NOVA IMS Information Management School

MDSAA

Master's degree Program in
Data Science and Advanced Analytics

DESCRIPTIVE ANALYSIS OF ONLINE ROULETTE GAMBLERS

Segmentation of different gamblers based on their behavior using data mining algorithms

Henrique Maria Dantas Machado Rosa Vaz

Dissertation presented as a partial requirement for obtaining the master's degree Program in Data Science and 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

DESCRIPTIVE ANALYSIS OF ONLINE ROULETTE GAMBLERS

by

Henrique Maria Dantas Machado Rosa Vaz

Dissertation report presented as a partial requirement for obtaining the Master's Degree in Data Science and Advanced Analytics, with a Specialization in Data Science

Supervisor: Mauro Castelli

Co Supervisor*:* Fernando Peres

November 2022

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process, leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

DEDICATION

I would like to dedicate this thesis to my family: my parents, who always had my back and strongly encouraged me to embark on this challenge; my two brothers have always been an example to follow, and may I be the same to them. Additionally, my girlfriend is the person who always drives me forward and finds a way to motivate me. And finally, to my grandparents, who will always be some of the best references for life.

ACKNOWLEDGMENTS

I must mention both of my mentors in the development of this thesis. Fernando Peres and Mauro Castelli advised me as well as possible and were always available to share ideas and thoughts on the project.

ABSTRACT

The popularity of gambling activities has been increasing over the last decades, with online-based gambling being a key driver of its growth due to the ease of accessing online platforms. Consequently, there is a severe concern that the negative social impact of gambling arises, and regulatory agencies are identifying and managing those effects. In this context, a potential solution to address those effects is based on the concept of 'Responsible Gambling', which means playing consciously, with complete control of time and money. The present study aims to segment online gamblers based on their playing behaviors, differentiating groups as much as possible and ultimately identifying a cluster with players of concern. This is achieved using unsupervised learning algorithms such as K-Means, Hierarchical Clustering, or Self-Organizing Maps. By the end of the study, it was possible to identify some insightful groups of players with distinctive behavioral patterns. The information on which this project is based reflects the activity on some of the Portuguese online gambling platforms over 2019. Available data covers multiple aspects such as the gambling institution, type of gambling, player identification, each player's total bets, and the following outcomes of it.

KEYWORDS

Machine Learning; Data Mining; Unsupervised Learning; Clustering; Online Gambling.

INDEX

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

As technology evolves in all known manners, soars the number of venues it provides. In this case, the focus is on the internet and its immersive and versatile nature. Gambling is no exception to the expansion and development of the internet. As a result of that same evolution, the proportions of online gambling also got more relevant, especially in atypical conditions like the case of the emergence of a pandemic. From 2018 to 2019, the online gambling European market felt an increase of around 18% in gross online gambling revenue (€22.2bn in 2018 to €26.1bn in 2019). Considering the total proportion of all-type gambling (land-based and online), in 2019, the online share was around 25%. As expected, this share only rose in the following years: in 2020, the gross online gambling revenue also expanded. However, the most notable change was in the percentage of online gambling compared to combined gambling. If in the previous year that value was 25%, in 2020, it jumped to 38%.
Furthermore, another great indicator of the online gambling expansion concerns the number of licenses provided in Europe from 2018 to 2020. While in 2018 there were 121 entities with authorization to provide online gambling services, that amount jumped to 234 in 2020. It represents nearly double in just a 2-year period (EGBA, 2021). The numbers above indicate how much gamblers are willing to join the online gambling community and sometimes even prefer it over land-based playing.

The easiness of accessing an online casino or betting house compared to land-based services can be very harmful and have severe social consequences. As Clark (2014) says, gambling disorder is already considered a pathological addiction. Not only is it a problem of addiction, but there is also a batch of other issues attached to online gambling: there is not a social component related to it, as opposed to land-based gambling services. Moreover, it makes it easier for a potentially addicted player to conceal how often and much they bet. In addition, while gambling online, there is a slight perception of unlimited money because credit cards enable fast and straightforward deposits.

Despite many disadvantages linked with the emergence and development of online gambling services, it holds a significant advantage. Having gamblers uniquely registered to play online allows the entities to track records of all the actions a player executes. From there, it is possible to generate either personal or group analysis to describe and examine players' behavior.
Having historical data on what players tend to do in online gambling is essential to regulate and control it. Each country has its gambling regulation entity, and in Portugal, it is up to "SRIJ – 2022 Serviços de Regulação e Inspeção de Jogos | Turismo de Portugal" to store those records. It has information from all authorized entities that exercise online gambling services and contains features covering dates, amounts, frequencies, game types, ids, and other quantitative and qualitative indicators.

With all said above considered, it is relevant to perform some pertinent analysis on this data to characterize different types of players and extract useful and maybe crucial insights about gamblers. When accessing a large amount of data, like in the case of this project, the use of machine learning techniques gains central importance. It concedes the investigator a chance to find patterns and trends that could not be found in high-level scrutiny.
The problem in hands is treated as a segmentation type of problem. This sort of problem is typically approached using unsupervised learning/data mining techniques, which is a branch of machine learning. This thesis will further describe the main machine learning techniques (Chapter 2).

As the desired result of applying such techniques in this dissertation, clusters of players are expected to be obtained. Each player must, in the end, be assigned to one specific collection, and its belonging players should be similar and have identical behaviors. In an optimal scenario, a cluster should have as different attributes as possible when compared to other clusters. It will also be a vital point of the project to finally show a cluster of players potentially developing gambling disorders so that those case cases are referred to specific regulation entities.

1.1. RESEARCH QUESTIONS

The main goal of the project is to conduct a scientific and non-biased analysis that will eventually lead to the answers to a set of research questions that represent the problem at hand:

1- **Is it possible to perform segmentation on online gamblers based on their playing behavior and tendencies?**

2- **How is gambling attendance distributed among the different types of players? (Professional, leisure, others)**

3- **Is it possible to identify potential or actual gambling disorders being developed by a gambler based on his playing behavior and tendencies?**

4- **Is playing frequency directly or indirectly linked with gambling problems?**

1.2. OBJECTIVES

In the first instance, the main objective of this thesis is to analyze different playing behaviors and tendencies of online gamblers. From there, it should be possible to perform segmentation and assign the players to different groups/clusters.

With the clusters in hand, the following step is to break through the different classes of online gamblers and find patterns within those groups. The author expects that by the end of the analysis, it is possible to characterize players based on their playing frequencies, amounts, results, and other relevant indicators.

Another important goal of the project is to inspect whether there is or not a relevant group of gamblers that may be developing or have already set potential pathological gambling disorders. Gambling disorders have been considered a significant public health problem for years, and it is estimated that between 0.12% and 5.8% of online gamblers are in the scope of pathological gambling (Brodeur et al., 2021).

The dataset used contains records from January 2019 to December 2019 and regards online gambling activity in Portugal. This data was made available by the Portuguese entity Turismo de Portugal. Although there are several styles of online gambling, this thesis will only focus on online roulette players.

To address the goals of such a task, machine learning will be a crucial tool for the project. In a problem of segmentation, as is the represented one, unsupervised learning techniques must be explored, tried, and combined if needed. In that sense, different clustering

algorithms can be experimented aiming to converge to an optimal or sub-optimal solution. In addition, insightful graphics must be obtained to explain the clusters as well as possible.

1.3. THESIS STRUCTURE AND APPROACH

A meticulous structure must be drawn priorly to define the best possible approach for the proposed assignment. Only having a concrete strategy can the goals be met, and questions can be answered.

This thesis will start with a **Theoretical Background** review in which the fundamentals and theories behind explored algorithms/techniques will be covered.
The concept of machine learning and its different branches will be described in the first place.
After explaining those basilar concepts, the author will approach a Data-Preprocessing stage. Preprocessing is a crucial phase in any data-related project since it is in the data that the answers to the project's questions rely on. All steps of data preprocessing will later be explained. Those phases include data exploration, outlier treatment, feature engineering, scaling, and normalization.
Ultimately, the Theoretical Background will characterize the modeling part. In this phase, each algorithm will be explained and its application to the model is also going to be uncovered. Several techniques were tested, but that will be regarded afterward.

Once the theory is covered, practical **Data Preparation** will be issued. In this section, exploration, outlier treatment, feature engineering and selection and feature scaling will be addressed to the problem.

The following chapter is **Cluster Analysis and Results**. As the name indicates, this is the moment in which the algorithms' outputs are considered. The author will examine infographics to start shaping conclusions. It will be possible to display different performance indicators and decide on the best configurations to hold results.

Moreover, the **Model Selection and Conclusions** section follows when the final clusters are determined and reasoned. In this leg, the objective is to make comparisons and figure out differences between groups of players based on their playing behaviors and tendencies. In this stage, the results are also faced with some previously defined research questions in the interest of finding possible answers.

Finally, the author will share ideas and thoughts on **Recommendations for Future Work** on a related subject. Emerging adversities and obstructions will be enumerated, and also room for further development in similar activities.

2. THEORETICAL BACKGROUND

2.1. MACHINE LEARNING

Being so popular nowadays, it becomes critical to know what Machine Learning is and how to leverage its potential. Machine learning is a statistical approach to studying and making inferences about data that utilizes a variety of algorithms suited for answering different types of questions (Algren, M. et al, 2021). Such approach enables a handful of possibilities on subjects like predictions, segmentation, sentiment analysis and many others. It does that by easily and quickly computing mathematical models that would take too long for humans to do. With different objectives, there are different branches of machine learning to be used, which will be described below.

2.1.1. Supervised Learning

Supervised learning is one of the most common and heard types of learning in machine learning. This type of learning is a powerful method that provides the user with the tools to perform classification, prediction and forecasting algorithms. In supervised learning, there are two primary components which can be called the supervisor (a given response variable, also called a label) and the learning agent (a machine learning algorithm). The learning agent will develop in such a way that it tries to predict how a given set of inputs (explanatory variables) leads to a given output (response variable or label) (Gonsalves & Upadhyay, 2021).
From a practical point of view, the training data would include both the explanatory and response variable when facing a supervised learning problem. From there, the algorithm would be trained based on those observations. The main goal of the algorithm is to be then able to make as good as possible approximations of the response variable to new and unseen records based uniquely on their explanatory variables.

2.1.2. Unsupervised Learning

Contrarily to the above category, unsupervised learning uses only unlabeled/unclassified data. Such an aspect can represent an advantage for some problems once one less requirement is needed to complete this branch of machine learning tasks.
The most frequent unsupervised learning algorithm in all its different techniques is clustering. A clustering algorithm aims to identify and extract meaningful patterns and tendencies in unlabeled/unclassified data (Bouchefry & Souza, 2020). The result of a clustering technique is often a set of classes that can satisfactorily explain the different data groups.
Only the explanatory variables are fed to the model's training data when developing an unsupervised learning algorithm. Based on those observations, it is expected that the model can identify different groups or clusters. The output of such a model will typically be as good as the intra-cluster similarity arises and the inter-cluster similarity decreases.

2.1.3. Reinforcement Learning

Reinforcement learning algorithms work based on a reward-penalty system. This type of learning typically has access to little or no prior knowledge once it does not compare the obtained and actual inputs. Such an algorithm is to evaluate an action or a sequence

of actions, and either rewards it if performed correctly or penalizes it otherwise. The long-term goal of an algorithm like this is to maximize the total reward obtained in the end (Bouchefry & Souza, 2020).

Algorithm	Input Data
Supervised Learning	Labeled
Unsupervised Learning	Unlabeled
Reinforcement Learning	Labeled/Unlabeled

Table 1 - Input fed into different types of algorithms

2.2. DATA PREPROCESSING

Data often comes in different formats. As a result, it is frequently uncleaned and unsuitable for most machine learning algorithms. Therefore, as a developer approaches a data-related subject/project, he must ensure data quality. Several factors must be evaluated to achieve such a state, and some procedures must be considered.

The concept of data preprocessing is the combination of all actions applied to data to make it more complete, consistent, and easily readable for algorithms. Furthermore, by executing this preprocessing precisely, a model is more likely to output reliable and sharp results. The operations linked with data preprocessing will are described in detail below.

2.2.1. Data Exploration

After loading data, there should always be a meticulous exploration. This stage is where a developer has a first glance at the facts. Then, features like the shape of the data frame (the number of variables and rows) and each column's data type are determined. After that, there is still room for diving into some descriptive statistics. Apart from essential python functions (ex.: *.describe(), .head(), .columns()*, and others), pandas profiling can be a handy tool since it provides a massive amount of information on the data.

Still, in the frame of exploration, some data cleaning is considered. The exploration phase also takes care of things like missing values, null values, and other inconsistencies.
Along with dealing with inconsistencies, assumptions and row filtering are made in this stage. This two are commonly done considering each specific project. Data may appear irrelevant or redundant and thus not suitable for the project. In these cases, depending on an appropriate justification, it is acceptable that some records are removed for work consistency.

Once the author completes this part, a new and clean data frame is stored and kept for future work.

2.2.2. Outlier Treatment

An outlier is an individual pattern that substantially differs from most sample instances (Fernandez et al., 2022). The task of outlier treatment/detection is a sizable concern

because outliers can often be strong enough to ruin a machine learning model training and subsequently result in undesired and far from accurate predictions. In the case of an unsupervised learning problem, these misleading observations may result in points being assigned to individual clusters or at least spoil the appropriateness of another group.

The challenge of detecting outliers is an unsupervised learning problem once there is no a priori reference to which points should be considered outliers. Not only is it not known which observations will be regarded as anomalies, but the definition of an anomaly in each case may be subject to uncertainty. However, for the sake of concreteness, several methods were developed by the author in the interest of defining which patterns to classify as outliers in each case. In this thesis, two of them were used and will be enumerated below.

In the first place, manual thresholding of some variables took place. That is done only after analyzing those variables' distributions, obtained through a boxplot. After studying each variable's boxplot, the investigator chooses a reasonable value, and data is then filtered accordingly.

After doing the above, a predefined method was used: the interquartile range (IQR). The IQR measures how wide the distribution is since it contains 50% of the observations of a data frame (or a specific column) (Malato, 2021). Therefore, the first step when filtering data through IQR is defining boundaries that will be calculated using the respective quantiles. The references is this method are Q1 and Q3, which are the 25th and 75th percentiles, respectively.
Figure 1 helps to understand how IQR defines boundaries and filters data. The borderlines created are the upper bound and the lower bound. The first is obtained by subtracting t x IQR from the quantile number 1 (Q1), thus [Q1 – *t* x IQR]. On the opposite side of the boxplot, the author finds the limit by adding t x IQR to the third quantile (Q3), and so [Q3 + *t* x IQR]. As noticed, there is a t value in each of the formulas. t is called the multiplier and determines how much one is willing to filter. Although there is not a correct value for t, it is prevalent to see it being assigned 1.5, as is the case shown in Figure 1.

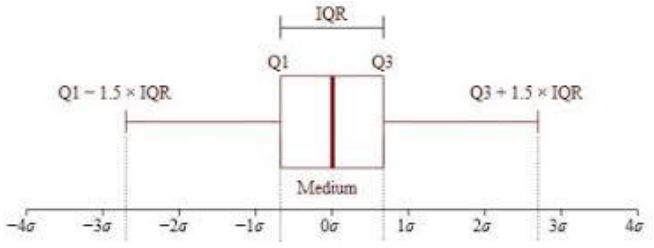


Figure 1 - Interquartile Range Method

Lastly, in machine learning, it is usual to combine more than one method. Combinations allow greater robustness to whatever the application is. In the case of outlier treatment, developers often use it as well. The goal, in this scenario, is to cross results and maintain those that stand for each method applied. Finally, when every filter was used, and the outliers were detected, is the moment to decide whether to keep, remove, or assign new values to those same outliers.

2.2.3. Feature Engineering and Selection

Two critical tasks emerge throughout this data preprocessing subchapter: Feature Engineering (FE) and Feature Selection (FS). The first is how to create new variables deriving from the original ones. On the other hand, feature selection consists of selecting a set of features based on a careful evaluation of their importance and relevance compared to others. (Bocca et al., 2016)

FE may be the first propeller for the FS process. Creating new features based on the original ones can lead to combinations that store information from two or more attributes. The most typical example of this combination is the formulation of ratios. A ratio is no more than a calculation performed between two variables that could prove to be more versatile and valuable than those two variables individually. Still, in the FE section, another precious component is generating new individual features. Besides performing ratios and mathematical combinations, it can be helpful to create flags for different purposes (e.g., 1 if a customer buys product Z, 0 if he does not).
Once FE is complete, all attributes advance for FS, intending to identify how valuable to the model they can be.

Selecting a feature carries a lot of impact and influence on results and predictions. When performing FS, there are constant challenges to face, and the first one concerns the number of features to choose. If the investigator selects too few features, there might be some loss of information associated. On the other hand, if the investigator picks too many variables, there is a high risk of having irrelevant or redundant information in the set. (Tirelli et al, 2011). Following this redundancy issue, another interrogation arises: which combinations of features show more compatibility? In many cases, attributes perform better when modeled with other specific ones, and the opposite can also happen. Some variables are more explanatory as individuals and lose relevance when combined with others.
Besides the abovementioned issues, FS plays a crucial role in computerized performance and model efficiency. The higher the number of features, the higher the expense of computing them. Therefore, FS is also a procedure that must ensure the best accuracy-performance trade-off.

Two main FS criteria come forth in the branch of FS and the case of an unsupervised learning problem.
One is based on the human sense and highly depends on business/project understanding. It is reasonable to concede to an expert on a specific matter the opportunity to point out which attributes better explain a problem. Although the machine is the main protagonist in an ML project, the results will often be evaluated by humans, and it is vital to present explainability.
The second criterion follows the paragraph above and regards relevancy and redundancy. As mentioned before, the user must grant consistency as much as possible. It is known that the inclusion of irrelevant or redundant attributes in a model will recurrently result in decreased prediction accuracy. To find which features may worsen a particular model, the correlation matrix (CM) is commonly used. CM is a matrix of size n x n, n being the number of features in a dataset. Correlations between each pair of variables are obtained by the computer and inserted into the matrix. An example of a result of a CM appears in Figure 2.

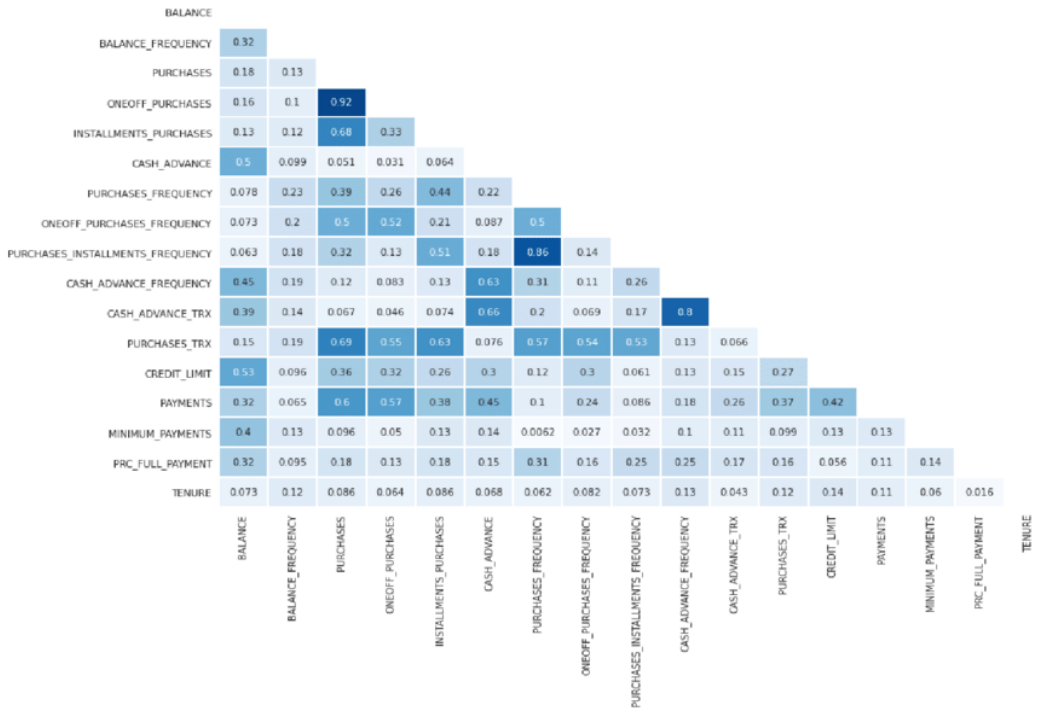


Figure 2 – Example of a Correlation Matrix

Source:

Even knowing many correlations suitable for a CM, Pearson Correlation, r, is the most popular. The formula to reach Pearson Correlation between two variables is:

$$r = \frac{\sum(x - \overline{x})(y - \overline{y})}{\sqrt{\sum(x - \overline{x})^2 (y - \overline{y})^2}}$$

2.2.4. Feature Scaling

Another crucial operation when preprocessing data is to scale or normalize the values in the dataset. This step gains considerable relevance once most ML algorithms base their computations on distances (Euclidean is the most known). In real-life problems, different attributes come in various scales, and their values belong to very different ranges. When features appear on different scales, their contribution to the predictions is expected to vary accordingly. Feature scaling/normalization comes up as a way of equating the magnitude of the features and consequently equalizing their contribution to the calculations (Niño-Adan et al., 2022).

Two main scalers can often be used to reach this contribution balance between features.

Feature Normalization (FN):

The goal with FN is to bring the values of each attribute all into the same predefined range of values. Usually, researchers make values fall between [0, 1]. However, negative values can sometimes carry significant weight; in those cases, the hypothesis of using a range like [-1, 1] gains considerable relevance. Therefore, the following formula is used to calculate the normalized value of a feature:

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

Feature Standardization (FS):

Unlike FN, FS does not bring each attribute to a selected range of values. Instead, it equalizes the contributions by transforming each feature's mean into 0 and standard deviation equal to 1, creating a Standard Normal Gaussian Distribution for every variable. That is achieved with the application of the formula below:

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

In the equation above, μ is the mean of the original feature, and σ represents the corresponding standard deviation.

