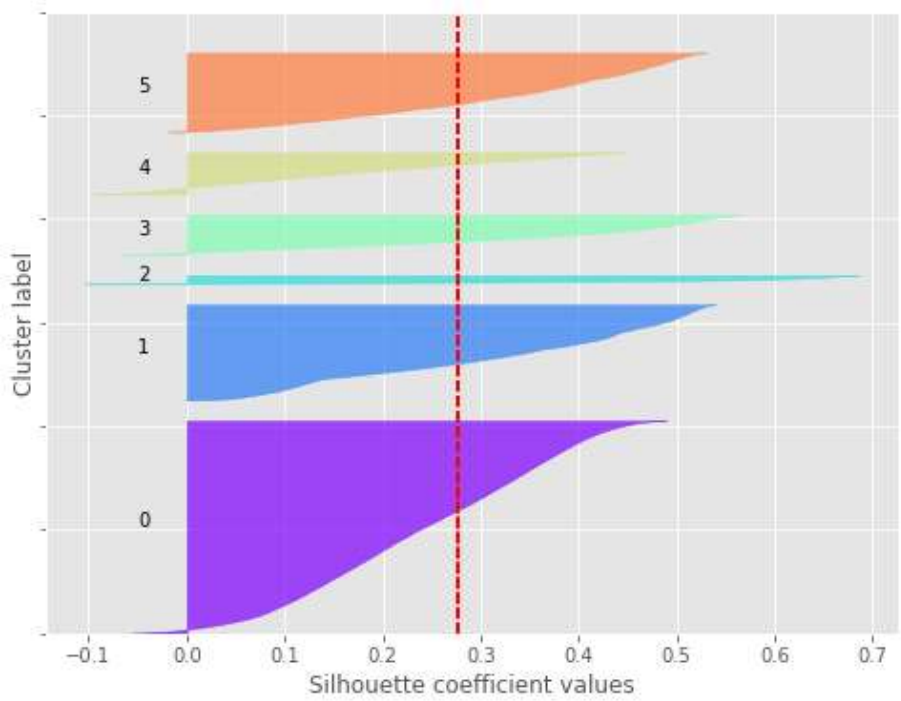

Figure 3. 15: Silhouette Plot for Model 6 (with 6 clusters)

The plot shows that clusters 0, 2, 3, 4 and 5 have some silhouette coefficients below 0 which indicates that some records might have been incorrectly classified. The thickness of these lines, however, indicate that this did not happen for many records. The thickness of the lines also helps in identifying that cluster 0 is the largest cluster and cluster 2 is the smallest cluster. And it is also perceptible that clusters 1, 2, 3 and 5 have most records with silhouette coefficients above the average of 0,28.

Figure 3.16 displays the silhouette plot for Model 7.