2.3. MODELING

After preprocessing, data should be near the optimal point. Preprocessing aims to transform data and make it as readable and valuable to the models as possible. When the researcher reaches this stage, the modeling phase follows.

As said above (Chapter 1), this dissertation focuses on a typical problem of unsupervised learning. This type of problem is commonly modeled with clustering techniques. A clustering algorithm aims to create groups of data points that have similarities and can be described based on identical inherent characteristics and patterns. That being the case, each cluster will be pictured based on a selection of features that are considered the most explanatory ones.

There are many clustering algorithms, and choosing one that better fits a problem is attached to different factors. Examples of those factors are the distribution of data in the space, the amount of data available, and the quality and appropriateness of it. These algorithms are also divided into families/methods that regard their computations. All models tried in this thesis will be unfolded in the following points.

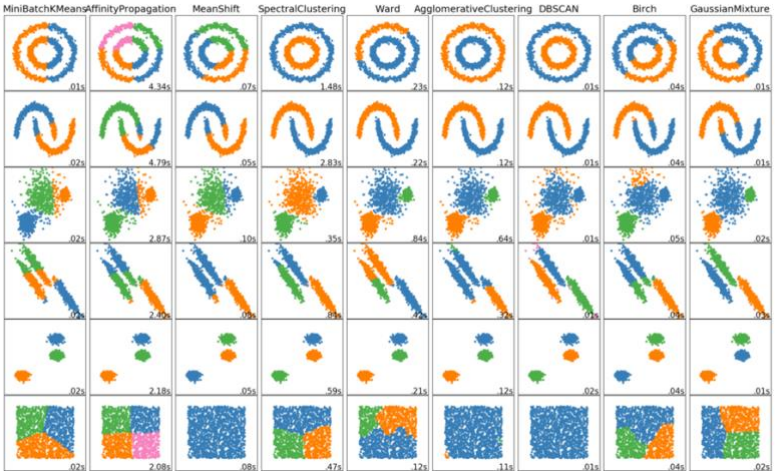


Figure 3 – Effect of Different Clustering Algorithms on Different Data Shapes

Source: https://scikit-learn.org/stable/modules/clustering.html

2.3.1. Partitioning methods

The first genre of clustering methods approached is the partitioning method. These are widely considered the most popular clustering algorithms. The way such methods work is by (typically) minimizing a partitioning criterion, often the distance between points in the same cluster, until a near-optimal clustering solution is obtained. One key feature of partitioning clustering is that the number of clusters commonly needs to be predefined. For that purpose, several techniques can be used and explored in this thesis. K-Means is an example of a partitioning method and was the only one to be tested in this dissertation. The description of such an algorithm will follow below.

2.3.1.1. K-Means

K-Means has proven to be very practical and straightforward within a wide range of partitioning examples, mainly when data is organized in blobs. However, as the algorithm is highly dependent on distances and uses the mean to describe the clusters, it becomes clear that K-Means is very susceptible to outliers.
K-Means operates by completing the following steps.

1. It starts from a point where the number of clusters, k, is defined.
2. After the number of clusters is defined, k random points are selected as the seeds of the groups.
3. Then, each point is assigned to the cluster that stands the closest to it. (Distances are typically computed using Euclidean Distance)
4. When each point has been assigned to a cluster, the centroid of each cluster is updated.
5. Steps #3 and #4 are repeated until no convergence is left.
6. The algorithm is interrupted, and the centroids describe each cluster.

K-Means reaches convergence when the distance between points in the same group (Intra-Cluster Distance) is minimal, and contrarily, the distance between clusters' centroids (Inter-Cluster Distance) is maximal. An image of graphical K-Means iterations is shown below.

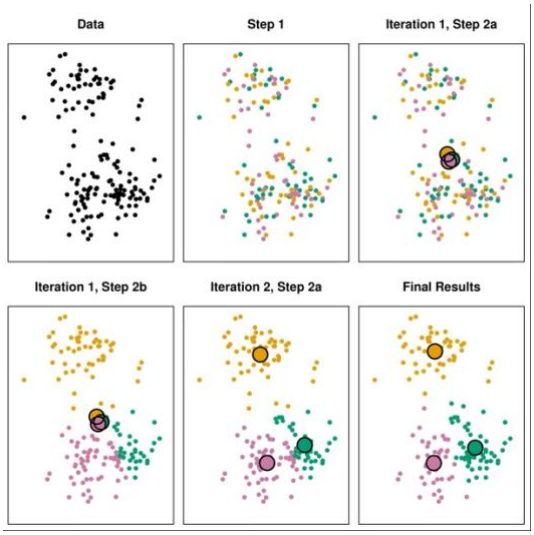


Figure 4 – K-Means Flow Through Each Iteration

Source: https://stackoverflow.com/questions/51263331/kmeans-save-each-iteration-step

As said above, K-Means must be provided the number of clusters, k, before its initialization. The choice k can sometimes be challenging. However, two main procedures can be used. The first one is the Elbow Method. It commonly uses the within-cluster sum of squares value of a solution, which is no more than a ratio between Intra-Cluster Distance and Inter-Cluster Distance. Then, a line chart is drawn, plotting the evolution of inertia through the different values of k. The preferred number of k will be the one where inertia stops decreasing significantly (inflection point).

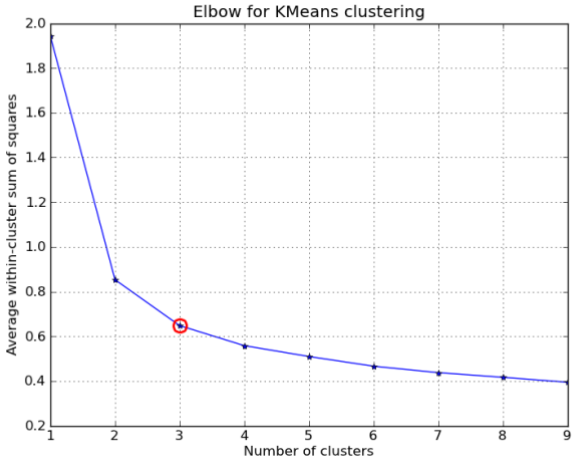


Figure 5 – Elbow Method Representation

Source: https://machinelearninginterview.com/topics/machine-learning/how-to-find-the-optimal-number-of-clusters-in-k-means-elbow-and-silhouette-methods/

Another way of evaluating the adequate number of clusters, k, is by computing each data point's Silhouette Coefficient (SC). The SC measures the appropriateness of a clustering solution. Values for SC fall in the range [-1, 1], where a higher silhouette coefficient refers to a model with more coherent clusters (Belyadi, 2021). More specifically, -1 meaning clusters are assigned in a wrong way, and 1 meaning the clusters are easy to distinguish and well distanced from each other. With that said,

after calculating the SC for solution, an Average Silhouette Score can be attained and describe the suitability of a solution. A complete silhouette plot also shows the information on the distribution of data points per cluster. Some results may show great values of Intra and Inter-Cluster distances but have faulty cluster proportions.

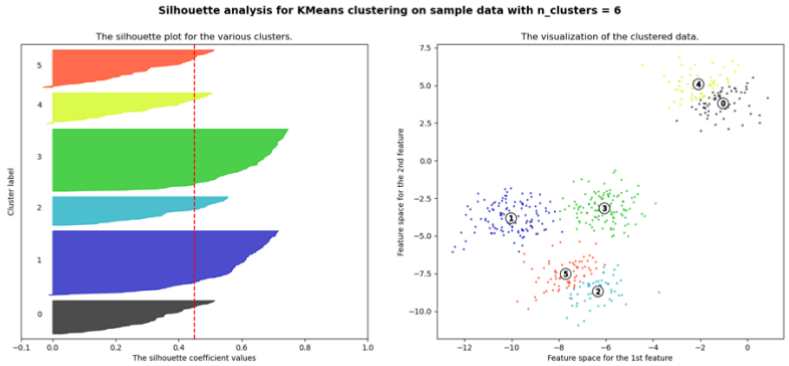


Figure 6 – Silhouette Plot Representation

Source: https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

2.3.2. Hierarchical methods

Another popular method used for clustering is the Hierarchical Clustering (HC) approach. The point standing out the most regarding HC is that, contrary to partitioning methods, the investigator does not predefine the number of clusters in HC. Adding up, HC provides very informative descriptions and visualizations of the possible clustering structure (Liu et al., 2021). However, in terms of computational cost, HC comes out much more expensive due to an extremely high number of calculations that must be done.

HC splits into two categories: agglomerative hierarchical clustering and divisive hierarchical clustering. Only the first one will be explained since it was the unique one used in this thesis.

2.3.2.1. Agglomerative Hierarchical Clustering

This technique is easy to conduct since it relies on simple calculations and repetitions.

1. A distance/proximity matrix is calculated (distance from each point to its peers).
2. The model regards every data point as one individual cluster.
3. Merge the two closest clusters into a single new cluster.
4. Update de distance/proximity matrix using the recently formed cluster.
5. Repeat steps #3 and #4 until one single cluster remains.

During this process, it is fair to question how the distance between clusters is calculated and its influence on the clustering solution. In the scope of this dissertation, the author tested four different linkage types: single, average, complete, and ward's.

Single:

With single linkage, the distance considered as the distance between two clusters is obtained using the smallest distance between any two points (each point belonging to one set only).

Complete:

Contrarily so the previous example, in complete linkage, the distance is the largest one between any two points (each point belonging to one cluster only).

Average:

As the name suggests, average linkage considers the distance as the average distance between every point on a cluster to every point on the other set.

Ward's:

Ward linkage operates by minimizing the sum of squared differences within all clusters. In such cases, combinations of clusters will be chosen based on the sum of squared differences in the formed groups.

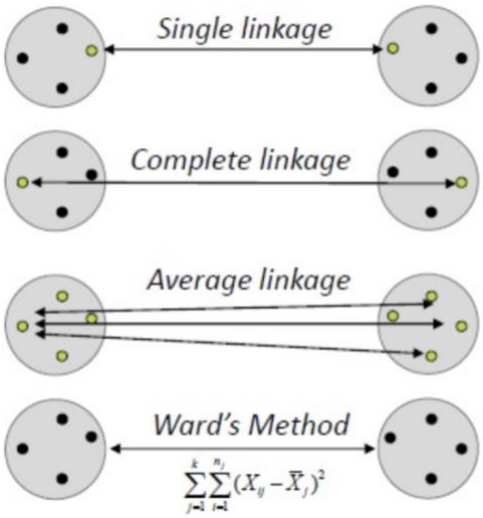


Figure 7 – Different Linkages in Agglomerative HC

After producing the distance/proximity matrix and forming the consequent clusters, the shape of our clustering structure can be obtained through a dendrogram. Dendrograms are visualization tools that illustrate the course of merging in a clustering solution. In other words, it represents which clusters were combined and the distance between every collection. Typically, this graphic is very informative and gives meaningful insights into the adequate number of clusters to use.

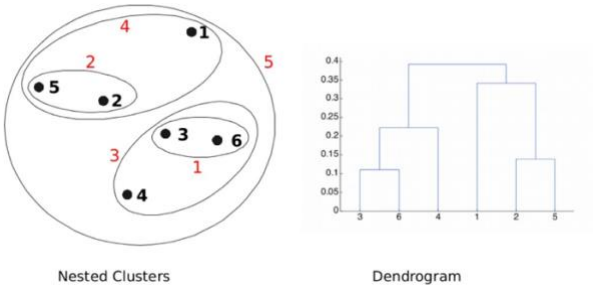


Figure 8 – Representation of Clustering Structure Using Dendrogram

Source: https://slideplayer.com/slide/3287898/

2.3.3. Density-based methods

Density-based clustering algorithms have proved very useful in data mining, especially when locating non-ball-shaped clusters without defining the number of groups in advance (Hu et al., 2021). Another great advantage of density-based algorithms considers the fact that these algorithms are very robust to noise. Such points are often automatically allocated into a category that only considers records classified as noise/faulty points.

However, all advantages come at a cost. These techniques are very computationally expensive, making it harder to run them on large numbers of data. Additionally, when an optimal clustering solution shows two or more close/adjacent clusters, such algorithms may misclassify some points.

Regarding this thesis, two algorithms of this clustering method were tried, and both are described in the following sub-chapters.

2.3.3.1. Mean-Shift

Mean-Shift (MS), like many other clustering algorithms, works through an iterative process. This technique's main objective is to find where the largest concentrations of points are. It is described below how the iterative process of the algorithm acts.

1. Select a random point in the data.
2. Define a circle/window around the selected point, with radius R.
3. Find the mean value of all points within the circle/window.
4. Trace a new circle centered on the above-calculated mean.
5. Iterate through steps #3 and #4 until the mean does not change.
6. Once this iterative process is over, each point should be assigned to a cluster.

One key factor in this algorithm is the radius of the circle/window. That is the only parameter needed to be defined before starting the iteration process. The name of the radius in the scope of MS is called bandwidth. The bandwidth selection directly affects the density estimation (Wu et al, 2007), In other words, this value will influence the calculation of the mean since it is through bandwidth that the size of the circle is determined. Therefore, such a parameter must be carefully picked. Too large values of bandwidth may result in a solution that only finds the global mean of the dataset and, thus, a single cluster at times. On the other hand, too small values may cause the program to point to small insignificant groups.

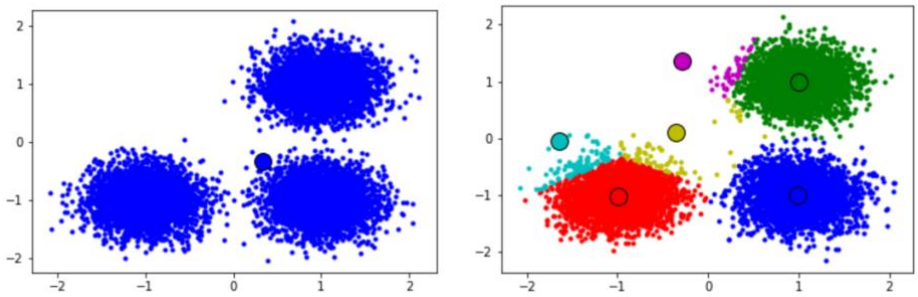


Figure 9 - Representation of Large (left) and Small (right) Bandwidth Values

Source: https://towardsdatascience.com/understanding-mean-shift-clustering-and-implementation-with-python-6d5809a2ac40

2.3.3.2. DBSCAN

DBSCAN stands for Density-Based Spatial Clustering of Application with Noise. As the name indicates, one of this algorithm's main recognized advantages, like other density-based algorithms, is its robustness to outliers. Not only that, but DBSCAN is also carrying a solid ability to detect arbitrary cluster formations in data (Zhu et al., 2021).

In the case of DBSCAN, explaining the technique's parameters is pertinent before explaining the program. There are two necessary parameters to define: minPts and eps (ε).

minPts:

Minimum number of data points belonging to a dense region for it to be considered a cluster

eps (ε):

Epsilon is a distance measure (typically Euclidean Distance) and works as a minimum value to assign a data point to the neighborhood of another.

Following the parameter explanation, some concepts will be clarified before diving into the DBSCAN sequence: core data point, border data point, and noise data point.

Core data point:

A point with at least 'minPts' within a distance of 'ε'.

Border data point:

A point within 'ε'distance from a core data point but is not itself a core point.

Noise data point:

A point that is neither core nor border point.

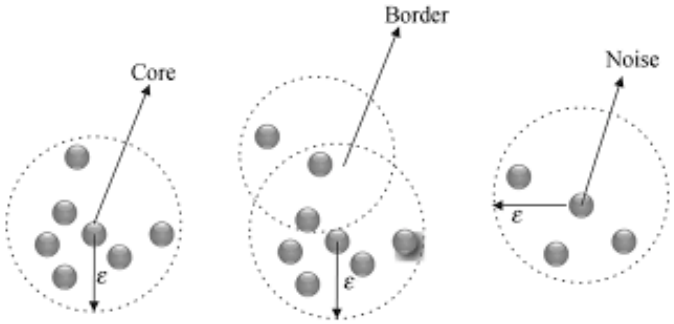


Figure 10 - Different Types of Points in DBSCAN

Source: https://www.researchgate.net/figure/DBSCAN-core-border-and-noise-points_fig1_258442676

Now that all essential aspects of DBSCAN are covered, the steps for the algorithm to proceed are enumerated below:

1. First, a random unvisited point in the dataset is picked.
2. Points within maximum distance 'ε'are considered neighbors.
3. If 'minPts' are found in the neighborhood, the clustering process starts. If not, the point will be considered noise.
4. The process is repeated to all points that keep getting added to the cluster. Continue until every point in the set has been visited by the algorithm.
5. Jump to an unvisited point and repeat from step #2 onward.
6. All points in the and must either be assigned to a cluster or considered noise.

Figure 3, at the beginning of this chapter, show how well density-based algorithms, and in this case, DBSCAN, can perform when facing arbitrary and non-convex data shapes.

2.3.4. Self-organizing maps (SOM)

SOMs are another method for clustering/segmentation among an immense number of techniques. These algorithms have been gaining more prominence due to their versatility and the feedback they carry. A SOM is an unsupervised artificial neural network that can translate high-dimensional data into a two-dimensional space (Tang et al., 2022). This operation brings an excellent advantage to the user considering that one of the most common issues when dealing with multi-dimensional is the difficulty of visualizing data points in that space.

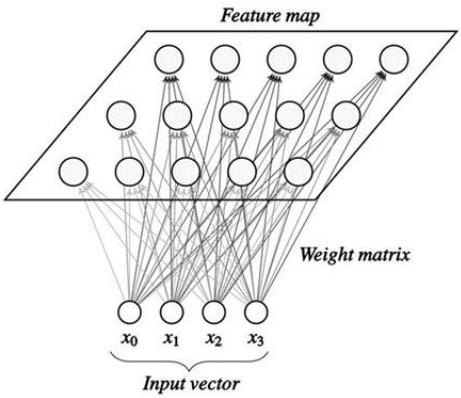


Figure 11 - Example of Input Space Translated in a SOM Feature Map

Source: https://medium.com/machine-learning-researcher/self-organizing-map-som-c296561e2117

The algorithm goes through the following iteration process:

1. The nodes are initialized. The investigator chooses a specific map *a priori*.
2. A data point is randomly picked from the training data.
3. After pairing that data point to every node in the map, the one that most ensembles the given data point is the winning node, also called the Best Matching Unit (BMU).
4. The neighborhood of the BMU is found.
5. Node weights are then updated according to a learning rate. The better a BMU ensembles a given data point, the higher will be the alteration on each node.
6. Repeat from step #2 until every data point is approached.

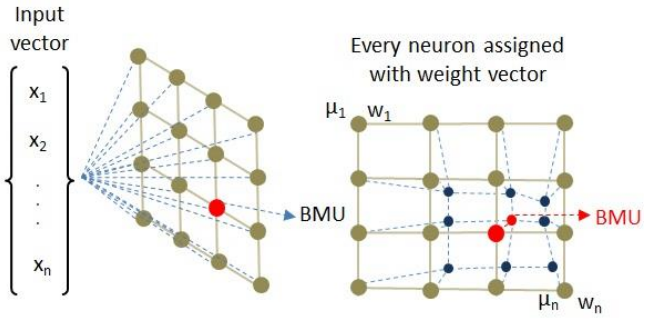


Figure 12 - Graphical Representation of BMU Identification

Source: https://www.analyticsvidhya.com/blog/2021/09/beginners-guide-to-anomaly-detection-using-self-organizing-maps/

Knowing the process and how SOMs operate makes understanding this technique's underlying parameters easier.

Firstly, SOMs usually split into two phases: the unfolding phase and fine-tuning phase. In the first one, the goal is to spread the units in the region of the input space. In this stage, the neighborhood function should use a large radius so the units can move and adapt with adequate freedom. Contrarily, the second phase is used to perform minor adjustments and reduce quantization errors. In such a stage, learning rates and

neighborhood radius are considerably lower; consequently, the time it needs is longer. The length of each of these phases needs to be defined before initialization.

Following the determination of unfolding and fine-tuning phases, the size of the feature map must be decided. This parameter will determine how narrow or expanded the analysis will be. Naturally, the bigger the map, the more detailed the output is. However, the user should manage it carefully. Not only a more extensive map is computationally more expensive, but it may result in insignificant and too personalized output.
After the feature map, parameters like learning rate, neighborhood function, and initialization are also defined. These can vary from a considerable number of options. Even though the effect of such parameters is not as intuitively seen as is the case of the grid size (feature map), all play an essential role in the results obtained.

As mentioned at the beginning of this subchapter, one favorite central point of SOMs is the facilities it delivers to visualize high-dimensional data in lower-dimensional spaces. That can be achieved through a series of tools belonging to SOM.

The first to mention is known as component planes. These graphics inform the user of the intensity of each feature in each node location.

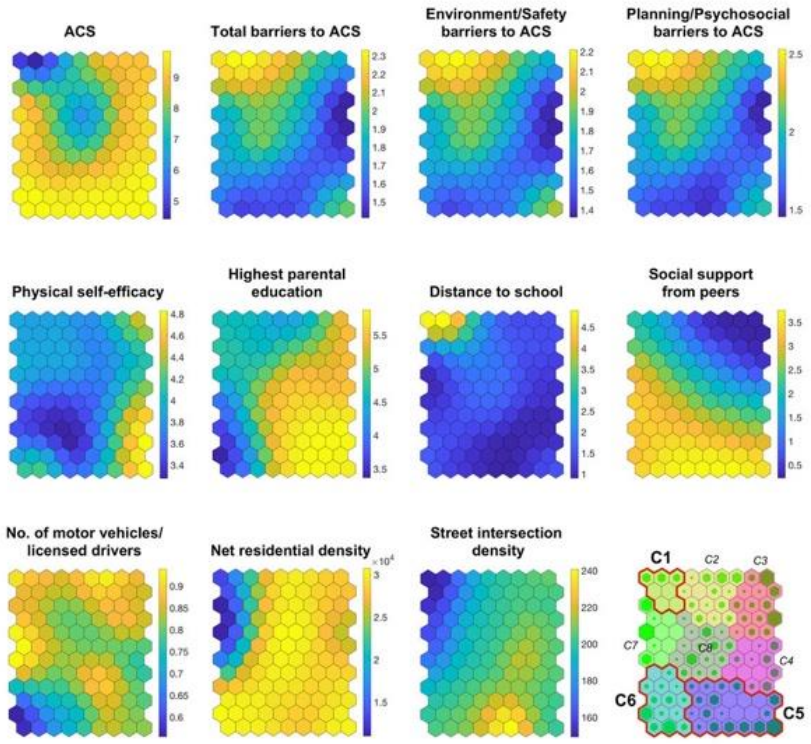


Figure 13 – SOM Component Planes for 11 Features

Source: https://www.researchgate.net/figure/Component-planes-clusters-and-hits-obtained-by-the-Self-Organizing-Maps-approach-Hits_fig1_330031677

Next up, another interesting insight provided by the technique in focus is called the Node Counts. The underlying information of such a plot is the concentration of points assigned

to each node. That way, it is possible to understand which locations are denser in terms of observations.

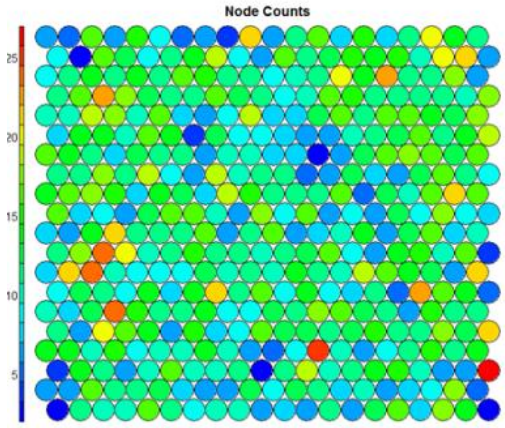


Figure 14 - SOM Node Counts

Source: https://medium.com/@yolandawiyono98/introducing-self-organising-maps-som-2b6af3e9b0ff

Finally, there are other visualizations that a programmer can make use of when working with SOMs. The number of tools such an unsupervised artificial neural network provides is enormous, which is why algorithms like this are vastly used for clustering, dimensionality reduction, data visualization, outlier detection, and other purposes.

2.3.5. Combinations

Although there might be more versatile techniques than others, such a work-always clustering algorithm does not exist. As said above, depending on the shape of data and its distribution, some work-frames suit problems better, and others do not. In cases where no algorithm seems to perform as well as desired, it can be helpful to test combinations of different unsupervised learning algorithms.

In this dissertation, for distinct reasons, single clustering algorithms were not outputting considerably significant results. As a consequence of that, two different combinations were tested. The first regards the use of hierarchical clustering on top of a K-Means solution. The second one contemplates a self-organizing map combined with K-Means.

2.3.5.1. K-Means + Hierarchical Clustering

K-Means and hierarchical clustering are two very different algorithms in terms of operation. When dealing with a large amount of data, the second might be too costly in computations and, in some cases, even impossible to use directly. Nonetheless, if hierarchical clustering must be implemented, a convenient strategy is to use it on top of a primary K-Means solution.
One can apply K-Means to the initial dataset using a very high number of clusters (>> 100). After doing so, a new dataset containing k observations is kept, and it should be a good picture of the initial one. Then, when the number of records is diminished enough, hierarchical clustering can be applied (as described in chapter 2.3.2. of this document).

2.3.5.2. SOM + K-Means

The second combination of methods is a more sophisticated one. Nevertheless, despite that, the logic of its application stands near to the previous one.
SOMs are generally followed by some clustering algorithm. Such a thing happens because of SOM's nature. Typically, SOMs will produce a map containing an ample number of neurons, which in other words, means a large number of clusters. The idea, then, is to apply K-Means on that grid. Since it is a two-dimensional representation of an initial dataset, K-Means should easily compute new clusters based on the distribution of the grid's neurons.
Finally, it is essential to note that a reasonable and meaningful number of clusters should be chosen when K-Means is applied.

3. DATA PREPARATION

3.1. EXPLORATION

This preliminary stage begins with data ingestion. For that, an excel file containing the data used was loaded into a jupyter notebook using the *pandas* library from *Python.*

Before performing any operations on raw data, the shape of the data frame was (777410, 24), the first the number of rows and the latter referring to columns. Following that, the data types of each column were retrieved and are shown in Table 2 below.

Column Name	Description	Data Type
operation	Type of operation	object
entity_id	Entity in which player is operating	int64
player_id	Unique identifier of a player	int64
day_dt	Day of year	int64
day_num	Day of month	int64
day_of_week	Day of week	int64
amount	Total amount spent in all operations in one day	float64
amount_var	Amount variance for the operations of a player in one da	float64
amount_std	Amount standard deviation for operations of a player	float64
hours_played	Total hours played in a day	int64
count	Number of bets placed in one day	int64
wins	Number of successful bets in one day	int64
balance	Net balance for all operations of a player in one day	float64
dawn	Number of bets place during dawn in one day	int64
morning	Number of bets place during morning in one day	int64
afternoon	Number of bets place during afternoon in one day	int64
night	Number of bets place during night in one day	int64
amount_mean	Average bet amount during one day	float64
wins_perc	Percentage of successful bets in one day	float64
dawn_perc	Percentage of bets placed during dawn	float64
morning_perc	Percentage of bets placed during morning	float64
afternoon_perc	Percentage of bets placed during afternoon	float64
night_perc	Percentage of bets placed during night	float64
balance_perc	Net balance in percentage of the amount spent	float64

Table 2 - Data types of raw data columns and feature description

After knowing the data types and using the *pandas .describe()* method, the author conducted a high-level analysis to inspect each feature's fundamental trends and scales. Values like mean, maximum, and minimum can quickly tell if there are significant issues in data or if everything appears normal. Figure 15 pictures some basilar values for each variable.

	count	mean	std	min	25%	50%	75%	max
entity_id	777410.0	3.349455e+00	1.904957e+00	1.00	2.000000e+00	4.000000e+00	4.000000e+00	1.100000e+01
player_id	777410.0	8.569461e+07	1.429763e+08	10002.00	2.183773e+07	7.011631e+07	8.451881e+07	8.385013e+08
day_dt	777410.0	2.019070e+07	3.544657e+02	20190101.00	2.019041e+07	2.019072e+07	2.019102e+07	2.019123e+07
day_num	777410.0	1.519268e+01	8.936800e+00	1.00	7.000000e+00	1.500000e+01	2.300000e+01	3.100000e+01
day_of_week	777410.0	2.957005e+00	1.994477e+00	0.00	1.000000e+00	3.000000e+00	5.000000e+00	6.000000e+00
amount	777410.0	4.972962e+02	6.187239e+03	0.00	3.000000e+00	2.150000e+01	1.205000e+02	2.927858e+06
amount_var	777410.0	6.533665e+02	2.635097e+05	0.00	8.928571e-03	5.712500e-01	6.756372e+00	2.313198e+08
amount_std	777410.0	3.823204e+00	2.527351e+01	0.00	9.449112e-02	7.558108e-01	2.599302e+00	1.520920e+04
hours_played	777410.0	2.038172e+00	1.885080e+00	1.00	1.000000e+00	1.000000e+00	2.000000e+00	2.400000e+01
count	777410.0	4.835532e+01	1.249784e+02	1.00	4.000000e+00	1.200000e+01	4.000000e+01	8.487000e+03
wins	777410.0	2.163546e+01	6.232399e+01	0.00	1.000000e+00	5.000000e+00	1.700000e+01	7.033000e+03
balance	777410.0	3.162824e-02	6.745698e+02	-97486.78	-1.000000e+01	-1.000000e+00	2.000000e+00	1.343660e+05
dawn	777410.0	1.072926e+01	4.872012e+01	0.00	0.000000e+00	0.000000e+00	2.000000e+00	3.274000e+03
morning	777410.0	7.787327e+00	3.751952e+01	0.00	0.000000e+00	0.000000e+00	0.000000e+00	3.195000e+03
afternoon	777410.0	1.625120e+01	5.751287e+01	0.00	0.000000e+00	0.000000e+00	8.000000e+00	4.108000e+03
night	777410.0	1.546267e+01	5.046883e+01	0.00	0.000000e+00	0.000000e+00	9.000000e+00	2.577000e+03
amount_mean	777410.0	6.208877e+00	3.934076e+01	0.00	5.000000e-01	1.650000e+00	4.666667e+00	2.460385e+04
wins_perc	777410.0	4.045092e-01	2.829359e-01	0.00	2.000000e-01	4.000000e-01	5.490196e-01	1.000000e+00
dawn_perc	777410.0	2.236916e-01	3.929566e-01	0.00	0.000000e+00	0.000000e+00	2.671756e-01	1.000000e+00
morning_perc	777410.0	1.424662e-01	3.184927e-01	0.00	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
afternoon_perc	777410.0	3.210642e-01	4.266968e-01	0.00	0.000000e+00	0.000000e+00	8.333333e-01	1.000000e+00
night_perc	777410.0	3.491068e-01	4.432150e-01	0.00	0.000000e+00	0.000000e+00	1.000000e+00	1.000000e+00
balance_perc	770563.0	inf	NaN	-1.00	-4.871795e-01	-1.139601e-01	1.250000e-01	inf

Figure 15 - Descriptive statistics of raw data

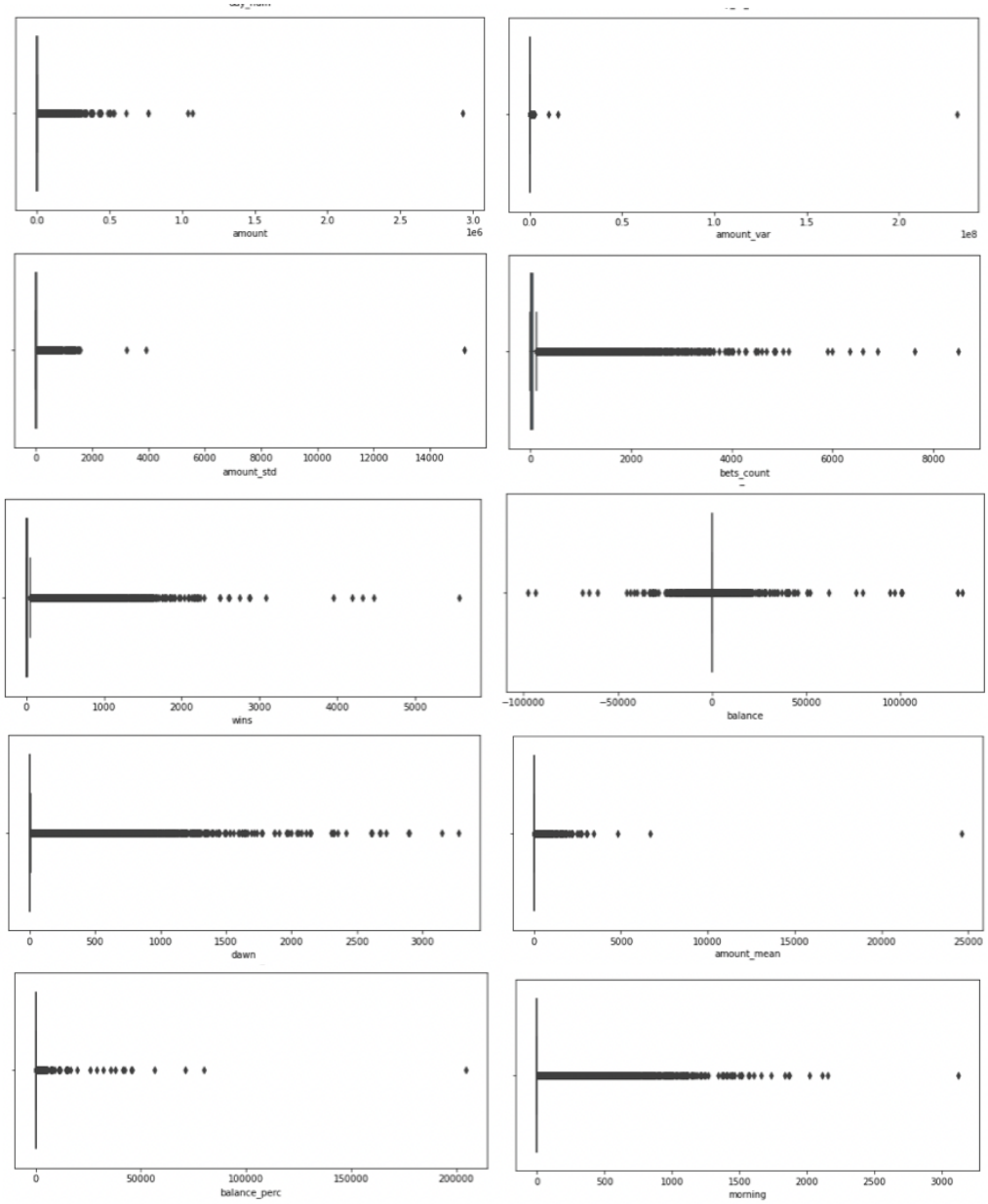
Since there were no apparent structure-wise mistakes in the dataset, a data cleaning process started. The first operation of such an approach is to deal with missing values. In the case of this dataset, only one column contained missing values: *balance_perc*. 6847 rows were missing this value. Considering that only represented a total of 0.89% of the whole data, using *dropna()*, those values were dropped from our observations.

After dealing with missing or null values, considerations regarding the frequency of play were made. For consistency purposes, only players that played at least five times a year (at least more than once each three months) were considered. To do so, an auxiliary data frame was generated, containing the number of days played for each player. Only those containing more than four days were kept. Once occasional players were removed, the data frame had 580665 rows and 25 columns, being *days_played* the additional one.
Another critical assumption made in this early stage of handling data was that operations containing an amount equal to 0 (zero) were disregarded. The reason to infer so is that plays with zero amount are not actual or relevant plays. As a consequence of this assumption, approximately 6000 rows were removed.

3.2. OUTLIER TREATMENT

By looking and Figure 15 in the above sub-chapter, it is possible to conclude that in many variables, there is a great distance between maximum values and the 75th percentile and also between minimum values and the 25th percentile. That is enough to force us to conduct an outlier analysis and treatment.

The author chose a multi-functional approach to deal with this chapter's task. The first part consists of thresholding features according to their distributions.



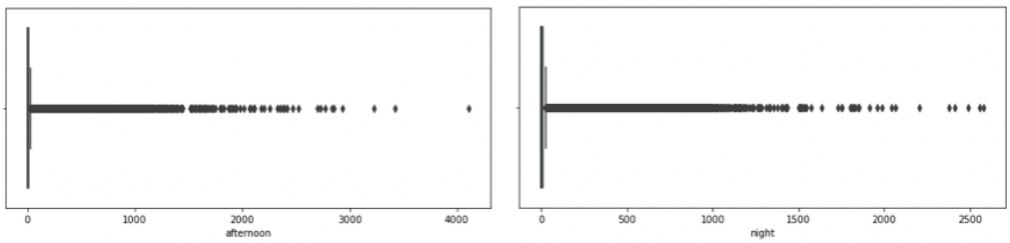


Figure 16 - Boxplots for each filtered feature

These boxplots tell us the amount of dispersion beyond the upper and lower limits. Those values were manually determined in this first instance, and individual and personalized filters were applied.

Lower Limit	Column Name	Upper Limit
-	amount	500000
-	amount_var	50000000
-	amount_std	2000
-	bets_count	4000
-	wins	2000
-50000	balance	50000
-	dawn	1500
-	morning	1500
-	afternoon	1500
-	night	2000
-	amount_mean	5000
-	balance_perc	25000

Table 3 - Manual thresholding values for outliers

Together with the abovementioned manual limitation of data, the interquartile range method (explained in Chapter 2.2.2.) was applied. The objective was to introduce an outlier treatment operation to every feature in the dataset. One important note in this phase is that the two methods were applied exclusively. The interquartile method would prevail if done differently due to a broader expansion of outlier identification.
By the time the two methods were applied and combined, 159 rows had misleading values, with 99% of the data being kept after removing outliers.

3.3. FEATURE ENGINEERING AND SELECTION

This critical stage of any machine learning project starts by ingesting a new dataset that contemplates the removal of the majority of outliers. At the moment of reading that dataset, the shape of it is 580496 rows x 24 columns.

The first step toward a successful Feature Engineering operation is to discuss the hypothesis of adding new features to the frame. It is widespread to have features that individually hold minimal relevance but are very explanatory when combined with others. In this case, a set of features was generated from the original ones in the first approach:

'**is_weekend**': flag with value 1 if a given day corresponds to a weekend day, 0 otherwise.

'**amount_per_hour**': reveals the average amount a player spends per hour.

'**bets_count_per_hour**': indicates the number of bets made by a player per hour.

'**perc_year_played**': percentage of days in a year that a player was active (played)

'**total_amount_year**': total amount spent by a player in a year

Further, the author generated variations of some features due to the high density of extreme values in their distributions. The 95th percentile of each was used as an upper boundary to tackle such an issue. These features' values above the 95th percentile were converted into the 95th percentile. That ensured a more algorithm-friendly distribution and brought better results after scaling data. Examples of such transformations are: 'amount_var_perc95', 'total_amount_year_perc95', 'bets_count_perc95' and 'wins_perc_perc95'. Below this paragraph shows an example of the code used for that transformation.

```
# Set value for 95th percentile of 'wins_perc'
perc95_wins_perc = df['wins_perc'].quantile(0.95)

# Create perc95 variable for wins_perc
df['wins_perc_perc95'] = df['wins_perc'].apply(lambda x : x if x < perc95_wins_perc else perc95_wins_perc)
```

Figure 17 - Code to generate 95th percentile variable

Following a successful introduction of new variables, data needed to be aggregated. In this thesis, it was decided that the optimal granularity to perform the analysis was to see the observations by player (having each player's corresponding mean values for each feature). To reach such a result, the *groupby()* function was used. As a consequence of this aggregation, there are a few variables that automatically lose interest: 'entity_id', 'day_dt', 'day_num', 'day_of_week'. After operating, the resulting data frame held 33003 rows distributed over 28 columns (this happens because 'player_id' works as an index of the data frame).

Afterward, having brought data to the desired scope, feature selection arrives. The first deciding criteria for whether to keep a variable or not were business understanding and common sense. Columns such as the ones mentioned in the paragraph above were immediately discarded. Not only due to a lack of interest and relevance but, more importantly, because of misleading values. Moreover, regarding business understanding, one should consider using variables related to the amount spent and measures of success in the activity due to the nature of the problem.
Besides these straighter-forward examples, every feature must be considered. For that reason, a correlation matrix was produced and inspected (Figure 18).

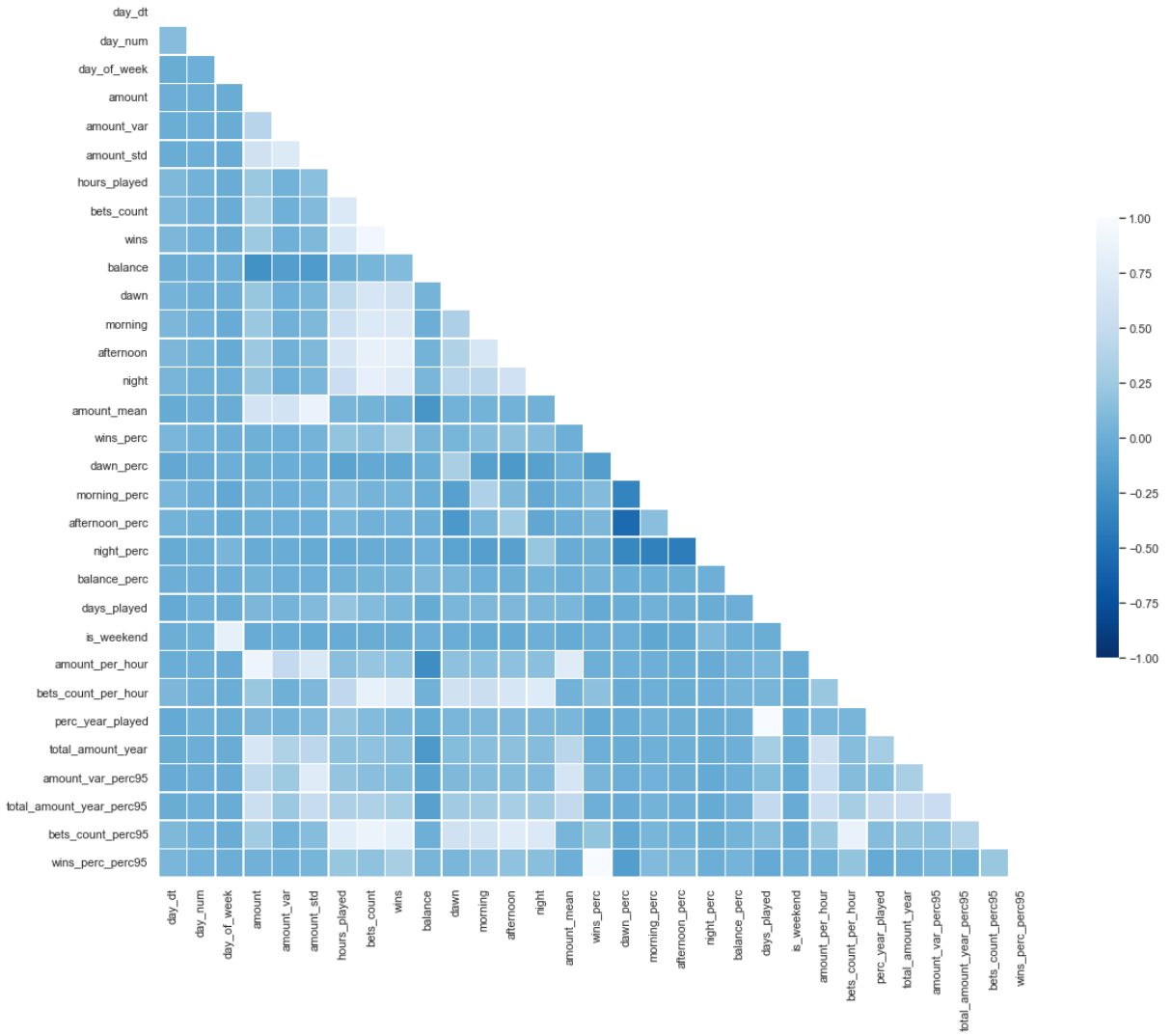


Figure 18 - Correlation Matrix

Although the correlation matrix does not tell precisely at first which set of features should be used, it indeed indicates which attributes should not be fed into a model together. High correlation values (either positive or negative) indicate potential redundancy, an effect that must be avoided at all costs. Another central takeaway when examining a correlation matrix is finding which features correlate to a broader number of others. Typically, when a variable holds a significant correlation with multiple ones, it indicates high explanatory power, and allows one to discard the others. Considering what is mentioned above, the author pointed out some considerations:

- Amount-related features hold significant positive correlations with each other, as expected:
    - ‘amount’ vs ‘amount_std’ : 0,60
    - ‘amount’ vs ‘amount_var’ : 0,41
    - ‘amount’ vs ‘total_amount_year’ : 0,67
    - ‘amount’ vs ‘amount_per_hour’ : 0,89

- The number of bets is highly correlated with several features:

- ‘bets_count’ vs ‘hours_played’ : 0.72
- ‘bets_count’ vs ‘wins’ : 0.93
- ‘bets_count’ vs ‘afternoon’ : 0,85
- ‘bets_count’ vs ‘night’ : 0,83

Simple considerations resulting from a direct study of the correlation matrix provide ease in decision-making. For the case being, it would be logical not to feed any model multiple amount-related features. Additionally, it would be fair to consider 'bets_count' as one key variable for our problem due to its high explanatory power and considerable versatility.