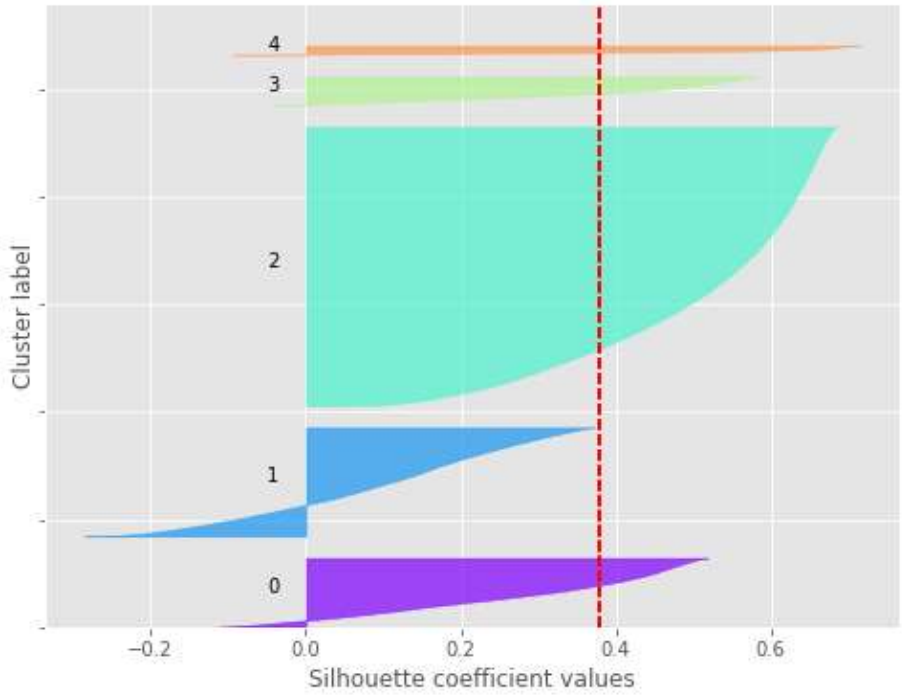

Figure 3. 16: Silhouette Plot for Model 7

For this model clusters 0, 1, 3 and 4 have some silhouette coefficients below 0. However, only cluster 1 has a significant number of records with coefficients below 0. And for this cluster all records have coefficient scores below the average. This means that the classifications for cluster 1 are the least reliable and that some records associated to this cluster could actually belong to another cluster.

Despite not being a completely satisfactory model, model 7 seems to be the most adequate for this data. It has the highest silhouette average score and it generates reasonably interpretable clusters.

3.1.5. Analysis of the Selected Model and Profiling

Model 7 has been chosen as the one that generates the best results for this dataset. The centroids for this cluster are represented in the table below.

Cluster	DAYS_PLAYED	AVG_HOURS_PLAYED	AVG_WIN_PERC	AVG_AMOUNT_VAR	AVG_BALANCE
0	15.05	2.47	0.34	0.28	26.94
1	56.30	3.86	0.18	0.37	-9.21
2	15.28	2.02	0.16	0.20	-7.55
3	191.86	3.87	0.17	0.44	-8.24
4	46.50	3.15	0.20	18.33	-125.04

Table 3. 20: Centroids for Model 7

The records of this dataset are distributed through the clusters as described in the following table.

Cluster	Number of Records
0	12.795
1	20.246
2	51.865
3	5.414
4	1.701
Total	92.021

Table 3. 21: Number of Records in Each Cluster of Model 7

By performing the profiling of these clusters to identify the characteristics of each one, the following profiles were defined.

Cluster	Cluster Name	Description	Compulsive Behavior	Proportion of Cluster
0	Winning Player	Plays sporadically and has a high percentage of wins. This leads to positive balances for the player.	Low	14%
1	Small Loser	Plays once a week and for many hours. Loses small amounts of money and the amounts do not vary much during the day.	Medium	22%
2	Sporadic Player	Plays very sporadically and for few hours each day. Loses small amounts of money and the values of the bets do not vary much during the day.	Low	56%
3	Frequent Player	Plays with a high frequency and for many hours each day. Does not lose much money and the values of the bets do not vary much during the day.	High	6%
4	High Value Player	Plays once a week. Loses a high amount of money and the betting values vary a lot during the day.	High	2%

Table 3. 22: List of Profiles

The profiles were defined based on the tendency to have compulsive gambling behavior. Clusters 3 and 4 are the ones that represent players with high probability of having addictive gambling habits. Cluster 3 represents players that play with a high frequency and cluster 4 represents players that lose large monetary values. These players with high compulsive behavior represent 8% of the users in the dataset.

As it has been noted previously, there are differences in user characteristics based on the entity in which they played, probably due to differences in business and marketing strategies applied by each entity.

Figure 3.17 below shows the number of players associated to cluster 4 distributed by entity. This helps in identifying which entities have more players associated to higher compulsive gambling habits.

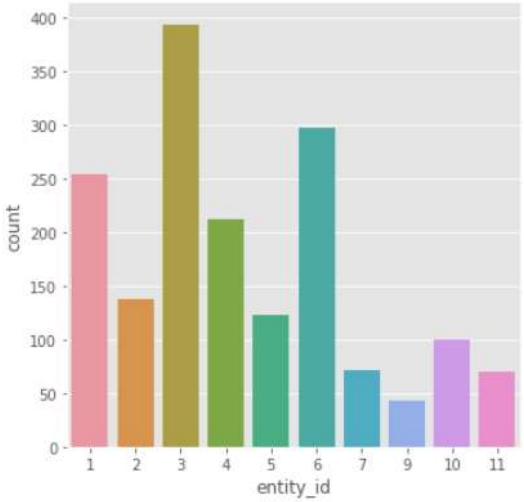

Figure 3. 17: Distribution of Cluster 4 by Entity

The figure above indicates that entity 3 has more players associated to cluster 4, followed by entities 6 and 1. This, however, indicates only which entities have more users associated do cluster 4 in absolute numbers. It is important to analyze this in proportional terms. So, the table below shows the percentage of users of each entity associated to each cluster.