As in most machine learning, and more specifically, data mining problems, multiple experiments must be implemented and bring the decision of a set of features down to their results. In this dissertation, different bags of attributes were tested, and the significant results will be depicted in the Cluster Analysis and Results chapter.

3.4. FEATURE SCALING

Feature scaling (or normalization) follows the same concept as feature selection. Therefore, there is no way of knowing which scaling or normalization method will perform better without trying different ones.

In the following chapters, the differences between using one or another will be determined by each model success, and the Model Section will contemplate the one that achieved better results.

4. CLUSTER ANALYSIS AND RESULTS

After the preprocessing stage, the actual clustering phase is reached. In this chapter, relevant experiments will be depicted with valuable infographics. Additionally, a summary table with all the experiments is visible, in which meaningful specifications, such as features used, scaler choice, and all relevant information for each technique is provided.

In this stage, the reader will be able to evaluate the most relevant and successful tests of the study. Apart from those, every other experiment is further detailed with plots in the Appendix, Chapter 8 of this report.

Table 4 shows the feature bags used in experiments described in this report. Such combinations were obtained either by resulting of the feature selection study (ex.: set 1) or by iterative experiments with the algorithms (ex.: set 7).

Set	Features
Set 1	‘amount_var’, ‘hours_played, ‘balance_perc’, ‘amount_per_hour’, ‘bets_count_per_hour’
Set 2	‘amount_var’, ‘hours_played’, balance’, ‘amount_per_hour’, ‘amount_mean’, ‘bets_count’
Set 3	'amount', 'hours_played', 'bets_count'
Set 4	‘wins_perc, ‘hours_played’, ‘bets_count’
Set 5	‘amount_var’, ‘hours_played’, bets_count, ‘wins_perc’
Set 6	‘wins_perc_perc95, ‘bets_count_perc95’, ‘amount_var_perc95’, ‘perc_year_played’, ‘total_amount_year_perc95’
Set 7	‘wins_perc_perc95, ‘bets_count_perc95’, ‘amount_var_perc95’, ‘total_amount_year_perc95’

Table 4 - Description of each feature set

Algorithm	Trial	Scaler	Feature Set	Scope
K-Means	1	MinMax	1	Year
K-Means	2	MinMax	2	Year
K-Means	3	MinMax	3	Year
K-Means	4	MinMax	4	Year
Mean-Shift	1	MinMax	1	Year
Mean-Shift	2	MinMax	5	Semester
Mean-Shift	3	MinMax	5	Trimester
DBSCAN	1	MinMax	1	Year
DBSCAN	2	MinMax	5	Semester
DBSCAN	3	MinMax	5	Trimester
K-Means + HC	1	MinMax	1	Year
K-Means + HC	2	MinMax	3	Year

K-Means + HC	3	MinMax	4	Year
K-Means + HC	4	MinMax	5	Year
K-Means + HC	5	MinMax	6	Year
K-Means + HC	6.1	MinMax	7	Year
K-Means + HC	6.2	MinMax	7	Year
SOM + K-Means	1	MinMax	7	Year

Table 5 - Trial specification

4.1. K-MEANS

All experiments of this trial are demonstrated in the appendix.

Final appreciation:

Even though further experiments were performed using K-Means, none of the obtained results were considered acceptable as a solution. Furthermore, the clusters produced by each of those trials, including the ones depicted in the Appendix, had no differentiation importance over the different groups. Therefore, K-Means as an individual algorithm was disregarded for this project.

4.2. MEAN-SHIFT

All experiments of this trial are demonstrated in the Appendix.

Final appreciation:

At the end of 3 experiments, manipulating the scope of data and its features, the researcher concluded that Mean-Shift was not applicable in this case. The results achieved indicated very low suitability of the model to the dataset. It could be the case that the shape of the data was not the most indicated. Additionally, this method showed increased computational effort. For those reasons, Mean-shift was disregarded as a possible solution.

4.3. DBSCAN

Continuing in the branch of density-based clustering algorithms, the DBSCAN takes place. As explained in the section of DBSCAN explanation, the first thing to do when using this algorithm is to find the eps value. For that, a K-Distance graphic can be used as an aid. Similar to the other algorithms, this one must also be put into different scopes of data and variables.
All experiments of this trial are demonstrated in the Appendix.

Final appreciation:

DBSCAN also demonstrated huge limitations in finding insightful patterns between different groups. Furthermore, it is reasonable to infer that the shape of data may not be density-based algorithms friendly.

4.4. K-MEANS + HC

The first method combination tested in this project was K-Means, followed by hierarchical clustering. The *modus operandi* to achieve such a thing is detailed in chapter 2.3.5.1.
This section will focus on implementing such a technique, going through the different configurations tested in terms of parameters and feature selection.
The order in which this task is executed is as follows:

1. Within Cluster Sum of Squared Errors for the high k K-Means solution (k = number of clusters).
2. Clustering with K-Means and store the solution as the new dataset to work on.
3. R Squared plot for each different type of hierarchical clustering linkage method.
4. Dendrogram for the chosen linkage and consequent number of clusters choice.
5. Clustering with hierarchical clustering (with the parameters found previously).
6. Cluster profiling

Trial 6:

Scaler: MinMax Scaler (-1, 1)

Features: 'wins_perc_perc95, 'bets_count_perc95', 'amount_var_perc95', 'total_amount_year_perc95'

Results:

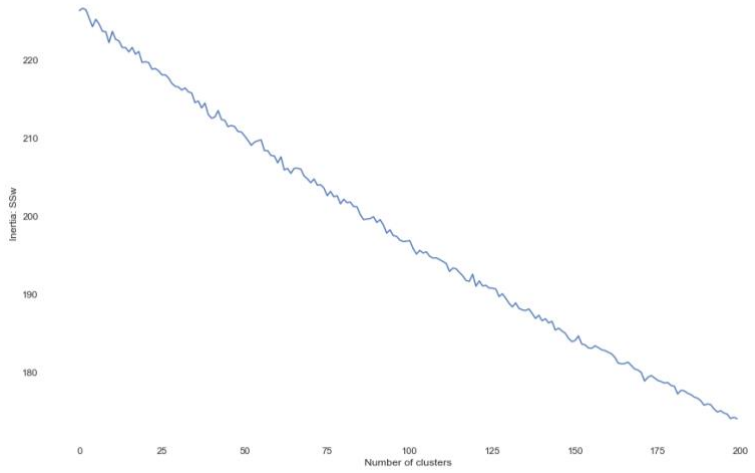


Figure 19 - WCS for K-Means + HC Trial 6

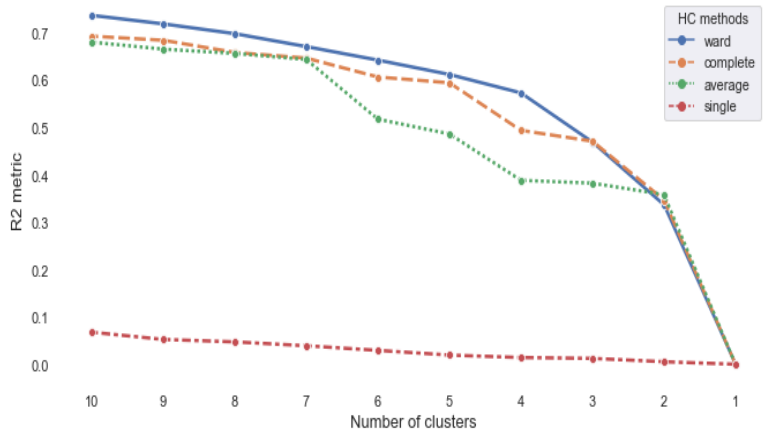


Figure 20 - R Squared for K-Means + HC Trial 6

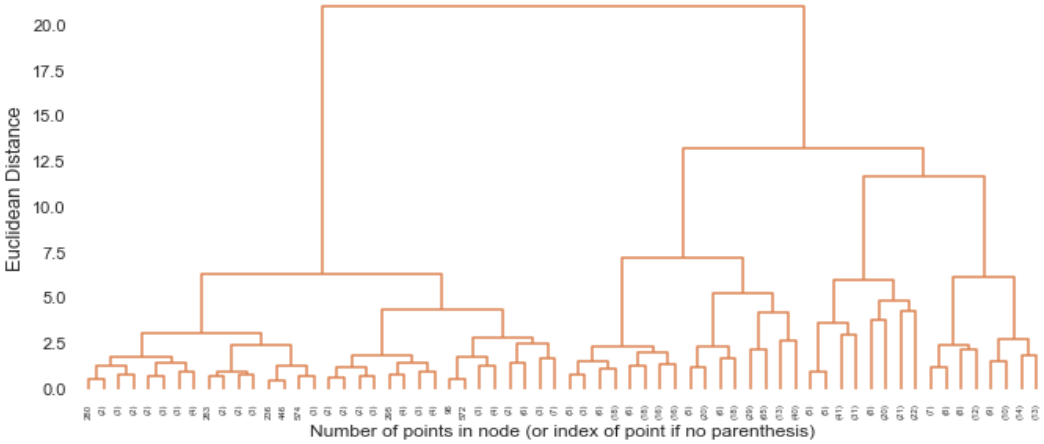


Figure 21 - Dendrogram for K-Means + HC Trial 6

Trial 6.1:

Group: Mean

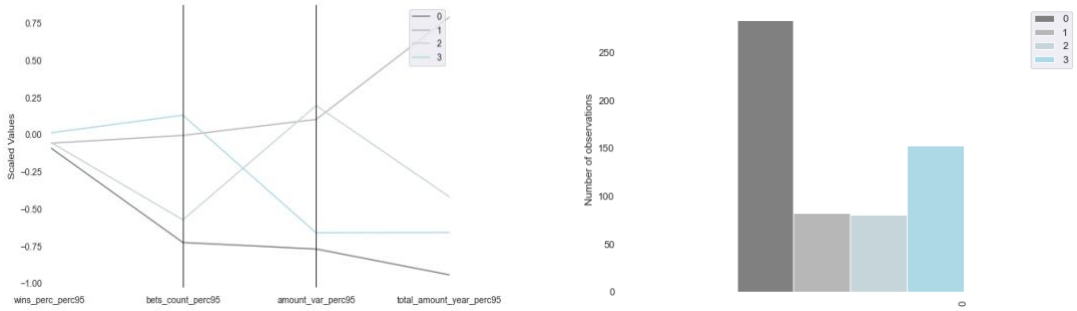


Figure 22 - Cluster Analysis for K-Means + HC Trial 6.1 k = 4

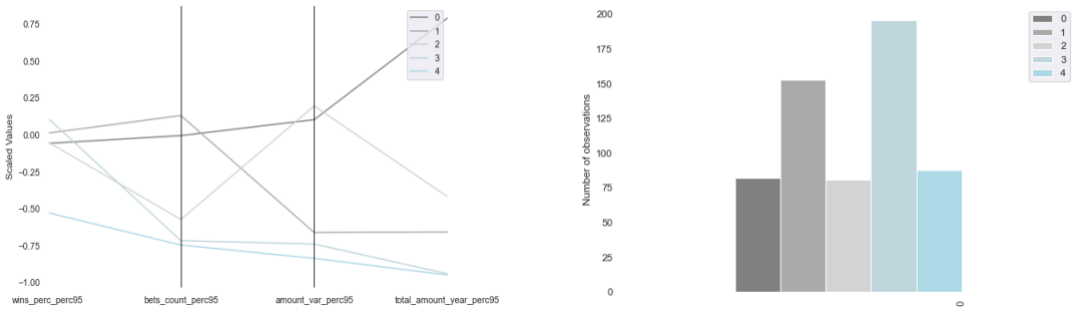


Figure 23 - Cluster Analysis for K-Means + HC Trial 6.1 k = 5

Comments:

Considering the 95th percentile brought considerable benefit to the model's results, that idea was kept in further trials. In this case, the number of features chosen was reduced back to four. The outcome of this experiment is very positive and can be used to explain players' behaviors. This solution adds value to conclusions with four relevant variables in hand and a decent distinction between clusters.

Trial 6.2:

Group: Median

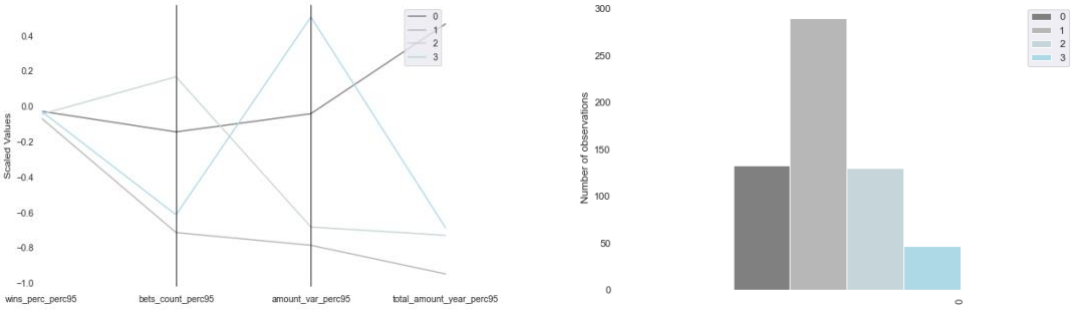


Figure 24 - Cluster Analysis for K-Means + HC Trial 6.2 k = 4

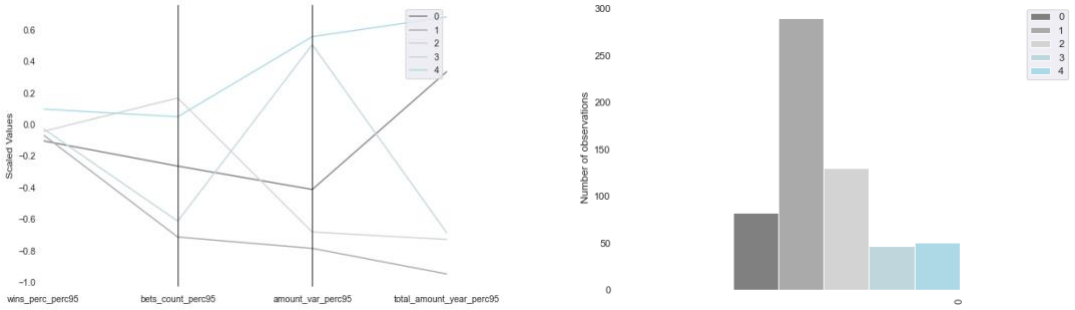


Figure 25 - Cluster Analysis for K-Means + HC Trial 6.2 k = 5

Comments:

The difference between 6.1 and 6.2 is in the aggregating function used to calculate the centroids of each cluster. In the first one, the author chose the mean to achieve such values, while in this case, the median was used to avoid the effect of extreme or irregular values. Nevertheless, the results are similar and considered reasonable solutions to the problem.

Final appreciation:

Without a doubt, there are two standing-out solutions in this subchapter: 6.1 and 6.2. Because 6.2 is less prone to extreme and faulty values in the profiling, it will be the experiment to look into in detail. From both solutions in 6.2, the one with k=5 will be considered to get an additional category for the final result.

Cluster 0:

Cluster 0 is not a predominant class, with only 13,6% of data points belonging to it. Nonetheless, it is one of the clusters that most differentiates from the others in general, and its features are described below.

> 'wins_perc_perc95': The percentage of winning plays for this class of players is the most negative in the whole dataset.
>
> 'bets_count_perc95': In terms of bets placed, this type of player is sitting nearly on average, just a fraction higher than it.
>
> 'amount_var_perc95': The variance in these gamblers' bet values is not too relevant considering the average player. However, it is still far from minimal.
>
> 'total_amount_year_perc95': The amount dispended by players in this cluster is the second most of all observations. These are participants who employ considerable amounts of money, and their average bet value is high, considering their 'bets_count'.

With the results described above, gamblers assigned to this cluster will be denominated **problematic gamblers**. The main traits of a problematic gambler are elevated sums of the amount spent in a year combined with a low success rate.

Cluster 1:

Moving on to Cluster 1, it is the most represented cluster, having almost half (48,3%) of the players assigned to it.

> 'wins_perc_perc95': Following the logic of Cluster 0 above, the winning percentage of players in this cluster is also reduced. Nevertheless, it is also far from entirely negative.

'bets_count_perc95': The number of bets placed by these gamblers is the lowest in the dataset.

'amount_var_perc95': In the case of the feature above, the amount variance of these players is also the smallest.

'total_amount_year_perc95': Finally, the total amount spent by Cluster 1 players is bottom level as well.

After considering each feature, these players will be referred to as **non-regular gamblers**. A low number of bets characterizes such a player, as do a reduced total amount expended, and weak amount variance.

Cluster 2:

In Cluster 2 reside 21,6% of the players, and there is a general trend in this cluster: the low values.

'wins_perc_perc95': As in most cases in this solution, the success rate of this cluster is also diminished. However, not as unfavorable as the previous cluster.

'bets_count_perc95': Players in Cluster 2 have the highest number of bets. That could be meaningfully combined with the total amount they spend on gambling.

'amount_var_perc95': The amount variance held by these players is relatively low, meaning there is no significant reactivity trait to be studied here.

'total_amount_year_perc95': Even though we saw above that players in this cluster are the ones that best the most in number, the same does not happen in amount. That tells us, as opposed to Cluster 0, these gamblers preferer to risk less at each bet. That also goes along with the low variance shown by them above.

Players in Cluster 2 are going to be named **leisure gamblers**. A leisure gambler does not want to take many risks and chooses to spend a more significant amount of time playing. That can be told by a considerable number of bets contrasting with the small amount expended.

Cluster 3:

This cluster contains 7,8% of all players studied, making it a minority class.

'wins_perc_perc95': Along with the other groups studied so far, the winning percentage is also negative but not far from the average.

'bets_count_perc95': This type of gambler's average number of bets is also thin.

'amount_var_perc95': Concerning amount variance, this cluster has a sense of either reactivity or randomness. Players do not usually bet similar values.

'total_amount_year_perc95': The average total amount expended is reduced, although it is not at the lowest level.

On Cluster 3 sit **reactive gamblers**. This type of player is mainly described by a small number of bets placed and a considerable amount variance. That can be a player that spends little time playing but tries different approaches while playing. A reactive player must also be followed by a reduced total amount paid.

Cluster 4:

Cluster 3 contains 8,5% of all players studied. This group of players has curious characteristics compared to the rest once they generally hit high marks in the variables under analysis.

'wins_perc_perc95': The winning percentage of these players is somewhat optimistic and above the remaining gamblers.

'bets_count_perc95': Cluster 3 also holds close to the maximum bets count.

'amount_var_perc95': There is a signal of constant reactivity and awareness of previous results in this cluster. The average amount variance value in this cluster is seriously high.

'total_amount_year_perc95': Lastly, the sum of the amount placed in bets by these players is the highest.

To the 4th and final cluster belong the **professional gamblers**. These players are defined by a positive winning percentage, which is this group's most notable feature. Additionally, there is an increased number of bets and the total amount spent in a year.

4.5. SOM + K-MEANS

After trying out four different clustering methods, including already combinations of other techniques, the application of K-Means on top of a SOM solution was tested.

This subchapter will contemplate comments on the results after each step due to the nature and format the outputs hold.

Trial 1:

Scope: Year

Scaler: MinMax Scaler

Features: 'wins_perc_perc95, 'bets_count_perc95, 'amount_var_perc95', 'total_amount_year_perc95'

Grid dimension: 20 x 20

Figure 26 - Training Process of SOM Grid

The plot illustrated in Figure 29 describes the trail of the training process. The graphic shows that the mean distance from each data point to its corresponding closer unit decreased throughout iterations. In other words, this represents convergence, the model achieving a better solution as the process evolves.

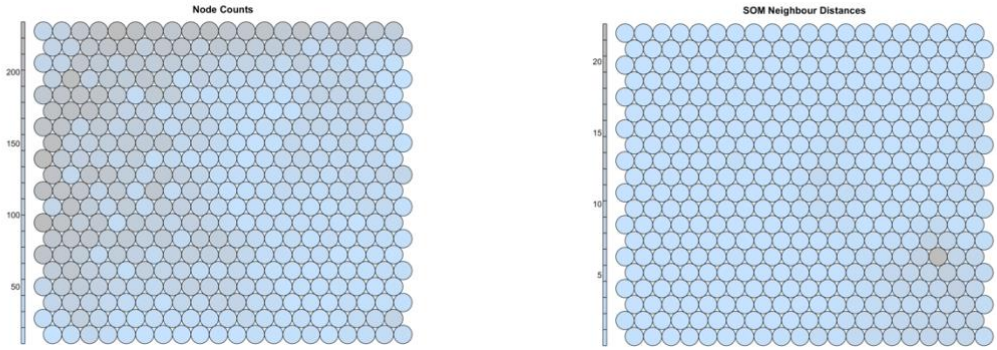


Figure 27 - Node Counts And Neighborhood Distances

After 2000 iterations, the aspect of the map is shown in Figure 30. The left-most
The graphic represents the Node Counts (refer to subchapter 2.3.4 for a detailed explanation of the illustration) and shows evidence of increased incidence on the top left side of the grid.
On the right, Neighborhood Distances are demonstrated. Despite residual locations where it is possible to find the considerable distance between neurons, it is fair to say that there is a homogeneous distribution of spaces between neurons.

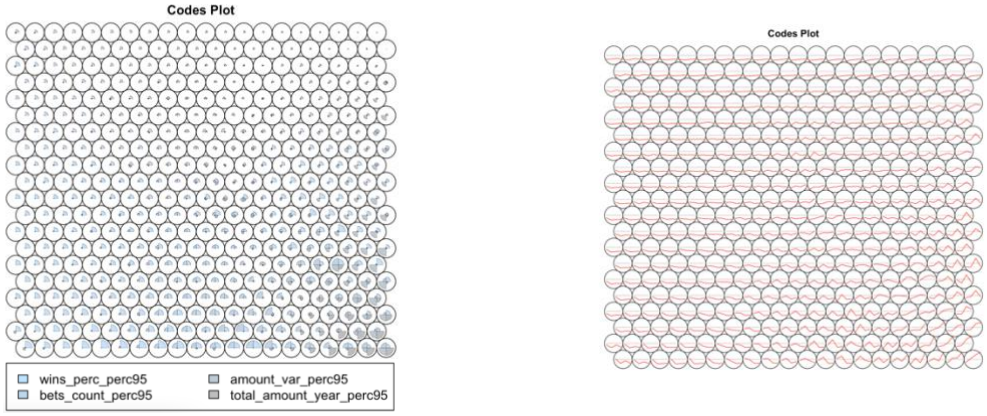


Figure 28 - Codes Plot

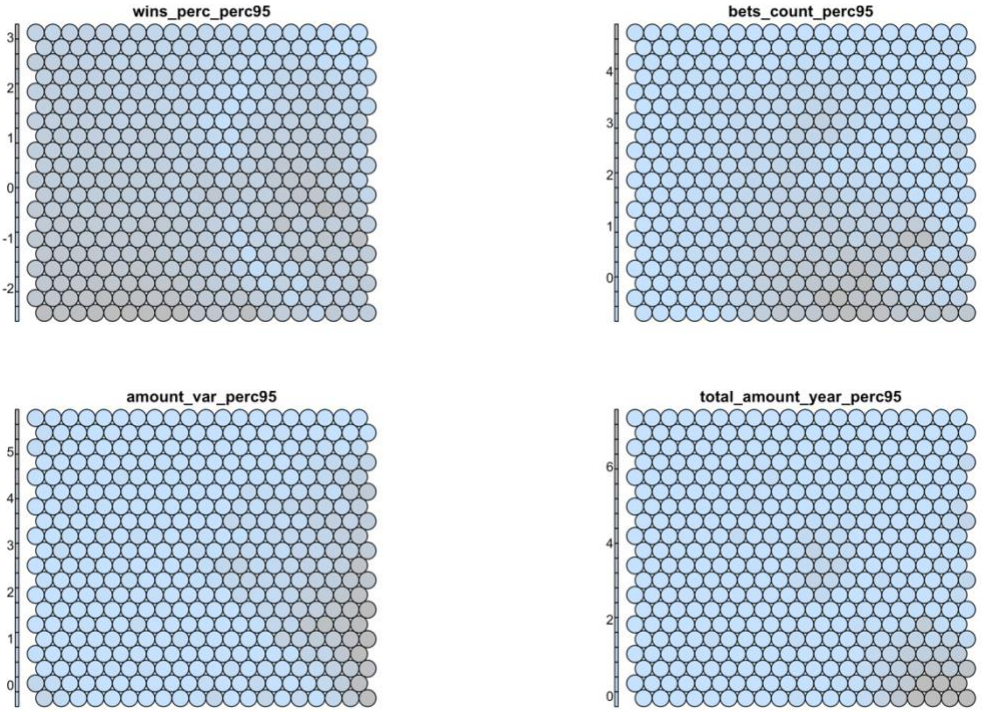


Figure 29 - Component Planes

The graphics above are indicators of each variable intensity in different grid locations. Figure 29, Codes Plots, indicate in the form of radars and lines how much each variable applies to each grid location. However, for concreteness and clarity, Figure 30, Component Planes, will be used to comment on this case. Through color intensity, it is possible to evaluate each variable influence separately. Such graphs inform the reader about the values that each feature holds in different locations of the input map.

Starting with 'wins_perc_perc95', which refers to a player's success rate, there is some dispersion. However, it is essential to highlight the increased incidence in the bottom-left corner. The darker color tone indicates those values are closer to 3. In other words, it is fair to say that there is an improved percentage of winning plays compared to different sides of the grid. Not only that location is impacted by a higher intensity of

'wins_perc_95'. There are also some units where this value arises on the middle-right side, although the frequency is not as considerable as in previous examples.

Moving on to 'bets_count_perc95', this is a straighter forward scenario. From the image, it is clear which zone is more affected by higher values of this feature. The bottom center of the map shows an increased intensity of the number of bets played. Values reaching 4 indicate that players allocated to the units in that location are, on average, more prone to place more bets (in number).

The following graphic shows which players tend to hold more variance between bets – 'amount_var_perc95'. Apart from a general homogeneity, a group of units is characterized by a higher variance between bet amounts. On the right side of this feature's plane, it is possible to see many grey scale colors. That means that players assigned to these units can hold up to 5 times more variance than the more consistent players in the amounts placed. This feature is a good indicator of player stability and consistency, with higher variance possibly meaning a lack of reasoning, for example.
Lastly, 'total_amount_year_perc95' is analyzed. As the variable name suggests, this variable is a sum of each player's amount throughout the year, as explained before. The bottom-right corner is suggestive of more significant amounts played. With amounts as outstanding as 6, this variable's intensity seems to be focused on a particular zone of the grid.

After applying SOM, the number of data points to work with dropped from around 33000 to 400, a nearly 99% decrease in number of observations. Even though each unit in the grid represents much more than one observation, this model has great relevance in terms of visualization and perception of point distribution.
Along with the visualization benefits described and demonstrated above, it is much easier to do other computing. In this case, the application of K-Means on the SOM result was performed.

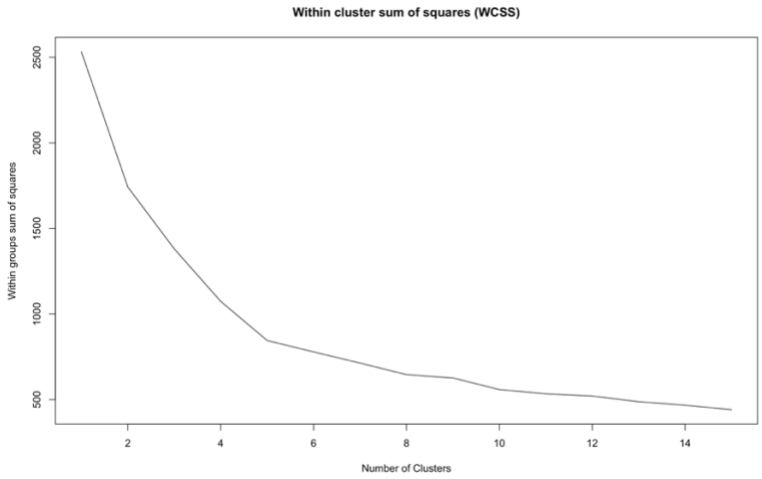


Figure 30 - Within Cluster Sum of Squared Errors

The first thing to do was to use the Within Cluster Sum of Squared Errors (WCSS) for a K-Means clustering solution on this SOM model, presented in Figure 31, to decide the number of clusters in advance. Using the elbow method (described in Figure 5 of chapter 2.3.1.1.), 5 was the chosen number for the number of groups. As seen in the line, there

is a less abrupt decrease in WCSS; a significant amount of differentiation is kept while having as few clusters as possible.

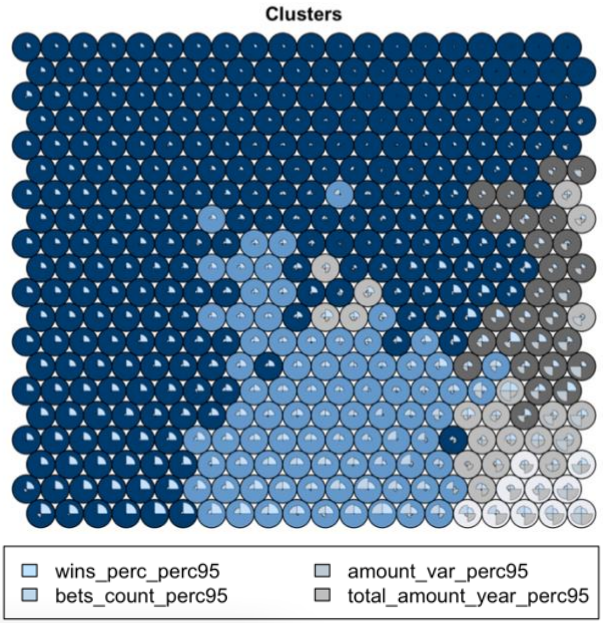


Figure 31 - Clustering Solution Shape

After having the number of clusters defined, K-Means was used to form clusters. Figure 32 represents each cluster in different colors, detailing the intensity of each feature in the clusters' units. To figure the size of a collection, the relation between the number of units in that cluster and node counts (Figure 30) must be observed.

For the case being, clusters will be named according to their colors: Dark Blue (DB), Light Blue (LB), Dark Grey (DG), Light Grey (LG), and White (W).

Dark Blue:

Considering the number of units assigned in the cluster solution (Figure 34) and the Node Counts (Figure 30), this cluster is the one aggregating more data points. From the clustering solution map, we can see that it incorporates nearly 75% of the units. This fact, together with the high incidence in the number of data points in the zones of the cluster (see Figure 30), is the explanation for this first conclusion.
Furthermore, in terms of the intensity of each variable, not only does Figure 34 holds information about it, but also the Component Planes (Figure 32) can provide that information.
From Figure 32, we take different conclusions:

'wins_perc_perc95': This variable seems to hold generally higher values in this cluster. However, due to its dimension, this cannot be considered transversal nor informative of the whole set.

'bets_count_perc95': The number of bets made by players in this cluster is much reduced compared to other units in the remaining groups. The area covered by dark blue units corresponds to the site with light blue on the 'bets_count_perc95'

component plane. Thus, in this case, we can say that players in the DB cluster usually are not regular players.

'amount_var_perc95': The gamblers in this cluster are also the most low-key in terms of bet amount variance. The respective component planes show that these players do not usually bet considerably different amounts. In other words, this cluster is characterized by low reactivity, opting for a more stable flow of volumes.

'total_amount_year_perc95': Following the trends of previous variables, DB clusters is the cluster detaining the least total amount sum for the year.

After considering each variable individually for this cluster, it is possible to classify this type of player as a **non-regular gambler**. These users commonly feature considerably low values of bets played, amount variance, and the total amount bet in a year.

Light Blue:

Following the same logic used in the DB cluster, this second cluster is the one holding the second most records. Therefore, not only by looking at its dimension in Figure 34, in which the LB cluster is detainer of 25,25% of the units but also by considering the Node Counts where it is possible to see the incidence of some grey nodes in the same zone of the LB cluster.

'wins_perc_perc95': In the same way this variable did not hold much information for the DB cluster, it is also quite inconclusive for the LB cluster. Even though a fraction of this cluster is characterized by considerable winning percentages, it does not apply to the entire set uniformly.

'bets_count_perc95': Contrarywise, when it comes to the number of bets made, players belonging to LB clusters are some of the most frequent gamblers. Comparing them to most collections, these players generally place a higher number of bets.

'amount_var_perc95': Looking at the amount variance in this group, they can be considered very steady in those matters. That indicates, as in the case of the DB cluster, that these are probably non-reactive gamblers. The amounts used are relatively constant throughout every bet.

'total_amount_year_perc95': The values are somewhat low for the whole amount spent by this group. In general, it is appropriate to call these gamblers small spenders.

With every feature considered, these players will be classified as **leisure gamblers**. Such players are described by a considerably high number of bets placed. However, typically low amount per bet and also very low variance.

Dark Grey:

This cluster is the third most represented in the clustering solution, with 6,75% of the units on the map having dark grey color. That gains more relevance since it has some features that differ from other clusters.

'wins_perc_perc95': Considering the relation between the final solution map and this specific component plane, it is possible to see that the zone with dark grey units corresponds to a location of grey tones in the component plane. Such a thing indicates a generally positive tendency in winning percentages. However, although there is a positive trend, the borderline between the positive and negative in this case is very shallow.

'bets_count_perc95': The number of bets placed by these players usually is significantly reduced compared with other clusters.

'amount_var_perc95': Being the most standing out characteristic of this cluster, the amount variance between bets is substantially elevated, implying that players tend to bet with some sense of reaction.

'total_amount_year_perc95': This cluster also follows the tendencies of previous sets. The sum of the amounts used over the year is typically low, with little room for exceptions.

In this cluster, more reactive players can be found. They possibly bet with previous results in mind or simply hit random amounts. Even though that can be considered a negative behavior, once the number of bets made and the overall amount spent are regularly short, there seems to be no genuine concern with these gamblers. All considered, these players will be classified as **reactive gamblers**.

Light Grey:

Moving on to the fourth cluster, this one represents 5% of the units on the clustering solution map (Figure 34). Again, this cluster demonstrates new and unseen characteristics compared to previously examined ones.

'wins_perc_perc95': From this component plane, it can be seen that there is a negative winning percentage across the majority of its nodes. Light blue tones precisely indicate that. In this case, the intensities shown within each node in Figure 34 can help to reach that conclusion. Furthermore, it is possible to see from those radars that there are generally low values for 'wins_perc_perc95'.

'bets_count_perc95': As in the case of the last variable, the radars of Figure 34 get relevant again. It is possible to see from those mini charts that there is undoubtedly an above-the-average trend for this cluster's players to bet in higher volume. Although it is not valid in every case, it can be considered an outline of this group.

'amount_var_perc95': Another critical characteristic of gamblers in this cluster is the variance in their bets. The component planes help to conclude this. For example, looking at the near-bottom-right side of the grid, the presence of light

grey to dark grey is visible. That indicates players in this cluster tend to hold much higher variance than players in cluster DB or LB, for example.

'total_amount_year_perc95': In terms of the total amount spent per year, players in the light grey cluster are also top seaters. Players in this cluster belong to a niche in which considerable money is expended, another trait of this group.

There is in this cluster room for some interpretation. These players possess distinctive specifications compared to the majority of the cluster. In the first place, the winning percentage is vastly unfavorable. That success rate not being positive is only aggravated by the fact that there is a high volume of bets when talking about this group – both in the count of bets and money spent in total. Finally, as said above, high bet value variance also appears to be a feature of these participants, which can probably mean reactivity and even a lack of emotional control. Wrapping, experts could follow these results to detect any possible situation of dangerous/addictive gambling. Thus, players in this cluster will be referred to as **problematic gamblers**.

White:

Reaching the final cluster, white, is the case of a clear minority. In the sense of the relation pondered previously, if the cluster final solution grid (Figure 34) and Node Counts (Figure 30) are overlapped, it is possible to imagine a low density of data points assigned to white nodes. White nodes comprise 2,5% of the total 400 present on the clustering solution.

'wins_perc_perc95': First thing to note in this cluster is that, unlike in the majority of other groups, there is a slightly positive trend in this case. Even though the tendency is short, it is notable from the component planes.

'bets_count_perc95': The number of bets placed by this type of player is also considerably higher than the average player in our dataset.

'amount_var_perc95': In terms of the amount variance in the different bets, the infographics are somewhat inconclusive. Almost perfectly half-split, there are nodes with higher variance and others where that value drops.

'total_amount_year_perc95': Regarding amounts invested in gambling, these players lead the whole sample. Each observation in this group is connected to a significant amount of money spent on gambling.

Players in this cluster reveal some exciting characteristics. The one that stands out the most would be the ratio of the amount spent and the success rate. That, combined with the number of bets placed, reveals possible experience in the field. White cluster players can be assumed to be **professional gamblers**.

4.6. SUMMARY

After trying various techniques to address the problem, the author progressively reached answers to the proposed questions. This subchapter aims to summarize the course to reach meaningful clusters.

Initially, K-Means was applied to find valuable sets of players. With this algorithm, four distinct combinations of features were tested. Unfortunately, such an experiment revealed useless clusters because of the lack of characterization or the few distinctions between groups.

Following K-Means, the author implemented Mean-Shift. As in K-Means' case, different bags of attributes were combined to find the best possible approach. For this algorithm, it is necessary to define a parameter: bandwidth. Such a value will affect the number of clusters to use. However, the number of resulting clusters was often not satisfying, even with different bandwidth estimations. Therefore, this technique was disregarded.

Still in the scope of density-based clustering algorithms, DBSCAN was tested. As in Mean-Shift, there are parameters to be defined *a priori*. Even though the model achieved interesting numbers of clusters, their sizes were significantly misadjusted. After concluding that, the author dropped this method.

Since clustering algorithms individually were not handling the task effectively, the author decided to test combinations of different methods. The first was a combination of K-Means and hierarchical clustering. Considering different trials, in which the main difference were the variables in use, it was possible to retrieve relevant clusters. Then, across all players, the author divided them into five categories: problematic, reactive, non-regular, leisure, and professional gamblers. Finally, one of the solutions was kept, and its groups were detailed in subchapter 4.4.

Finally, a mixture of SOM, K-Means, and hierarchical clustering was performed. The sequence uses K-Means to find the grid size, then uses SOM with that map size and applies hierarchical clustering on that solution. This experiment also presented a suitable solution. In the end, it was possible to identify five clusters. The names given to these new collections are the same in K-Means + HC solution. Moreover, such a combination of algorithms provides valuable multi-dimensional data visualization tools.

5. MODEL SELECTION AND CONCLUSIONS

Chapter 4 and its corresponding appendix illustrate an extended set of algorithms tested and the respective results. Different techniques were tested, and that segmentation aims to address predefined research questions and find suitable answers. This section's purpose is to define the best models, select a favorite one to tackle the problem and use its results to reach meaningful conclusions and answer those research questions.

It is clear that there were two models considered best performers. That is also why only two of the trials in the chapter above had clusters being described: **K-Means+HC Trial 6.2** and **K-Means+SOM+HC**. Both solutions held explanatory clusters and were put through the same set of variables: winning percentage, number of bets, bet amount variance, and total amount spent in the year. As for the results of these trials, the groups of players found were also similar and thus named accordingly. Moreover, the names of the clusters are self-explanatory: **non-regular gamblers**, **leisure gamblers**, **reactive gamblers**, **professional gamblers,** and finally **problematic gamblers**.

Reaching this point of the study, relating the results obtained with the scope of the project, which is concerned with the concept of responsible gambling, becomes critical. From the clusters mentioned above it is fair to disregard most of the categories due to lack of concern in terms of potential psychological issues in gambling. However, one of the groups must be considered and treated cautiously: problematic gamblers. As the name indicated, these are players that typically demonstrate bad gambling habits, both in amount and frequency. As said above in the explanation of the responsible gambling, it is concerned exactly with those two variables of gambling. There is a chance that players siting in that cluster are developing pathological issues in gambling and should be monitored carefully.

Considering that the categories generated by the models and the features used were the same for both, the algorithm selection will be based on the graphics and their exactness. Even though the second combination uses interesting and different visualization tools, the graphics of the first combination are more precise and potentially more reliable. Combining quantitative and qualitative appropriateness of the solution, the author decided to use **K-Means + HC Trial 6.2** approach to reach answers to the research questions.

1- Is it possible to perform segmentation on online gamblers based on their playing behavior and tendencies?

This work project shows that, by resorting to appropriate data mining tools, it is possible to segment online roulette gamblers based exclusively on behavioral features.

2- How is gambling attendance distributed among the different types of players? (Professional, leisure, others)

Although there are possibly dozens of player types, this thesis encountered five different categories that can explain the distribution of gambling attendees: non-regular, leisure, reactive, professional, and problematic gamblers.

3- Is it possible to identify potential or actual gambling disorders being developed by a gambler based on his playing behavior and tendencies?

It is possible to identify patterns in players that generally hold negative attributes simultaneously. These attributes can go from the amount spent to success rates. However, assigning a disorder condition can only be done by experts in this field.

4- Is playing frequency directly or indirectly linked with gambling problems?

In this thesis, playing frequency was addressed only in the form of bet frequency and amount spent. Considering that, the results show that problematic gamblers spend considerable money and have a relatively increased number of bets. Although, it is not right to link that directly to gambling issues since professional players, for example, also score high on these two variables.

6. RECOMMENDATIONS FOR FUTURE WORK

Upon completion of this project, the author found room for improvements in the work developed and in features that can be added.

Beginning by stating improvements on this thesis's results, there are a few points that could be added or changed:

1. Working with approximately 33000 observations limits the model's ability. Having more data points could improve results in quality and reliability.
2. The depth of data (quantity and variability) could also be broader. Personal attributes may also add value to the solution. For example, age, region, education, and other variables are relevant when performing a behavioral analysis.
3. Working with improved computational power would allow the researcher to test more model configurations simultaneously and, thus, have more options to choose.

Focusing on future work, on top of what was developed, there is also room to discuss. The most attractive feature to add to this project is a classification model. Such a model would be a supervised machine learning algorithm, which would learn based on the result of an unsupervised learning algorithm (like the ones created here). With that add-on working effectively, gambling agencies could predict if a given player is starting to develop harmful gambling behaviors and react accordingly.

7. BIBLIOGRAPHY

Algren, M., Fisher, W., Landis, A. (2021). *Data Science Applied to Sustainability Analysis – Machine learning in life cycle assessment* https://doi.org/10.1016/B978-0-12-817976-5.00009-7.

Belyadi, H., Haghighat, A. (2021). *Machine Learning Guide for Oil and Gas Using Python - Unsupervised machine learning: clustering algorithms.* https://doi.org/10.1016/B978-0-12-821929-4.00002-0.