Entity/Cluster	0	1	2	3	4
1	3,13%	21,53%	66,78%	6,95%	1,61%
2	7,77%	21,80%	62,57%	6,70%	1,16%
3	13,76%	26,80%	48,31%	9,04%	2,09%
4	0,81%	20,33%	72,21%	5,39%	1,27%
5	7,42%	25,12%	58,97%	5,93%	2,56%
6	43,17%	25,17%	22,63%	5,58%	3,45%
7	47,81%	15,68%	32,46%	2,26%	1,79%
9	56,05%	11,62%	26,76%	0,36%	5,21%
10	2,39%	14,06%	81,58%	0,08%	1,90%
11	39,59%	18,35%	39,74%	0,99%	1,33%
Total	13,90%	22,00%	56,36%	5,88%	1,85%

Table 3. 23: Proportional Distribution of Clusters by Entity

This table shows that entities 5, 6 and, especially entity 9, have a representation in cluster 4 that is above the total percentage value of 1,85%. This is an indication that these entities have more users associated to compulsive gambling behavior than the other entities. As it has been pointed this is probably part of a strategy applied by these entities to engage users in this type of gaming. However,

only a deeper study of this subject could help reach this type of conclusion. The objective in the case of this work is just to identify different types of user behavior and their characteristics.

3.2. Cluster Analysis for 2020 Data And Comparison with Data From 2019

This portion of the work will be dedicated to the evaluation of data from 2020. This data will also be compared to the data from 2019 with the objective of identifying differences in user behavior from one period to the other.

3.2.1. 2020 Data Loading and Preparation

As with the data from 2019, the data from 2020 was delivered by SRIJ and consolidated by the staff of NOVA IMS into the form of daily records. The consolidated file was delivered in the form of a CSV file with 1.758.745 records.

This file contains data only from the months of April, May, and June of 2020. This is due to the fact that the objective of this work is to evaluate changes in user gambling behavior during the pandemic. Since lockdown restrictions were implemented halfway through the month of March and most of these restrictions were lifted by the end of June, it was found appropriate to analyze only these specific three months.

The file contains the same variables as the file with data from 2019, with two exceptions:

- PLAYER_UID
- SELF_EXCLUDED

Considering that these two variables were not utilized in the selected model for the data from 2019, this is not a reason for concern.

Another difference in this file is that it contains data from two additional entities (numbered 12 and 15) that were not present in 2019. These two entities became active in 2020, either because they were only allowed to explore online gambling in that year or because they chose to start exploring this type of activity in 2020. In any case, since the objective of this work is to compare user behavior between two periods, it has been decided that the data regarding these two additional entities will be disregarded.

Additionally, it was identified that around 5% of the records in this file had the AMOUNT variables equal to zero. And since the AMOUNT_VAR variable is of high importance to the model that was chosen, all records that have been identified with this issue will be excluded.

Other than these issues reported above, all the same procedures that were executed in terms of data preparation for the data from 2019, were also executed with the data from 2020. This includes feature engineering, data aggregation and outlier treatment. The aggregation process reduces the data to a dataset of 177.448 records, each associated to a unique user.

The scope limitation of the data from 2020 was based on the number of days of the year in which players placed at least one bet, the same strategy that was applied to the data from 2019. Table 3.24 below shows the distribution of users according to the frequency in which they played.

Days Played	Percentage
(0,1]	33%
(1,6]	34%
(6,10]	9%
(10,50]	20%
(50,90]	4%

Table 3. 24: Distribution of users according to frequency of play (2020)

As the table shows, 67% of users were only active in online gambling in six days or less during 2020. In order to perform the same type of analysis that was executed for the data from 2019, users that were active in six days or less will be excluded from the dataset. This reduces the size of the dataset to 63.707 records.

3.2.2. Application of the Clustering Model to the 2020 Data

In this section the clustering model that was considered the most appropriate for the clustering of the data from 2019 will be applied to the data from 2020. This corresponds to Model 7 which has the following configuration:

- Algorithm: K-Means
- Variables: Excluding ENTITY_ID, SELF_EXCLUDED and AVG_DAY_TYPE; keeping only one AMOUNT variable; keeping two FREQUENCY variables; keeping one WINNING variable; excluding all PERIOD OF DAY variables
- Scaler: Normalization (Minmax)
- Outliers: Outliers will be limited to the 99^{th} Percentile

An observation that must be made is that since one of the main objectives of this analysis is to compare data from 2019 to data from 2020 and since data from 2020 is only available for three months, the DAYS_PLAYED variable has been multiplied by four. By doing this it will be possible to compare clusters in terms of intensity of play based on the same proportions.

The elbow graph for this model is show on Figure 3.18.

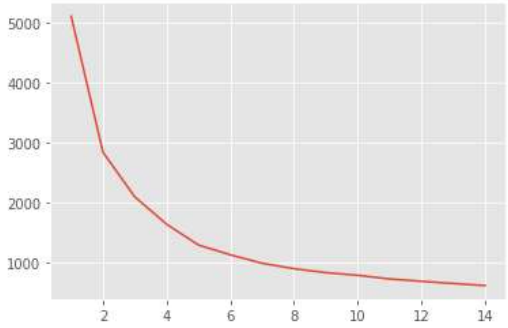

Figure 3. 18: Elbow Graph for Model 7 (2020)

The elbow graph indicates that the number of clusters could be between 3 and 5. So, in order to be coherent with the application of the model for the data from 2019, the model will be applied to the data from 2020 with 5 clusters. The centroids for these five clusters are represented in the following table.

Cluster	DAYS_PLAYED	AVG_HOURS_PLAYED	AVG_WIN_PERC	AVG_AMOUNT_VAR	AVG_BALANCE
0	133.12	3.13	0.17	0.38	-6.66
1	46.33	2.35	0.17	0.29	-6.97
2	262.27	4.33	0.17	0.54	-9.36
3	103.06	3.39	0.20	26.74	-138.89
4	47.65	2.16	0.47	0.21	14.21

Table 3. 25: Centroids for Model 7 (2020)

The records are distributed through the clusters as described in the following table.

Cluster	Number of Records
0	14.720
1	37.234
2	6.467
3	1.079
4	4.207
Total	63.707

Table 3. 26: Number of Records in Each Cluster of Model 7 (2020)

As a result of the analysis of each of the clusters, the profiles listed on Table 3.27 were identified.

Cluster	Cluster Name	Description	Compulsive Behavior	Proportion of Cluster
0	Frequent Small Loser	Plays twice a week. Loses small amounts of money and the amounts do not vary much during the day.	Medium	23%
1	Sporadic Player	Plays once a week and for few hours each day. Loses small amounts of money and the values of the bets do not vary much during the day.	Low	58%
2	Frequent Player	Plays with a high frequency and for many hours each day. Does not lose much money and the values of the bets do not vary much during the day.	High	10%
3	High Value Player	Plays twice a week. Loses a high amount of money and the betting values vary a lot during the day.	High	2%
4	Winning Player	Plays once a week and has a high percentage of wins. This leads to positive balances for the player.	Low	7%

Table 3. 27: List of Profiles (2020)

As it was done with the data from 2019, the profiles were defined based on the tendency to have compulsive gambling behavior. In the case of the data from 2020, clusters 2 and 3 are the ones that represent players with high probability of having addictive gambling habits. Cluster 2 represents players that play with a high frequency and cluster 3 represents players that lose large monetary values. These players with high compulsive behavior represent 12% of the users in the dataset.

As it was noted with the data from 2019, there are differences in user behavior based on the entity in which they played. The table below shows the percentage of users of each entity associated to each cluster.