Bocca, F., Rodrigues, L. (2016). *The effect of tuning, feature engineering, and feature selection in data mining applied to rainfed sugarcane yield modelling.*

Brodeur M, Audette-Chapdelaine S, Savard AC, Kairouz S. (2021). *Gambling and the COVID-19 pandemic: A scoping review. Prog Neuropsychopharmacol Biol Psychiatry*. https://doi: 10.1016/j.pnpbp.2021.110389.

Clark, L (2014). *Disordered gambling: the evolving concept of behavioral addiction. Annals of the New York Academy of Sciences*, *1327*(1), 46-61.

EGBA (2021). *European Gaming & Betting Association – European Online Gambling Key Figures 2021 Edition*

El Bouchefry, K. & S. de Souza, R. (2020). *Knowledge Discovery in Big Data from Astronomy and Earth Observation – Learning in Big Data: Introduction to Machine Learning.* https://doi.org/10.1016/B978-0-12-819154-5.00023-0

Fernandez, Á. & Bella, J & Donrronsoro, J (2022). *Supervised outlier detection for classification and regression. Neurocomputing, 486, 77-92.* https://doi.org/10.1016/j.neucom.2022.02.047

Gajjar, K. (2020). *Cluster Analysis with DBSCAN: Density-based spatial clustering of applications with noise* https://medium.com/analytics-vidhya/cluster-analysis-with-dbscan-density-based-spatial- clustering-of-applications-with-noise-6ade1ec23555

Gonsalves, T. & Upadhyay, J. (2021). *Integrated deep learning for self-driving robotic cars - Artificial Intelligence for Future Generation Robotics.* https://doi.org/10.1016/B978-0-323-85498-6.00010-1

Hu, L., Liu, H., Zhang, J., & Liu, A. (2021). *KR-DBSCAN: A density-based clustering algorithm based on reverse nearest neighbor and influence space. Expert Systems with Applications, 186,115763.* https://doi.org/10.1016/j.eswa.2021.115763

Liu, N., Xu, Z., Zeng, X., & Ren, P. (2021). *An agglomerative hierarchical clustering algorithm for linear ordinal rankings. Information Sciences, 557,170-193.* https://doi.org/10.1016/j.ins.2020.12.056

Malato, G. (2021). *Outlier identification using Interquartile Range*

Niño-Adan, I., Landa-Torres, I., Portillo, E., & Manjarres, D. (2022). *Influence of statistical feature normalisation methods on K-Nearest Neighbours and K-Means in the context of industry 4.0. Engineering Applications of Artificial Intelligence, 111, 104807.* https://doi.org/10.1016/j.engappai.2022.104807

Tang, W., & Lu, Z. (2022). *Application of self-organizing map (SOM)-based approach to explore the relationship between land use and water quality in Deqing County, Taihu Lake Basin. Land Use Policy, 119, 106205.* https://doi.org/10.1016/j.landusepol.2022.106205

Tirelli, T., & Pessani, D. (2011). *Importance of feature selection in decision-tree and artificial-neural-network ecological applications. Alburnus alburnus alborella: A practical example. Ecological Informatics, 6(5), 309-315.* https://doi.org/10.1016/j.ecoinf.2010.11.001

Wu, K., Yang, M. (2007). *Pattern recognition – Mean shift-based clustering* https://doi.org/10.1016/j.patcog.2007.02.006.

Zhu, Q., Tang, X., & Elahi, A. (2021). *Application of the novel harmony search optimization algorithm for DBSCAN clustering. Expert Systems with Applications, 178, 115054.* https://doi.org/10.1016/j.eswa.2021.115054

8. APPENDIX

K-MEANS

Trial 1:

Scaler: MinMax Scaler (-1, 1)

Features: ‘amount_var’, ‘hours_played’, balance_perc’, ‘amount_per_hour’, ‘bets_count_per_hour’

Results:

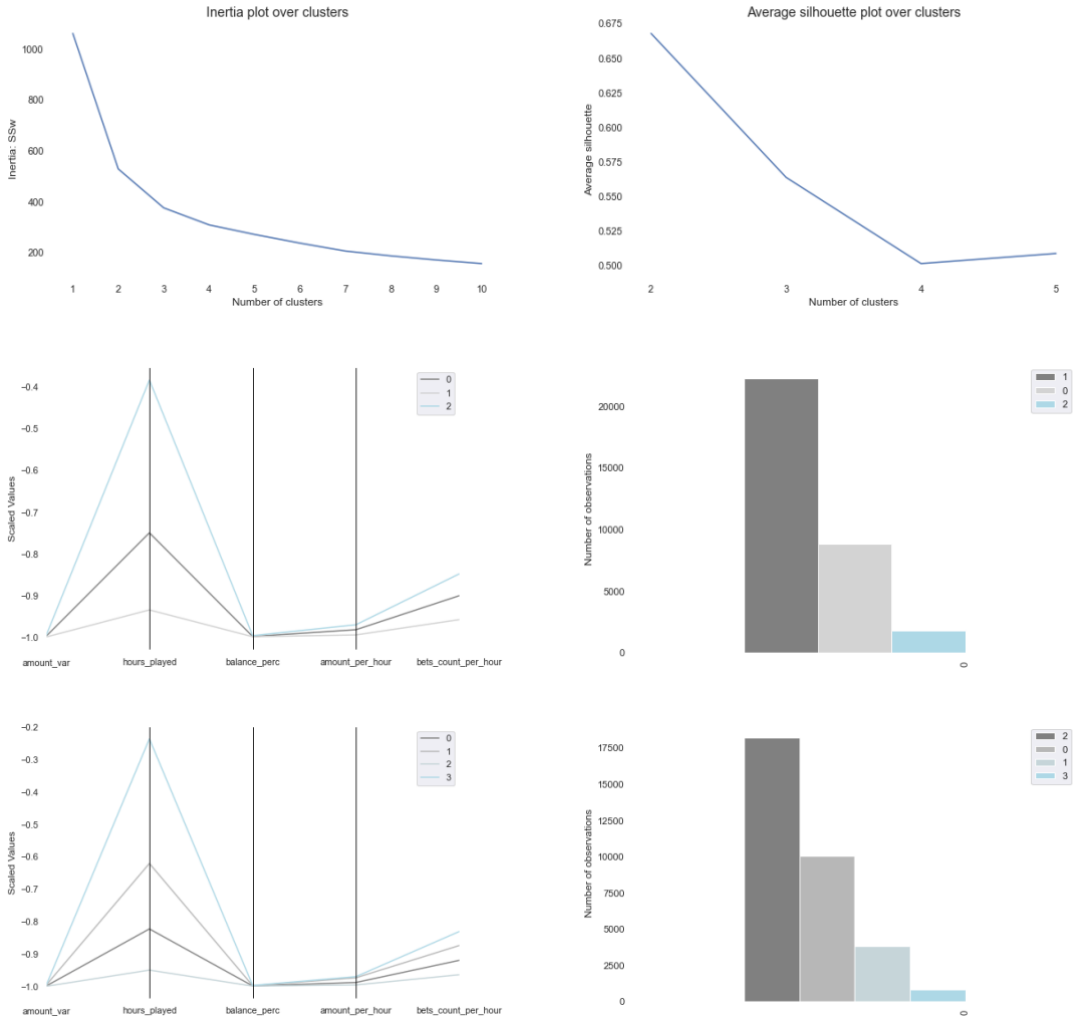


Figure 32 - Cluster Analysis for K-Means Trial 1 with k=3 and k=4

Comments:

Inertia and average silhouette score plots suggest the optimal number of clusters should be between 3 and 4.
After using K-Means, it did not generate relevant collections with any number of clusters. Additionally, the results indicate possible redundancy associated with the features included in this trial.

Trial 2:

Scaler: MinMax Scaler (-1, 1)

Features: 'amount_var', 'hours_played', balance', 'amount_per_hour', 'amount_mean', 'bets_count'

Results:

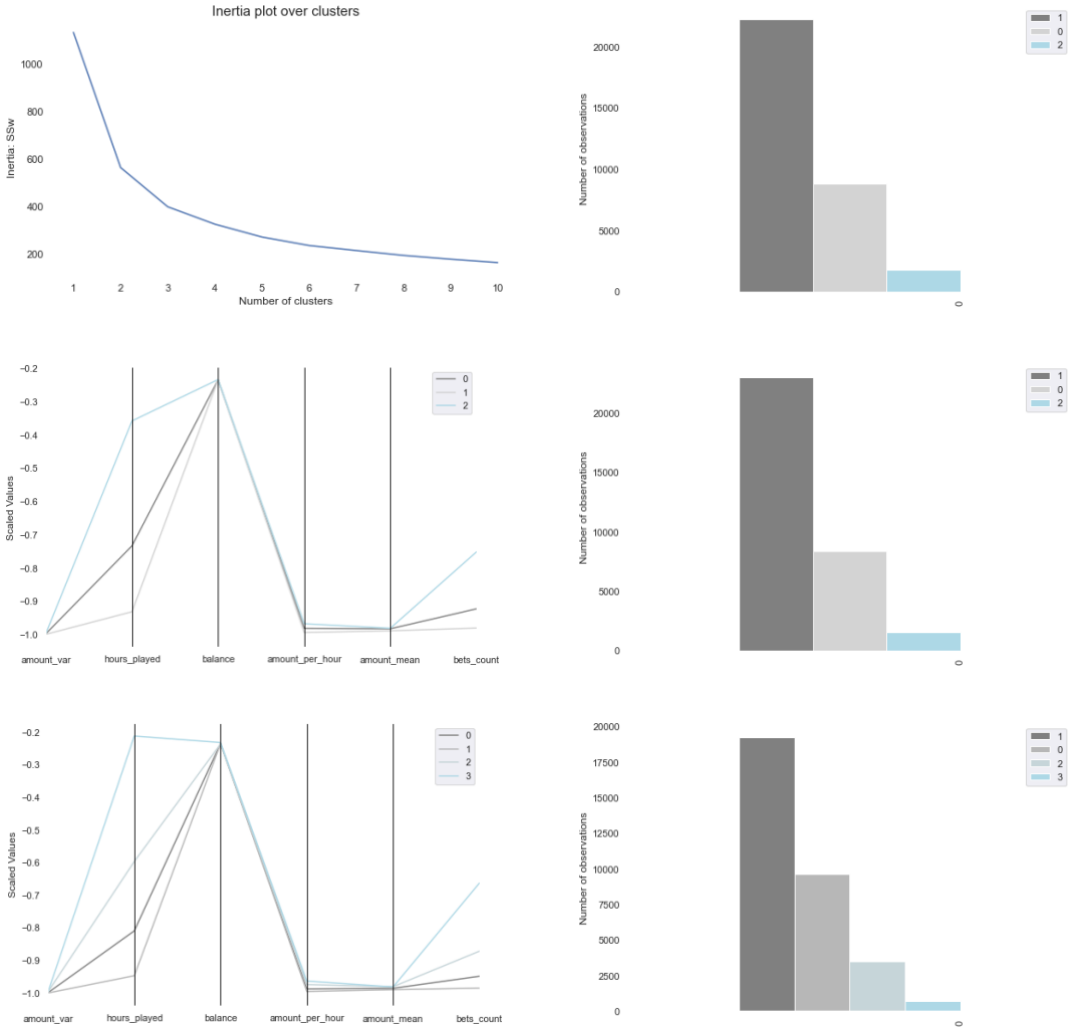


Figure 33 - Cluster Analysis for K-Means Trial 2 with k=3 and k=4

Comments:

As in the experiment above, inertia and average silhouette score plots also indicate that the appropriate choice for number of clusters is between 3 and 4.
When the model was applied, there were still no insightful clusters to use with any option. Similarly, the results indicate possible redundancy associated with the features included in this trial.

Trial 3:

Scaler: MinMax Scaler (-1, 1)

Features: 'amount', 'hours_played', 'bets_count'

Results:

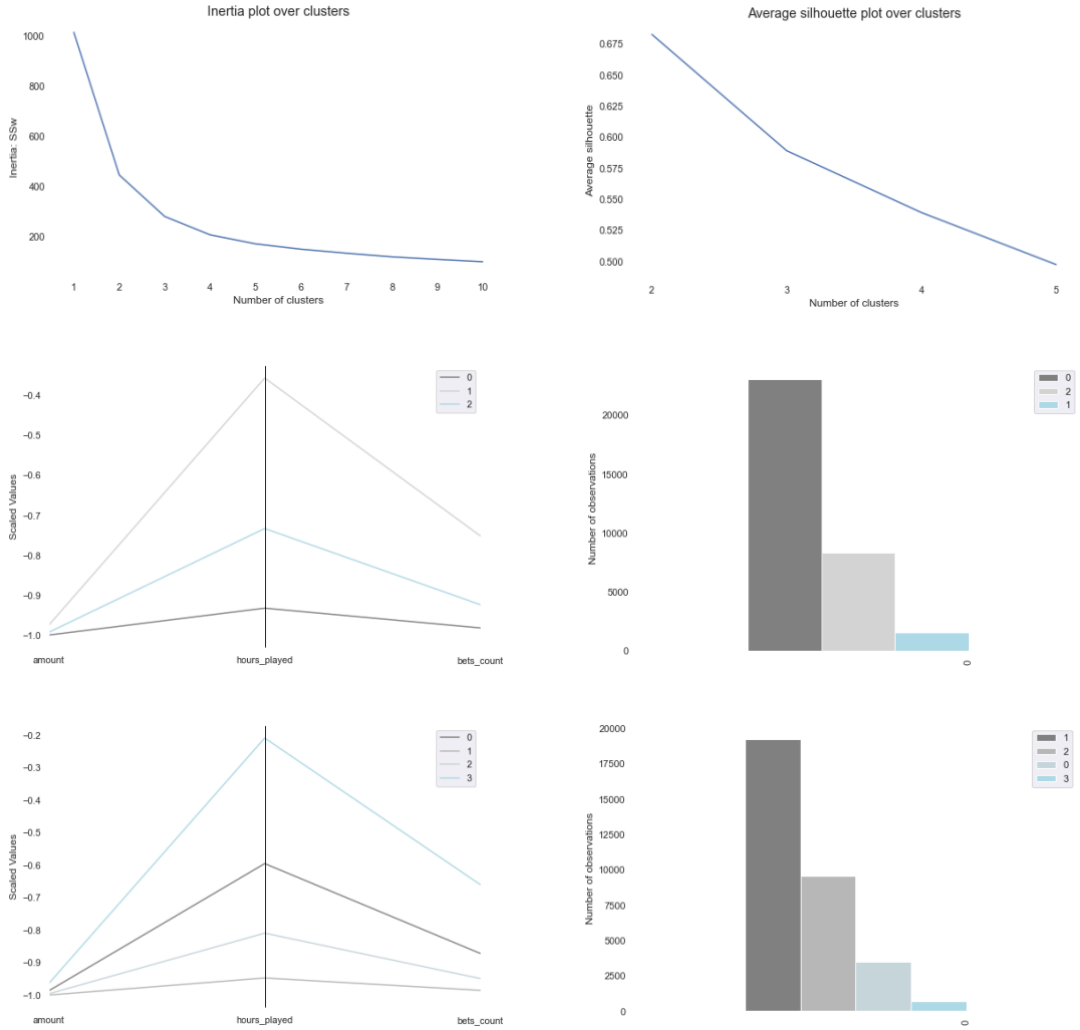


Figure 34 - Cluster Analysis for K-Means Trial 3 with k=3 and k=4

Comments:

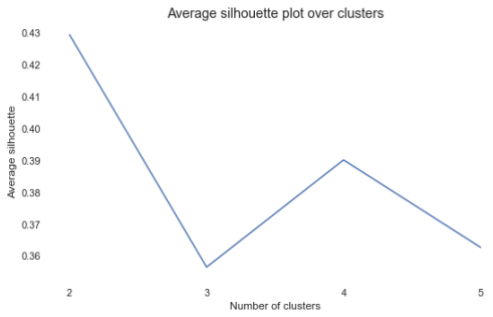
Presenting no difference from the counterpart trials, the most suitable number of clusters for this data selection is either 3 or 4 clusters.
In this attempt, the removal of some possible irrelevant features was done. The feature choice considered the result of the previous one.
Even though the results are not yet satisfying, there is a more apparent distinction between the elements of each cluster

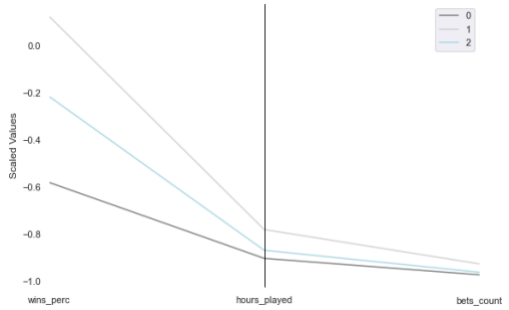
Trial 4:

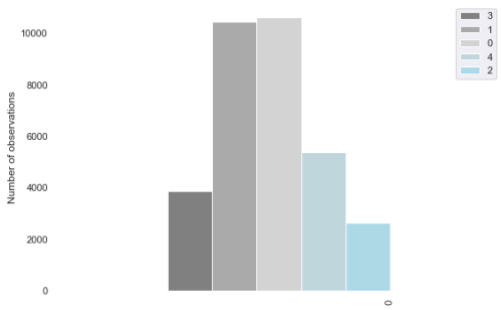
Scaler: MinMax Scaler (-1, 1)

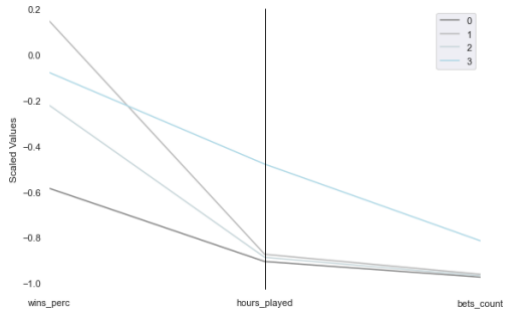
Features: 'wins_perc, 'hours_played', 'bets_count'

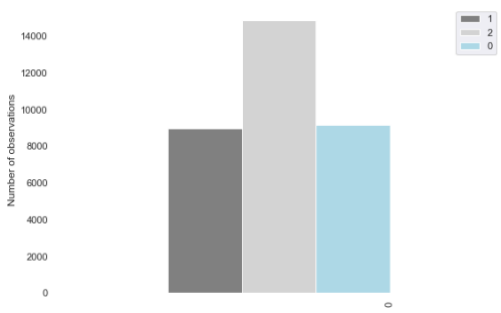
Results:











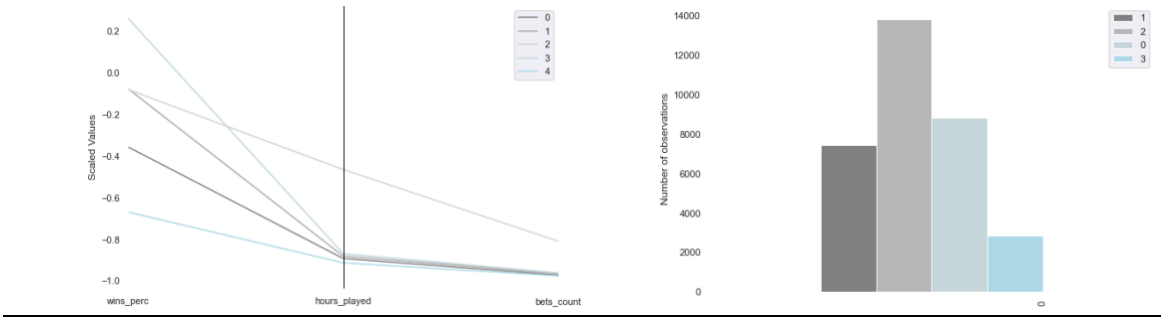


Figure 35 - Cluster Analysis for K-Means Trial 4 with k=3 and k=4

Comments:

In the fourth trial, and mainly due to the average silhouette score graphic, k=5 was also considered a possible beneficial solution.
The result of the current test is not better than any previous one. The results were not already sufficient nor provided any exciting segmentation.

MEAN-SHIFT

Trial 1:

Scope: Year

Scaler: MinMax Scaler (-1, 1)

Features: 'amount_var', 'hours_played', balance_perc', 'amount_per_hour', 'bets_count_per_hour'

Results:

Bandwidth	Number of Clusters
0.1129	99
0.1129 + 0.1	33
0.1129 + 0.2	18

Table 6 - Bandwidth effect on the number of clusters Mean-Shift Trial 1

Comments:

This first trial contemplated all data available in the project, which refers to the year's scope. Very likely, for reasons related to data shape, there was no suitable solution found. Too high numbers of expected clusters were obtained, indicating that the scope and feature choice may not be adequate.

Trial 2:

Scope: Semester

Scaler: MinMax Scaler (-1, 1)

Features: ‘amount_var’, ‘hours_played’, bets_count, ‘wins_perc’

Results:

Bandwidth	Number of Clusters
0.2965	20
0.2965 + 0.1	9
0.2965 + 0.2	5

Table 7 - Bandwidth effect on the number of clusters Mean-Shift Trial 2

Comments:

After applying the algorithm with **bandwidth = 0.296 5+ 0.2,** 5 Clusters were obtained, which could be a reasonable number of clusters for the problem at hand. However, the distribution of points was considered unequal, having the algorithm assigned 99% of the data points to a single cluster. Based on that fact, the author disregarded this solution.

Trial 3:

Scope: Trimester

Scaler: MinMax Scaler (-1, 1)

Features: ‘amount_var’, ‘hours_played’, bets_count, ‘wins_perc’

Results:

Bandwidth	Number of Clusters
0.3247	17
0.3247+ 0.1	9
0.3247 + 0.2	5

Table 8 - Bandwidth effect on the number of clusters Mean-Shift Trial 3

Comments:

The algorithm was applied using **bandwidth = 0.3247 + 0.2.** The results obtained are discarded for the same reasons in Trial 2.

DBSCAN

Trial 1:

Scope: Year

Scaler: MinMax Scaler (-1, 1)

Features: 'amount_var', 'hours_played', balance_perc', 'amount_per_hour', 'bets_count_per_hour'

Results:

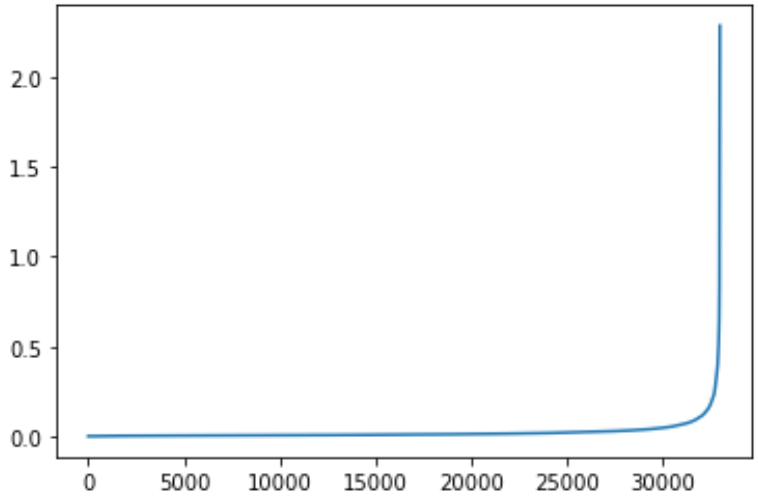


Figure 36 - K-Distance Graph for Year Scope on DBSCAN

eps	Number of Clusters
0.1	2
0.2	2
0.3	2
0.4	2
0.5	2

Table 9 - eps effect on the number of clusters DBSCAN Trial 1

Comments:

In the first approach, using all data available, the result was undesired. After testing it with the acceptable eps values according to the K-Distance graphic, the number of estimated clusters remained at 2. Since that is not considered a fair number of groups to answer the research question, this solution was dropped.

Trial 2:

Scope: Semester

Scaler: MinMax Scaler (-1, 1)

Features: ‘amount_var’, ‘hours_played’, bets_count, ‘wins_perc’

Results:

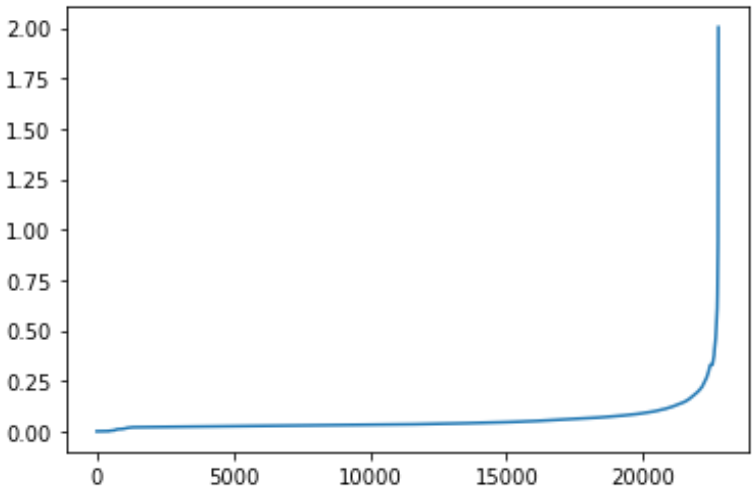


Figure 37 - K-Distance Graph for Semester Scope on DBSCAN

eps	Number of Clusters
0.1	3
0.15	2
0.2	2

Table 10 - eps effect on the number of clusters DBSCAN Trial 2

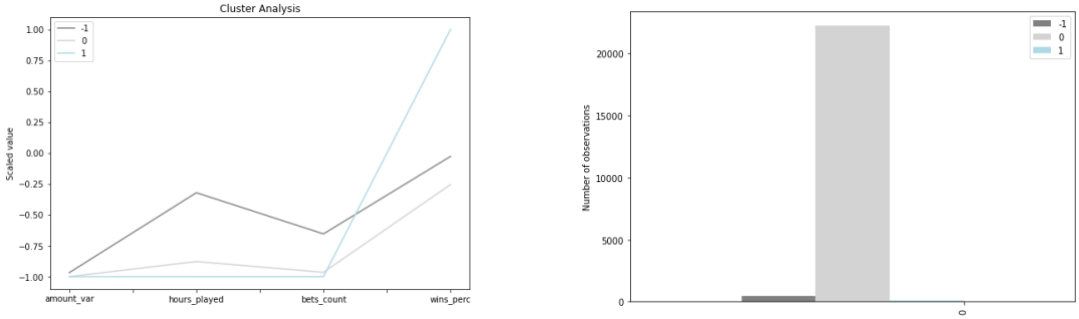


Figure 38 - Cluster Analysis for DBSCAN Trial 2 with Semester Scope

Comments:

From the charts above, one could firstly identify three attractive clusters by looking at the parallel coordinates graphic. However, any assumption made would be completely thrown by the information on the bar chart. Therefore, the conclusion of this experiment tells that there is no significance in the result, considering that one cluster has assigned most of the data points to it.

Trial 3:

Scope: Trimester

Scaler: MinMax Scaler (-1, 1)

Features: ‘amount_var’, ‘hours_played’, bets_count, ‘wins_perc’

Results:

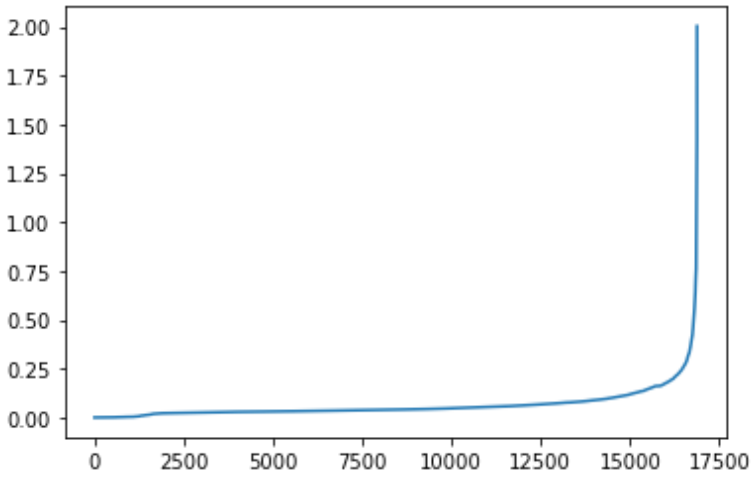


Figure 39 - K-Distance Graph for Trimester Scope on DBSCAN

eps	Number of Clusters
0.1	3
0.15	2
0.2	2

Table 11 - eps effect on the number of clusters DBSCAN Trial 3

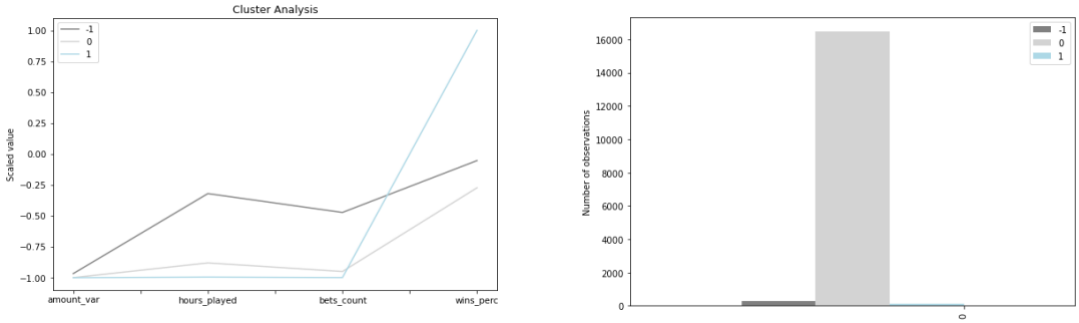


Figure 40 - Cluster Analysis for DBSCAN Trial 3 with Trimester Scope

Comments:

The result of the third trial is a picture of what happened in trial 2. Moreover, the dispersion and shape of clusters obtained in each attempt are very similar, which indicates that the scope did not play a relevant part in the results.

K-MEANS + HC

Trial 1:

Scaler: MinMax Scaler (-1, 1)

Features: ‘amount_var, ‘hours_played’, ‘balance_perc’, ‘amount_per_hour’, ‘bets_count_per_hour’

Results:

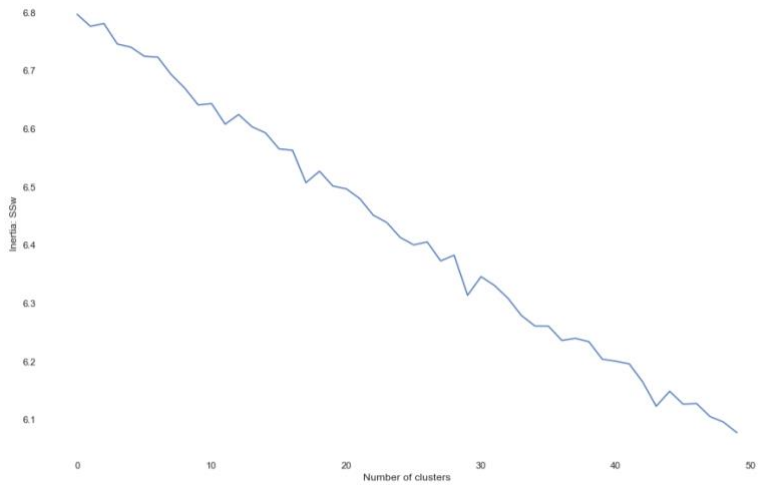


Figure 41 - WCSS for K-Means + HC Trial 1

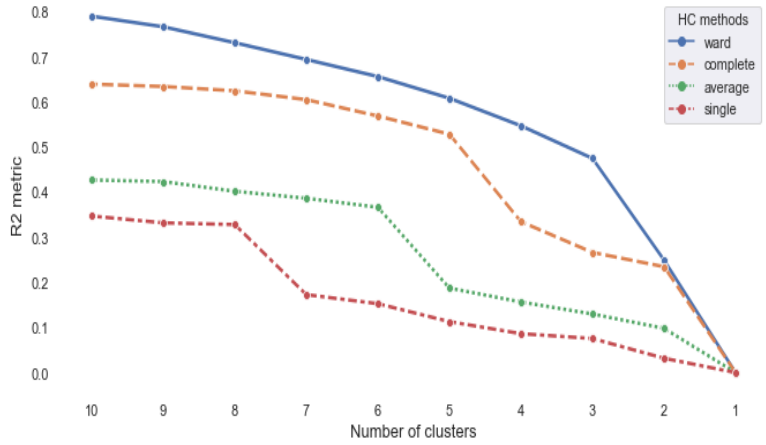


Figure 42 - R Squared for K-Means + HC Trial 1

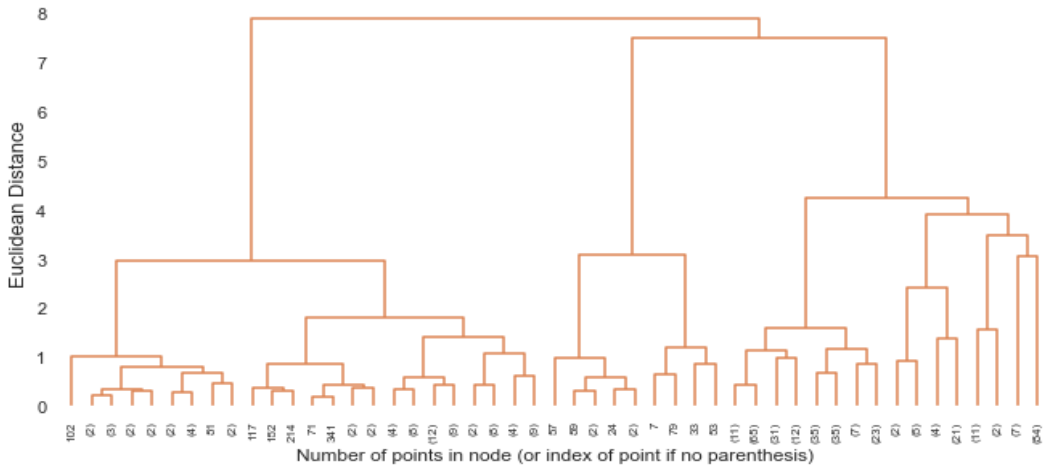


Figure 43 - Dendrogram for K-Means + HC Trial 1

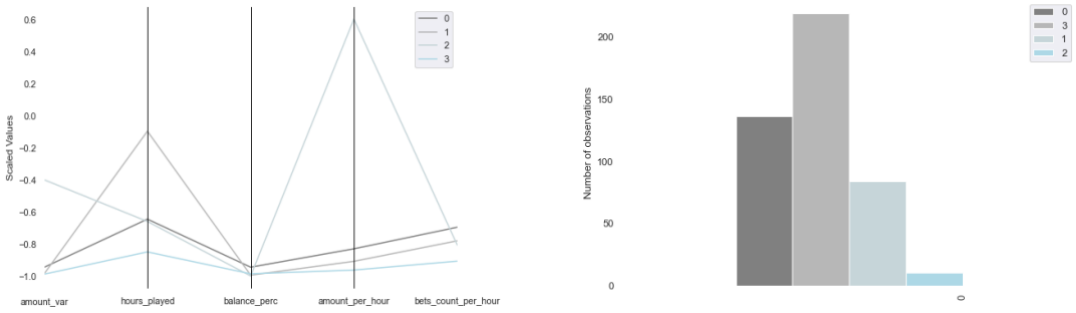


Figure 44 - Cluster Analysis for K-Means + HC Trial 1 k = 4

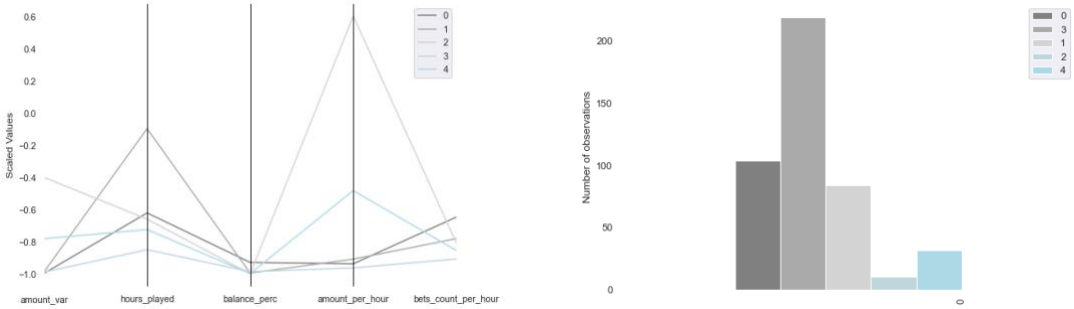


Figure 45 - Cluster Analysis for K-Means + HC Trial 1 k = 5

Comments:

The kick-start trial for this method did not reveal a suitable cluster solution with 4 and 5 clusters. Even though some variables felt interesting for further use and some dispersion and balance between sets were found, the result is meaningless.

Trial 2:

Scaler: MinMax Scaler (-1, 1)

Features: 'amount', 'hours_played', 'bets_count'

Results:

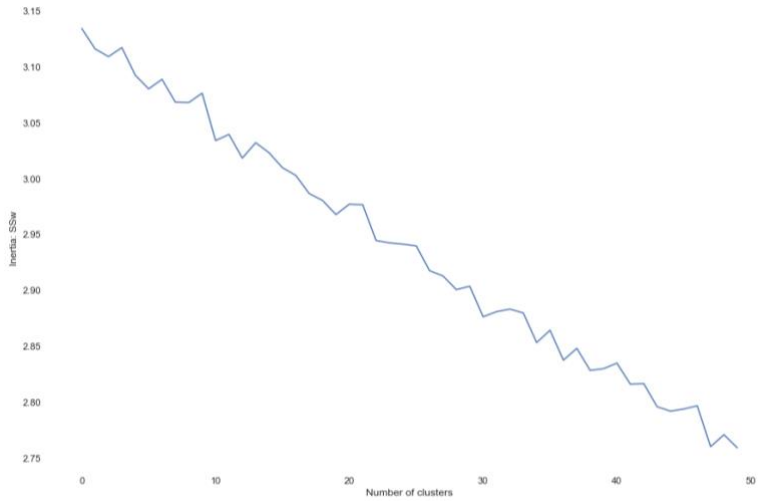


Figure 46 - WCSS for K-Means + HC Trial 2

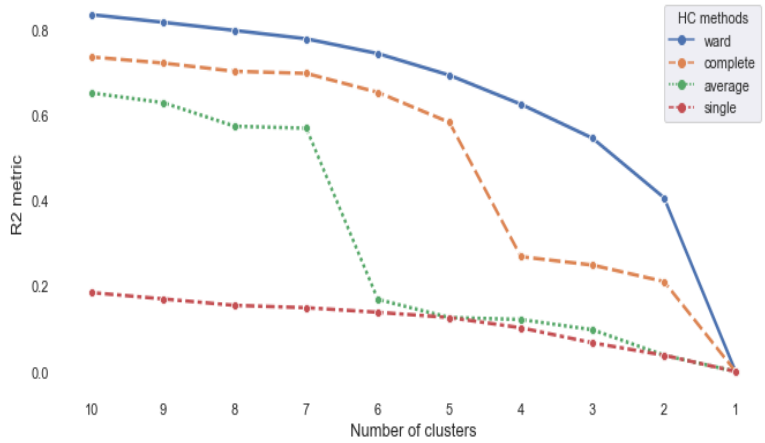


Figure 47 - R Squared for K-Means + HC Trial 2

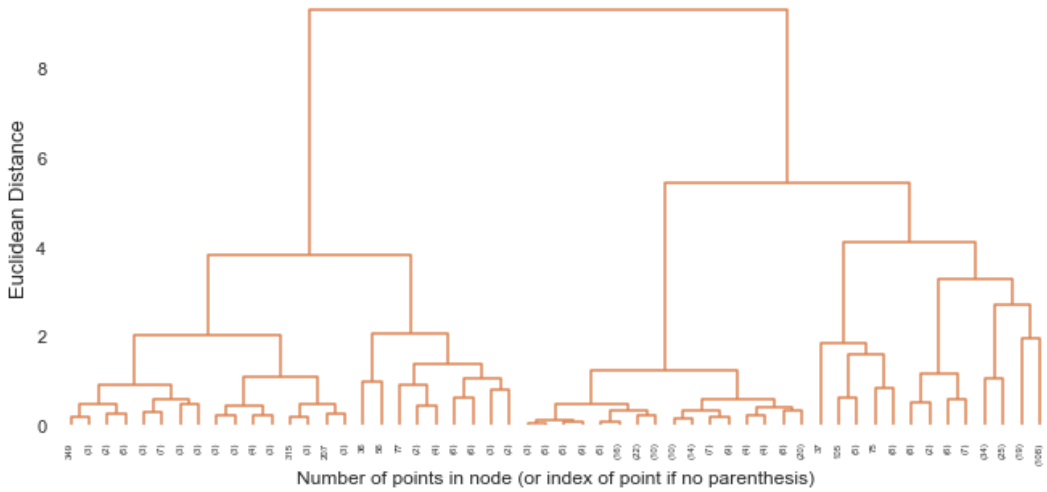


Figure 48 - Dendrogram for K-Means + HC Trial 2

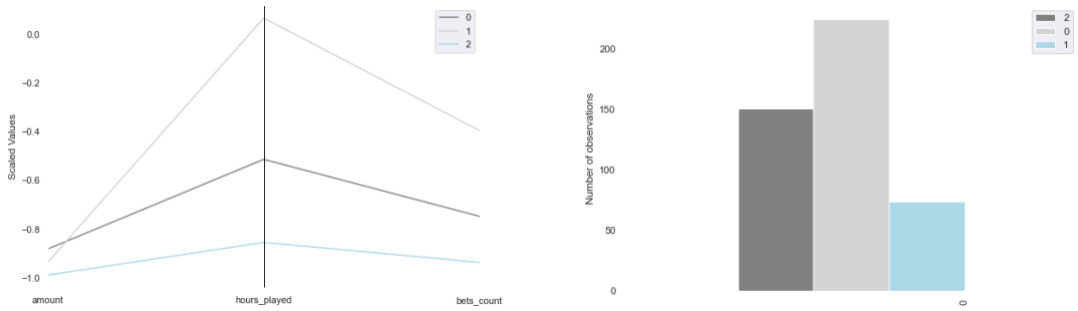


Figure 49 - Cluster Analysis for K-Means + HC Trial 2 k = 3

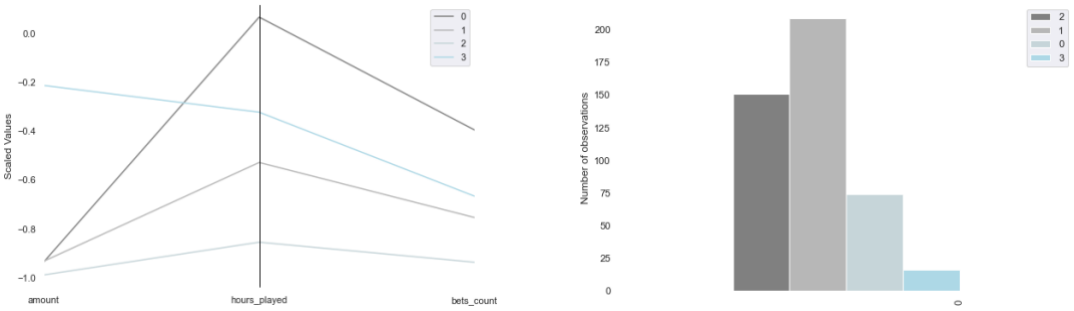


Figure 50 - Cluster Analysis for K-Means + HC Trial 2 k = 4

Comments:

According to the result of Trial 1, the number of variables dropped according to that result. What can be seen in the last two pictures are decent solutions. Some differentiation is achieved between clusters, and the model carried good distribution of data points per cluster. From the two solutions, with k=3 and k=4, the most relevant one is the second. It gets more meaningful with an additional collection that represents the highest amount of spending players.