Entity/Cluster	0	1	2	3	4
1	23,79%	61,26%	12,75%	1,86%	0,35%
2	21,00%	41,37%	9,02%	1,22%	27,39%
3	25,76%	59,21%	10,49%	2,33%	2,21%
4	22,99%	65,89%	9,97%	0,95%	0,20%
5	24,57%	58,15%	11,30%	2,85%	3,11%
6	26,86%	56,38%	11,54%	2,89%	2,33%
7	17,13%	31,71%	5,91%	0,89%	44,36%
9	13,95%	59,69%	3,57%	3,06%	19,73%
10	16,32%	77,96%	4,16%	1,48%	0,08%
11	23,72%	54,70%	11,78%	1,48%	8,32%
Total	23,11%	58,45%	10,15%	1,69%	6,60%

Table 3. 28: Proportional Distribution of Clusters by Entity (2020)

The analysis makes it clear that there are similarities in the clusters that were generated by the model for the data of each year. There are also, however, some differences in a few characteristics of these clusters and in the proportion each one of them represents. These aspects will be analyzed in the Results section of this work.

3.3. Cluster Shifts Between 2019 and 2020

An important analysis that can be made is to verify into which clusters of 2019 the data from 2020 would fit into. By doing this it is possible to verify if users have moved from one cluster to another during the lockdown period of the pandemic. And, in doing so, it is possible to quantify the number of users that have developed compulsive gambling behaviors and those that have either maintained previous behaviors or developed healthier ones.

There are many techniques available to classify datapoints into previously generated clusters. For this work, a supervised learning method called KNeighborsClassifier will be utilized. In this method each datapoint that needs to be classified is compared in terms of distance (Euclidean distance in this case) to all the datapoints in the original dataset and the K nearest neighbors are selected. Afterwards, by a form of vote the new datapoint is classified based on the classification of the majority of its K nearest neighbors (sklearn.neighbors.KNeighborsClassifier -scikit-learn 1.0.1 documentation).

So, in the case of classifying the data from 2020 into the 2019 clusters, KNeighborsClassifier will be implemented with a K of 3. Each datapoint (user) from the 2020 dataset will be compared to the datapoints from the 2019 dataset and the three closest neighbors will be identified. This datapoint will then be classified according to the classification of the majority of these 3 neighbors. So, for example, if two neighbors are associated to cluster 1 and one neighbor is associated to cluster 2, the 2020 datapoint will be associated to cluster 1.

After doing this, it will be possible to compare each user based on its original cluster in 2019 and on the cluster it would be associated to based on its variables in 2020.

And so, the following table shows the cluster shifts of the 31.615 users that were present in both 2019 and 2020 datasets. It summarizes the number of users in each of the possible cluster combinations in the years 2019 and 2020.

Cluster Shifts		2020				
		0	1	2	3	4
2019	0	882	996	638	511	57
	1	726	4.444	1.924	3.529	200
	2	1.054	4.145	4.975	2.090	173
	3	114	1.122	253	3.085	85
	4	31	132	74	89	286

Table 3. 29: Cluster Shifts

This analysis only contemplates the users that were present in both the 2019 and 2020 datasets, which is a total of 31.615 users, since it would not make sense to include users that were only active in one of the years in order to identify cluster shifts. And by analyzing this table, a few insights can be reached:

- 13.672 (43%) of users did not change to a different cluster in 2020.
- 6.560 (21%) of users moved from clusters of low (clusters 0 and 2) or medium (cluster 1) compulsive behavior to clusters of high (clusters 3 and 4) compulsive behavior.
- 1.726 (6%) of users moved from clusters of high compulsive behavior to clusters of either low or medium compulsive behavior.
- 5.141 (16%) of users moved from clusters of low compulsive behavior to a cluster of medium compulsive behavior.
- 4.444 (14%) of users moved from clusters of medium compulsive behavior to clusters of low compulsive behavior.

A few conclusions can be reached immediately by looking at these numbers, but they will be analyzed more objectively in the Results section of this work.

4. RESULTS AND DISCUSSIONS

This section of the work will be dedicated to the discussion of the final results and to answering the research questions that were presented in the Introduction.

One of the main objectives of this work was to classify online gambling users into clusters based on their behavior. Through the application of ML techniques, it became evident that these users can be distributed based on some of their gambling characteristics.

Initially, ML techniques were applied to user data from 2019 which was divided into five clusters. Theses clusters were then analyzed with the intent of identifying profiles according to the tendency of users having compulsive gambling behavior. Of the five clusters, two were associated with high compulsive behavior, either because of the frequency of play or because of the extremely negative user balance. These two clusters represent roughly 8% of the total user base in 2019.

The ML model that was used with the 2019 user base, was also applied to the data from 2020 in order to evaluate what user behavior would look like during the most restrictive period of lockdown in the Covid-19 pandemic. The data was, likewise, divided into five clusters that were analyzed according to the tendency of having compulsive behavior. Approximately 12% of users were associated with high compulsive behavior, which represents an increase by 50% when compared to user behavior in 2019.

Another conclusion that can be reached by analyzing and comparing clusters from 2019 and 2020 is that clusters from 2020 seem to be more extreme. This can be noticed by analyzing, specifically, the clusters associated with high compulsive behavior.

In 2019 the cluster that represents users that have extreme negative balance values has a centroid in which the average balance value is -125,04. In 2020 the equivalent cluster has a centroid in which the average balance value is -138,89.

This can also be noticed in the cluster that represents users that play with extreme frequency. In 2019 this cluster has a centroid in which the average number of days played is 191,86 and the average number of hours played is 3,87. In 2020 the equivalent cluster has a centroid in which the average number of days played is 262,77 and the average number of hours played is 4,33.

Finally, when analyzing users that were present in both the 2019 and the 2020 datasets, it was possible to identify the percentage of users that shifted to different clusters. By doing this, and considering the 2019 clusters as the baseline, it was observed that, during the lockdown period, 43% of users did not change clusters. Of the remaining users, 37% of users shifted to clusters associated to more compulsive gambling behavior and 20% of users shifted to clusters associated to less compulsive behavior.

To summarize, the results of this work, allow us to reach the conclusion that, during the most restrictive period of the lockdown, online gambling users, that are associated to compulsive behavior, became more representative and, on average, may have developed even more extreme habits.

In the following section, the research questions that have guided this work will be answered, in an attempt to wrap up the most relevant objectives that were initially established.

4.1. ANSWERING RESEARCH QUESTIONS

1. **Is it possible to identify specific user behaviors (clusters) based on the data from 2019 that can establish online gambling disorders?**
   By implementing unsupervised ML techniques to user data from 2019 it was possible to identify clusters of users based on gambling habits.

2. **Also based on the data from 2019, is it possible to identify a cluster of users that could be associated with online gambling problems?**
   By performing the profiling of the clusters, it was possible to identify clusters of users clearly associated to more compulsive gambling behavior. One of these clusters is associated to a higher frequency of gambling and another is associated with a tendency to lose larger monetary amounts.

3. **By comparing the segmentation results from 2019 and 2020 during the lockdown restrictions period, has there been a significant shift in cluster sizes?**
   By comparing the clusters that were generated with data from 2019 and clusters that were generated with data from 2020, it is possible to verify that the percentage of users associated to clusters that indicate addictive behavior increased from 8% to 12%. This indicates that these clusters had a significantly higher representation in 2020, at least during the months of April, May, and June, which were the months in which harsher lockdown restrictions had been imposed in Portugal.

4. **Did players, on average, engage in more intense gambling activities during the lockdown restriction periods?**
   It was verified that, on average, users in 2019 played for 3 days each month and for 2.6 hours each day. In 2020 users played, on average, for 7 days each month and for 2.7 hours each day. This means that users engaged in more intense gambling activities during the months of April, May and June of 2020 when compared to the gambling intensity in 2019.

5. **During the lockdown restrictions period, is it possible to determine if a certain number of players from other user clusters migrated to clusters of users with potential gambling disorders?**
   During the lockdown period 37% of users migrated to clusters associated to more compulsive gambling behavior.

5. RECOMMENDATIONS FOR FUTURE WORK

Based on what was developed during this work it is possible to build upon the acquired knowledge to undergo future improvements. A few recommendations will be listed below:

- This type of analysis can be applied to other equivalent types of online gambling activities, such as sporting bets and other kinds of games of chance.

- The analysis can be thought of in terms of individual gambling records instead of consolidated user activities. By doing this it would be possible to evaluate gambling habits based on its variation through time. This, however, would require extremely higher computational capabilities since it would involve stupendous amounts of data.

- Other types of ML models can be applied and the ones that were applied can be improved with alternative configuration of the hyperparameters.

- The models that were applied to the data can generate better results and greater insights if the data is complemented with categorical variables regarding the users, such as: age, gender, marital status, geographic location, and education level. With this type of information, users can be better classified, and this would help in identifying specific groups of users that are more susceptible to compulsive gambling behavior.

6. BIBLIOGRAPHY

Aggarwal, C. C. (2017). *Outlier Analysis.* Springer.

Alam, M. (2020). *Statistical techniques for anomaly detection - Five statistical tools for rapid assessment of anomalies and outliers.* Retrieved June 24, 2021, from Towards Data Science: https://towardsdatascience.com/statistical-techniques-for-anomaly-detection-6ac89e32d17a

Amelia, A. (2018). Retrieved August 30, 2021, from Towards Data Science: https://towardsdatascience.com/k-means-clustering-from-a-to-z-f6242a314e9a

Bação, F., Lobo, V., & Painho, M. (2005). Self-organizing Maps as Substitutes for K-Means. In: Sunderam V.S., van Albada G.D., Sloot P.M.A., Dongarra J. (eds) Computational Science – ICCS 2005. ICCS 2005. Lecture Notes in Computer Science, vol 3516. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11428862_65

Bhandari, A. (2020). *Feature Scaling for Machine Learning: Understanding the Difference Between Normalization vs. Standardization.* Retrieved June 27, 2021, from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

Birkett, A. (2019). *How to Deal with Outliers in Your Data.* Retrieved June 25, 2021, from CXL: https://cxl.com/blog/outliers/

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer.

Bock, T. (n.d.). *What are the Strengths and Weaknesses of Hierarchical Clustering?* Retrieved August 19, 2021, from DisplayR: https://www.displayr.com/strengths-weaknesses-hierarchical-clustering/

Britannica. (n.d.). *Artificial Intelligence.* Retrieved June 15, 2021, from https://www.britannica.com/technology/artificial-intelligence

Brownlee, J. (2016). *Machine Learning Mastery With Python - Understand Your Data, Create Accurate Models and Work Projects End-To-End.* Jason Brownlee.

Calado, F., & Griffiths, M. D. (2016). Problem gambling worldwide: An update and systematic review of empirical research (2000–2015).

Clark, L. (2014). Disordered gambling: the evolving concept of behavioral addiction.

Clements, J. (2019). *Introduction to Hierarchical Clustering - Uncovering Structure in State-level Demographic Data in R.* Retrieved August 18, 2021, from Towards Data Science: https://towardsdatascience.com/introduction-hierarchical-clustering-d3066c6b560e

Domingos, P. (2015). *The Master Algorithm - How the Quest for the Ultimate Learning Machine Will Remake Our World.* Basic Books.

European Comission. (2012). *COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS - Towards a comprehensive European framework for online gambling.* Retrieved June 15, 2021, from https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52012DC0596&from=ET

Galarnyk, M. (2018). *Understanding Boxplots.* Retrieved June 25, 2021, from Towards Data Science: https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow - Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media, Inc.

Gupta, A. (2021). *Elbow Method for optimal value of k in KMeans.* Retrieved November 02, 2021, from Geeks for Geeks: https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/

Håkansson, A. (2020). Impact of COVID-19 on Online Gambling – A General Population Survey During the Pandemic. Retrieved June 15, 2021, from https://www.frontiersin.org/articles/10.3389/fpsyg.2020.568543/full

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques.* Elsevier.

Henriques, R., Lobo, V., & Bação, F. (2012). Spatial Clustering Using Hierarchical SOM. In *Applications of Self-Organizing Maps, Magnus Johnsson, IntechOpen.* doi:10.5772/51159

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data Clustering: A Review.

Kassambara, A. (n.d.). *K-Medoids in R: Algorithm and Practical Examples.* Retrieved November 02, 2021, from Data Novia: https://www.datanovia.com/en/lessons/k-medoids-in-r-algorithm-and-practical-examples/

KHAZRI, A. (2019). *Self Organizing Maps (Kohonen's maps).* Retrieved September 03, 2021, from Towards Data Science: https://towardsdatascience.com/self-organizing-maps-1b7d2a84e065

Kho, J. (2019). *Annotated Heatmaps of a Correlation Matrix in 5 Simple Steps.* Retrieved July 01, 2021, from Towards Data Science: https://towardsdatascience.com/annotated-heatmaps-in-5-simple-steps-cc2a0660a27d

Kumar, S. (2020). *Feature Engineering — deep dive into Encoding and Binning techniques.* Retrieved June 29, 2021, from Towards Data Science: https://towardsdatascience.com/feature-engineering-deep-dive-into-encoding-and-binning-techniques-5618d55a6b38

Kumar, S. (2020). *Hierarchical Clustering: Agglomerative and Divisive — Explained - An overview of agglomeration and divisive clustering algorithms and their implementation.* Retrieved August 17, 2021, from Towards Data Science: https://towardsdatascience.com/hierarchical-clustering-agglomerative-and-divisive-explained-342e6b20d710

Kumar, S. (2021). *How to know which Statistical Test to use for Hypothesis Testing? - When to use which test — T-test, Chi-square Test, ANOVA.* Retrieved July 01, 2021, from Towards Data

Science: https://towardsdatascience.com/how-to-know-which-statistical-test-to-use-for-hypothesis-testing-744c91685a5d

LZP, J. (2019). *Distance Measures and Linkage Methods In Hierarchical Clustering.* Retrieved August 18, 2021, from Gitconnected: https://levelup.gitconnected.com/distance-measures-and-linkage-methods-in-hierarchical-clustering-8b7d488d7ebc

Madsen, J. H. (2018). Outlier detection for improved clustering : empirical research for unsupervised data mining. Retrieved from http://hdl.handle.net/10362/34464

Mitchell, T. M. (1997). *Machine Learning.* McGraw-Hill Science/Engineering/Math.

Patel, A. A. (2019). *Hands-On Unsupervised Learning Using Python - How to Build Applied Machine Learning Solutions from Unlabeled Data.* O'Reilly Media, Inc.

Piech, C. (2013). *K Means.* Retrieved August 24, 2021, from Stanford: https://stanford.edu/~cpiech/cs221/handouts/kmeans.html

Radečić, D. (2021). *How to Make Stunning Radar Charts with Python — Implemented in Matplotlib and Plotly - Easily visualize data beyond the 2nd dimension with Radar Charts — implemented in both Matplotlib and Plotly.* Retrieved October 05, 2021, from Towards Data Science: https://towardsdatascience.com/how-to-make-stunning-radar-charts-with-python-implemented-in-matplotlib-and-plotly-91e21801d8ca

Russell, S. J., & Norvig, P. (2010). *Artificial Intelligence - A Modern Approach.* Pearson Education.

Selecting the number of clusters with silhouette analysis on KMeans clustering - scikit-learn 1.0.1 documentation. (n.d.). Retrieved November 02, 2021, from https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Sharma, P. (2019). *The Most Comprehensive Guide to K-Means Clustering You'll Ever Need.* Retrieved August 24, 2021, from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/

sklearn.cluster.Kmeans - scikit-learn 1.0.1 documentation. (n.d.). Retrieved November 02, 2021, from https://scikit-learn.org/stable/modules/clustering.html#k-means

sklearn.neighbors.KNeighborsClassifier -scikit-learn 1.0.1 documentation. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

SRIJ. (2018). *Serviço de Regulação e Inspeção de Jogos*. Retrieved from https://srij.turismodeportugal.pt/pt/

Turismo de Portugal. (2019). *Atividade do Jogo Online em Portugal - 4º Trimestre de 2019.* Retrieved June 15, 2021, from https://www.srij.turismodeportugal.pt/fotos/editor2/estatisticas/Relatorio_4_trimestre_2019_Jogo_Online_PT.PDF

Whelan, C., Harrell, G., & Wang, J. (2015). Understanding the K-Medians Problem.

White, T. (2015). *Hadoop: The Definitive Guide.* O'Reilly Media, Inc.

Widmann, M., & Heine, M. (2018). *Four Techniques for Outlier Detection.* Retrieved June 24, 2021, from Knime: https://www.knime.com/blog/four-techniques-for-outlier-detection

Yiu, T. (2019). *The Curse of Dimensionality - Why High Dimensional Data Can Be So Troublesome.* Retrieved June 28, 2021, from Towards Data Science: https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e

Zair, M., Rahmoune, C., & Benazzouz, D. (2018). Multi-fault diagnosis of rolling bearing using fuzzy entropy of empirical mode decomposition, principal component analysis, and SOM neural network.