Trial 3:

Scaler: MinMax Scaler (-1, 1)

Features: ‘wins_perc, ‘hours_played’, ‘bets_count’

Results:

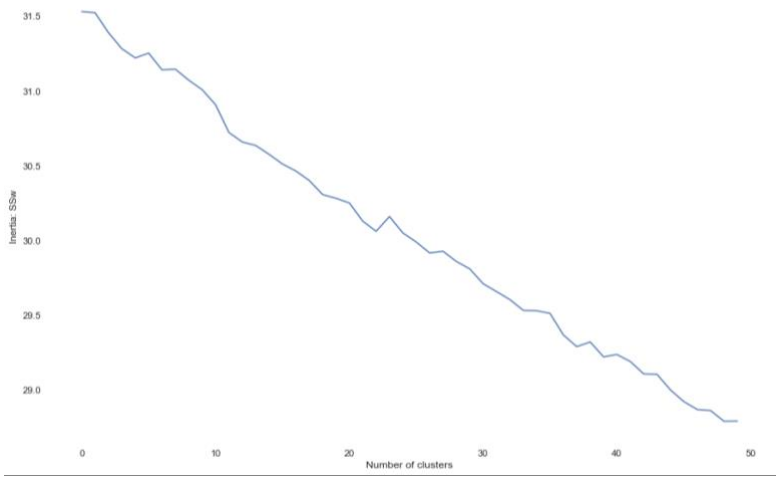


Figure 51 - WCSS for K-Means + HC Trial 3

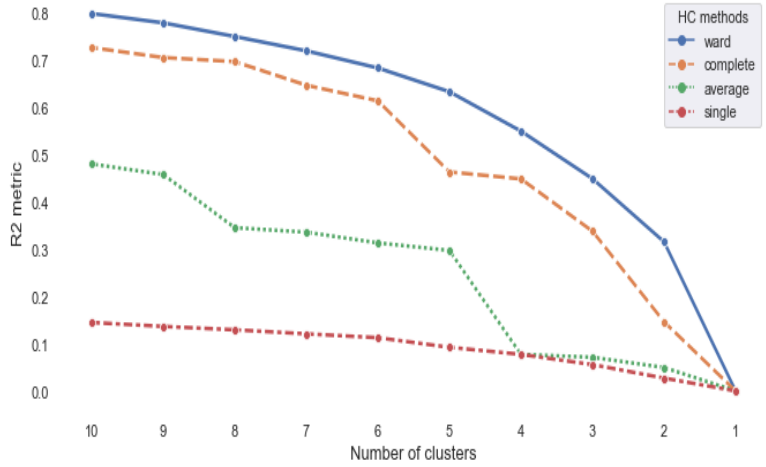


Figure 52 - R Squared for K-Means + HC Trial 3

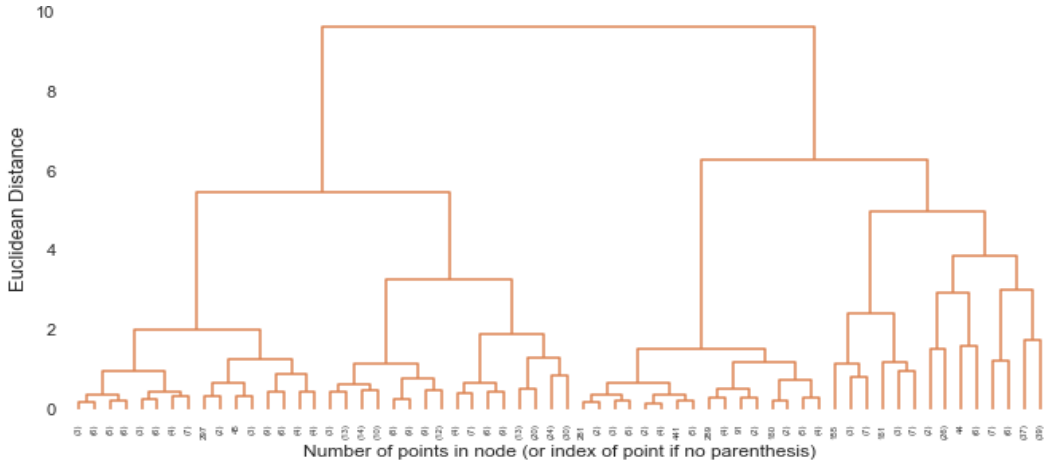


Figure 53 - Dendrogram for K-Means + HC Trial 3

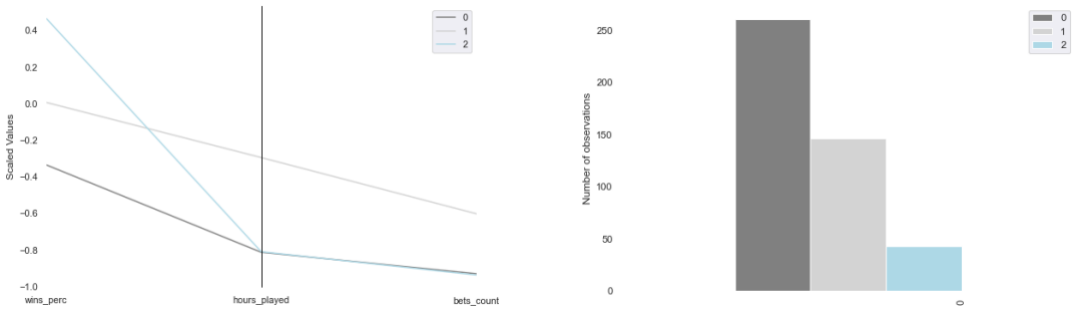


Figure 54 - Cluster Analysis for K-Means + HC Trial 3 k = 3

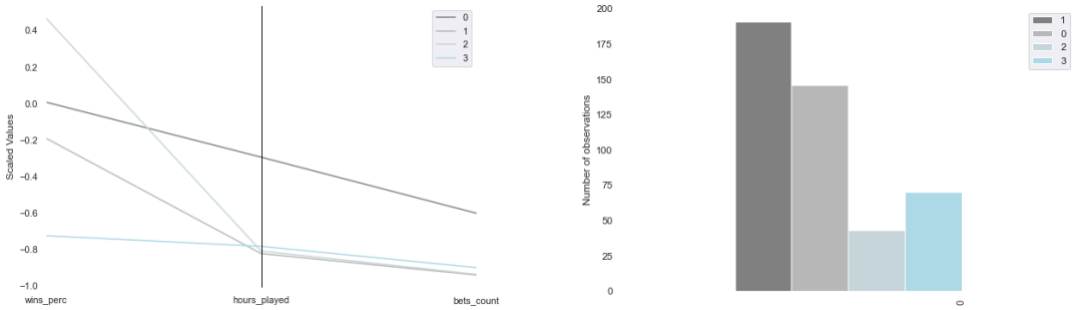


Figure 55 - Cluster Analysis for K-Means + HC Trial 3 k = 4

Comments:

Finding that 'hours_played' and 'bets_count' were features that could well explain clusters, the author kept both. Adding to that, 'wins_perc' was tested as it could be a noteworthy feature in the profiling. Although the results were not as insignificant as in Trial 1, it is also true that Trial 2 holds more explanatory power compared to Trial 1. Another issue in this trial is the number of variables to explain the players. If another meaning variable can be added to describe the different types of gamblers better, it should.

Trial 4:

Scaler: MinMax Scaler (-1, 1)

Features: ‘wins_perc, ‘hours_played’, ‘bets_count’, ‘amount_var’

Results:

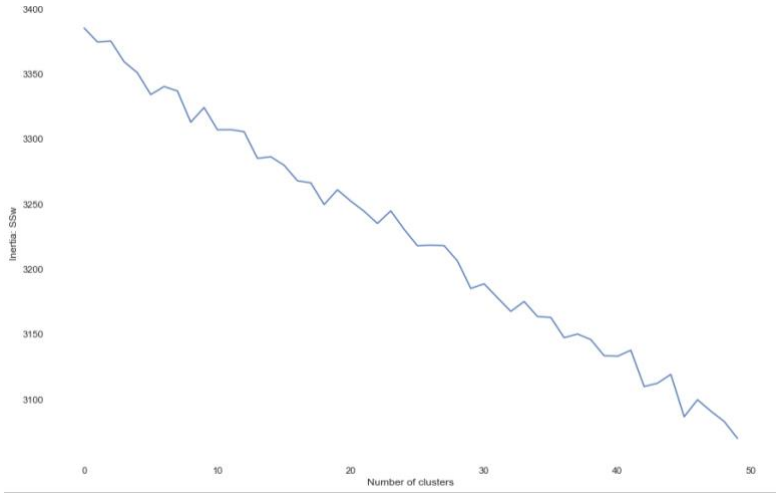


Figure 56 - WCSS for K-Means + HC Trial 4

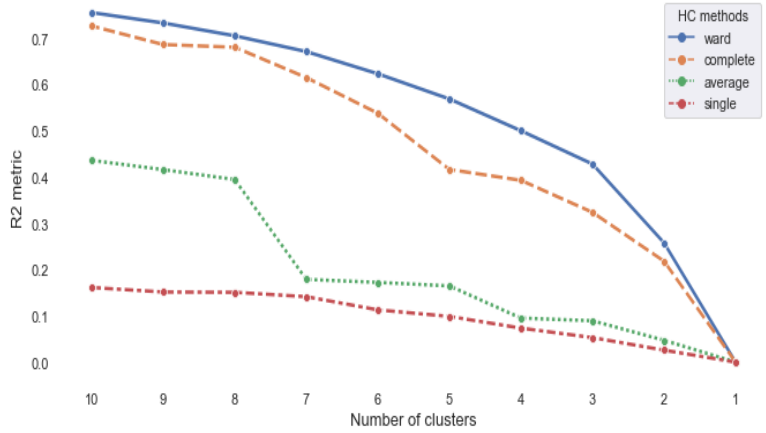


Figure 57 - R Squared for K-Means + HC Trial 4

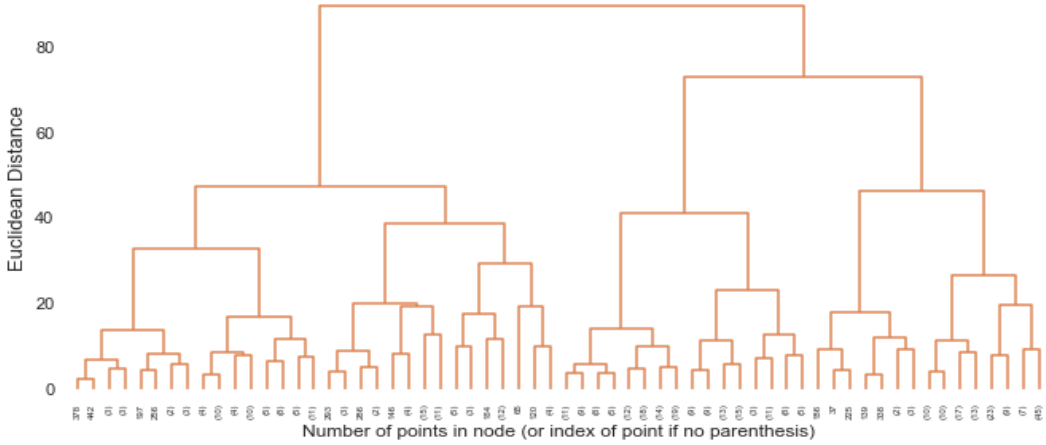


Figure 58 - Dendrogram for K-Means + HC Trial 4

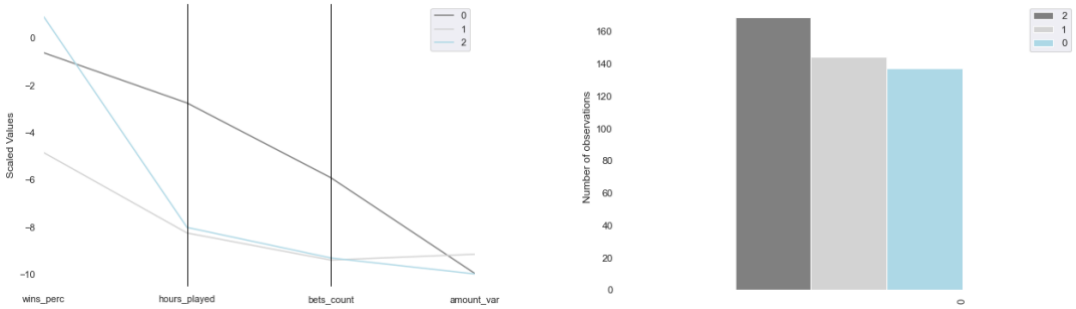


Figure 59 - Cluster Analysis for K-Means + HC Trial 4 k = 3

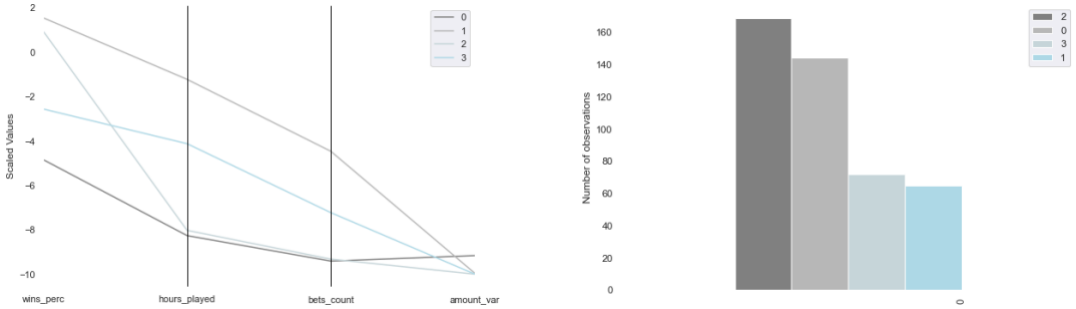


Figure 60 - Cluster Analysis for K-Means + HC Trial 4 k = 4

Comments:

In the 4th Trial of this technique, in response to the last issue raised in Trial 3, another feature joined the model – 'amount_var'. However, that variable did not add significant value to the model. Figure 48 indicates that it could have interesting results, except for 'amount_var'.

Trial 5:

Scaler: MinMax Scaler (-1, 1)

Features: 'wins_perc_perc95, 'bets_count_perc95', 'amount_var_perc95', 'perc_year_played', 'total_amount_year_perc95'

Results:

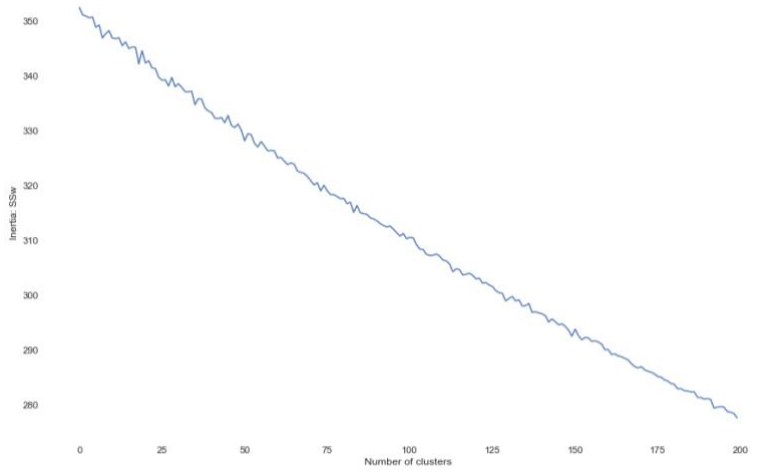


Figure 61 - WCSS for K-Means + HC Trial 5

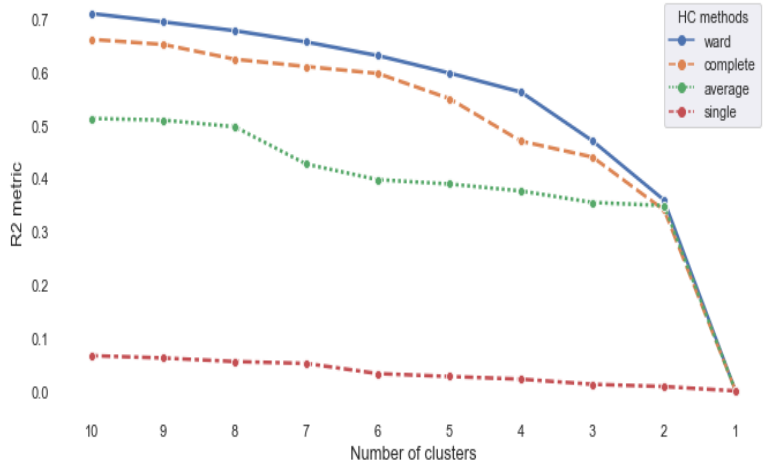


Figure 62 - R Squared for K-Means + HC Trial 5

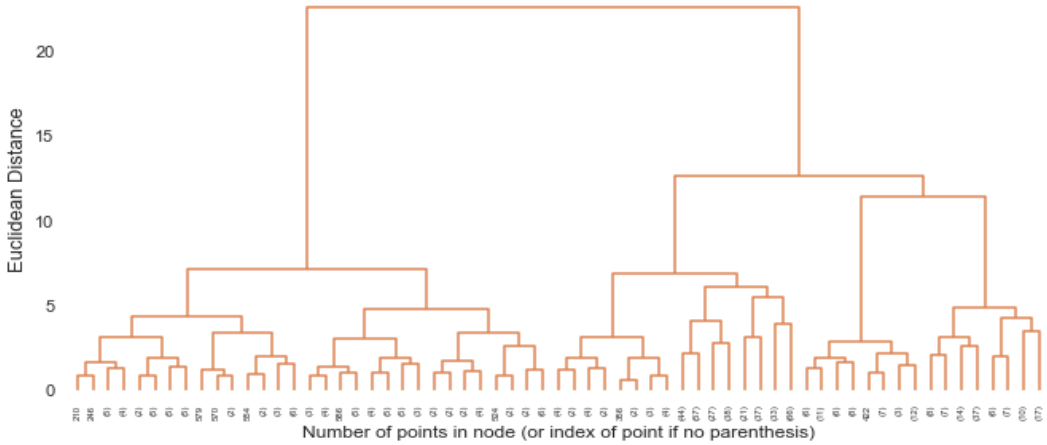


Figure 63 - Dendrogram for K-Means + HC Trial 5

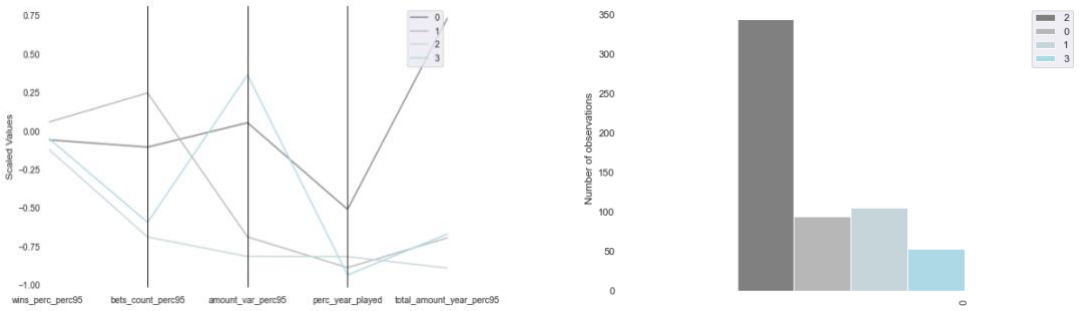


Figure 64 - Cluster Analysis for K-Means + HC Trial 5 k = 4

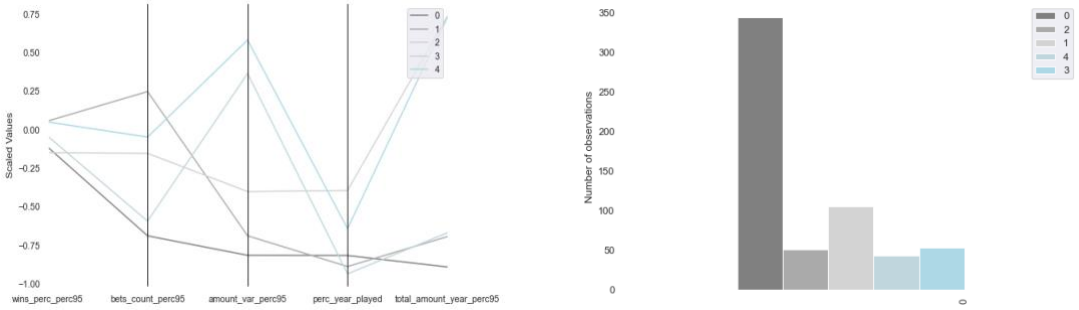


Figure 65 - Cluster Analysis for K-Means + HC Trial 5 k = 5

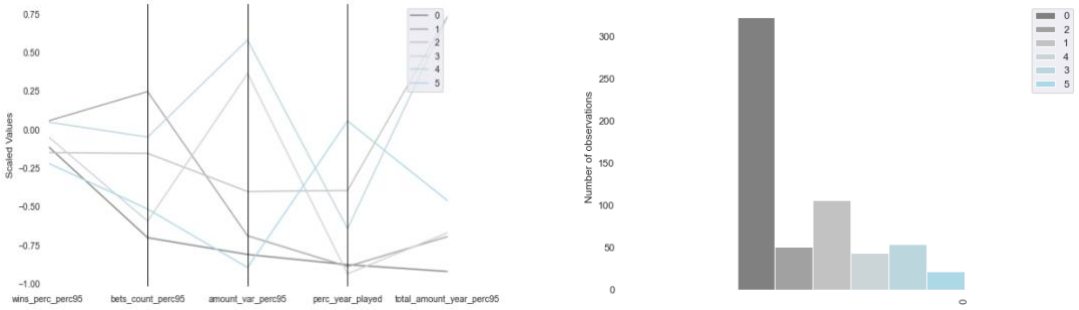


Figure 66 - Cluster Analysis for K-Means + HC Trial 5 k = 6

Comments:

At this point, the author questioned the original variables. There seemed to be too much redundancy in the chosen features, resulting in low variance in some cluster solutions. The 95th percentile was used for the selected attributes to fight that and avoid extreme values. The improvement in the results is visible in Figures 52 to 54. It was possible to achieve meaningful clusters using as many as five variables.

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa