WINE MATTERS

SEGMENTATION PROJECT

NOVA IMS
Information Management School

Descriptive Methods of Data Mining
Prof. Roberto Henriques, Lara Oliveira and Ana Aubyn

GROUP 5

Anastasia Nica, m20210516

Frederico Ferreira, m20210387

Inês Margarido, r20181131

Júlia Monteiro, m202105552

Margarida Castro, r20181099

Telma Silva, m20210558

2021/2022

INDEX

ABSTRACT

This report consisted of importing the data to do an analysis of descriptive statistics and find and clean errors among the data. Missing values, outliers and categorical values were found and corrected to get the exact information to be able to continue the report. We identified the correlations between the variables and scaled the data. We normalized the data and used various clustering algorithms to segment the data. Later on, the quality of the clusters was tested using scientific methods and, by analysing the values through visualizations, we defined the best algorithms and respective results to be used to better segment the wines and increase revenue.
The project objective was to create different wine segments, which WINE MATTERS still hasn't done, and then make catalogues for each one, based on similar characteristics.

INTRODUCTION

This project has the goal of identifying segments within the company's wines base. Therefore, we want to provide the best possible customer experience through monthly catalogues that are customised according to the customer's preferences.

By creating a plan and a quantitative approach to support marketing decisions, the WINE MATTERS company can improve its marketing efficiency in order to have better sales results and more satisfied clients as they will receive fewer but more targeted offers.

WINE MATTERS company has around 100,000 registered customers and serves more than 200,000 consumers a year with a variety of 120,000 wines, so by performing segmentation of similar paths that wines features have, we will identify solutions to overcome the problem and understand what can be done in a better way. The company presented solid revenues in the last 3 years. However, extra analysis is needed in order not to compromise the next 3 years, but to make them even more profitable.

BACKGROUND

For the development of this project, we used techniques, algorithms and methodologies learned in class but also others that we decided to research on our own, to deepen our knowledge and see if we could get better, more accurate results.

For **outlier** treatment, we used the Inter-Quartile Range to calculate the lower and upper inner and outer fences. In three cases where the variable had less than 3% of outliers, we decided to remove them. In the other two variables where we detected outliers, either through visualizations (e.g. boxplots) or skewness analysis, we replaced the values that were higher than the inner fence with the upper adjacent value, because they were affecting our distributions in a way that would negatively impact the identification of clusters. Additionally, we also maintain some outliers that we considered relevant to this study.

We decided to use **KNNImputer**, an algorithm present in the scikit-learn python library that uses a multivariate approach to treat missing values. It fills out the missing values using the mean of the k closest neighbours and according to the features of those records. The distance between samples is calculated using the "nan-euclidian" metric by default, which computes the standard euclidian distance where possible and nothing (NaN) where either of both observations is missing. In terms of the “n-neighbors” parameter, we decided to use another algorithm to find the best value possible – **KNeighborsClassifier**.

With KNeighborsClassifier, another scikit-learn class, we were able to study the best value for k by running the algorithm multiple times for different values and capturing the respective error and accuracy rate. With this information, we produced some visualizations and found the best value for the number of neighbors to be used in KNNImputer.

For the categorical variables with missing values, we used **SimpleImputer**, from the scikit-learn library, because these were simpler variables that only assumed two values each and had a low percentage of missing values. The strategy used was “most frequent”, so the missing values were filled in with the mode of each variable.

Using this technique, the values imputed will have less impact on the distribution of the data and will be more diverse, when compared to basic mean/median imputation.

Additionally, we used heuristics to get the initial values to use as the parameters of the algorithms. This allowed us to have a solid starting point instead of just using random values until we eventually reached the ones that made more sense.

Regarding **DBSCAN**, to select optimal values for minPts, there is a heuristic minPts = 2 * dimensions of the dataset (Sander et al., 1998). To find the best eps value, we can calculate the average distance between each data point and its k nearest neighbors and plot them in ascending order, through a K-distance graph. To select k, we can use k = (2 * dim − 1). Analyzing the graph, the ideal value for eps can be found close to the "first valley". All values positioned to its left are considered noise, while the ones to the right are assigned to a cluster. Nevertheless, it is advisable to test different values of eps and evaluate the results.

DBSCAN does not work well in datasets with high dimensions, being affected by the curse of dimensionality. Consequently, first we performed **Principal Component Analysis (PCA)**, a technique to reduce the dimensionality of datasets, while maintaining the maximum amount of information possible, which is measured through variance. The original variables are substituted with new ones, principal components, that are uncorrelated and linear combinations of the original variables.

To perform PCA, the data needs to be standardized. Then, the principal components are extracted through the covariance matrix. In sklearn, the principal components are computed through Singular Value Decomposition, a matrix factorization technique. To decide the number of principal components, there are several methods: Kaiser rule, screen plot or the variance explained criteria. We adopted the latter, where this number is chosen according to a threshold between 70% to 90% of explained variance (Jolliffe 2002). Finally, we can plot the reduced data and interpret the results.

MEDODOLOGY

Business Needs

Although for the past three years the company has had solid revenues, next years' growth seems less appealing. For this reason, the company has thought of a few strategies to improve it. One of them focuses on increasing the efficiency of the marketing campaigns. Our main goal is to identify different segments to be able to create different catalogues based on the characteristics of those segments.

We were provided a file with historical information about the characteristics of the wine and our client satisfaction, as you can see in table 1. This file has 129881 entries containing the respective information about each wine.

Variable	Description
WineID	Identifier of the wine
Aging_Time	Aging time of the wine in months
Litters_Barrel	Number of liters contained in each of the barrels while aging
Type	Type of the wine (red or white)
Magnesium	Grams of magnesium per 1000 litters of wine
Residual_Sugar	Grams of sugar per 1000 litters of wine
Sulphites	Indicates if there are sulphites, just a few or none at all in the wine
Barrel	Type of barrel used for aging (wood or aluminium)
Grapes	Grape varieties used for the wine (mixed grapes or single grape)
Acidity	Degree of acidity in wine (0 to 5 where 5 is the most acid)
Floral	Presence of floral tones in the wine (0 to 5)
Wood	Presence of wood tones in the wine (0 to 5)
Sweetness	Feeling of sweetness in the wine (0 to 5)
Red_Fruit	Presence of red fruit tones in the wine (0 to 5)
Citric	Presence of citric tones in the wine (0 to 5)
Density	Density of the wine (0 to 5 where 5 is the densest)
Color_Intensity	Color intensity of the wine (0 to 5 where 5 is the most intense)
Cloudiness	Cloudiness of the wine (0 to 5 where 5 is the most cloudy)
Alcohol	Alcohol taste in the wine (0 to 5)
Astringency	Astringency of the wine (0 to 5)
Satisfaction_France	Average satisfaction level of customers from France (0 to 10)
Satisfaction_Spain	Average satisfaction level of customers from Spain (0 to 10)
Satisfaction_Portugal	Average satisfaction level of customers from Portugal (0 to 10)

Table 1 - Variable's description

Data Exploring and Understanding

After importing our dataset to the Jupyter Notebook we started exploring it. First, we saw that it has 23 columns, one of them being the WineID. This variable is the identifier of the wine, so it is not correct to include it in our analysis, instead, we set it as the index.

Additionally, it was also important to understand which type of data we were working with. From the table 2 we can conclude that we have four categorical variables (data type objects) and all others are numerical, with one having fraction numbers (data type float) and the rest integers (data type int).

We can also realize from table 2 that we have missing values in the variables Residual_Sugar, Barrel and Grapes.

```
RangeIndex: 129881 entries, 0 to 129880
Data columns (total 23 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   WineID                129881 non-null  int64
 1   Aging_Time            129881 non-null  int64
 2   Litters_Barrel        129881 non-null  int64
 3   Type                  129881 non-null  object
 4   Magnesium             129881 non-null  int64
 5   Residual_Sugar        129490 non-null  float64
 6   Sulphites             129881 non-null  object
 7   Barrel                129825 non-null  object
 8   Grapes                129873 non-null  object
 9   Acidity               129881 non-null  int64
 10  Floral                129881 non-null  int64
 11  Wood                  129881 non-null  int64
 12  Sweetness             129881 non-null  int64
 13  Red_Fruit             129881 non-null  int64
 14  Citric                129881 non-null  int64
 15  Density               129881 non-null  int64
 16  Color_Intensity       129881 non-null  int64
 17  Cloudiness            129881 non-null  int64
 18  Alcohol               129881 non-null  int64
 19  Astringency           129881 non-null  int64
 20  Satisfaction_France   129881 non-null  int64
 21  Satisfaction_Spain    129881 non-null  int64
 22  Satisfaction_Portugal 129881 non-null  int64
dtypes: float64(1), int64(18), object(4)
memory usage: 22.8+ MB
```

Table 2 - Variable's data type

To continue exploring our data, we did the descriptive statistics for numerical variables, as you can see in table 3. Although our wines have, on average, an aging time of 39 months with 1981 litters per barrel, we need to be aware of the high dispersion of the data represented by the high value in the standard deviation parameter. Additionally, a large percentage of our wines have low levels of magnesium grams and residual sugar per 1000 litters of wine since we can see that 50% of the data has the value 0 and 75% has until 12 or 13 grams, respectively. With respect to the taste of our wines, all characteristics are between 2.5 and 3.5 which represents a medium presence/degree of those flavors.

	Count	Mean	Standard Deviation	Min	25%	50%	75%	Max
WineID	129881	64940	37493.5548	1	32470	64940	97410	129880
Aging_Time	129881	39.4279	15.119312	7	27	40	51	85
Litters_Barrel	129881	1981.42	1027.11903	50	1359	1925	2544	6951
Magnesium	129881	14.7136	38.071002	0	0	0	12	1592
Residual_Sugar	129490	15.0908	38.465273	0	0	0	13	1584
Acidity	129881	3.48591	1.292228	0	2	4	5	5
Floral	129881	3.34082	1.260586	0	3	3	4	5
Wood	129881	2.85197	1.443746	0	2	3	4	5
Sweetness	129881	2.99042	1.305965	0	2	3	4	5
Red_Fruit	129881	3.24911	1.318827	0	2	3	4	5
Citric	129881	3.38345	1.346087	0	2	4	4	5
Density	129881	3.51968	1.306524	0	3	4	5	5
Color_Intensity	129881	3.47209	1.305573	0	2	4	5	5
Cloudiness	129881	3.46674	1.273503	0	3	4	4	6
Alcohol	129881	2.83858	1.393	0	2	3	4	5
Astringency	129881	2.99065	1.527221	0	2	3	4	5
Satisfaction_France	129881	7.39135	2.312959	2	6	8	10	10
Satisfaction_Spain	129881	7.41152	2.30354	0	6	8	10	10
Satisfaction_Portugal	129881	6.70514	2.597452	0	4	8	8	10

Table 3 - Descriptive Statistics for numerical variables

As for the categorical data, present in table 4, we can conclude that most of our wines are white, have the presence of sulphites, aged in an aluminium barrel, and were produced with mixed grapes.

	Type	Sulphites	Barrel	Grapes
Count	129881	129881	129825	129873
Unique	2	4	2	2
Top	White	Present	Aluminium	Mixed
Freq	65900	62160	106045	89685

Table 4 - Descriptive Statistics for categorical variables

Data Preprocessing

Incoherences

After exploring and understanding our data we started treating it. In the descriptive statistics, we observed three incoherences in the data.

Regarding the 'Cloudiness' variable, it is stated in the project description that its values should range from 0 to 5. However, by looking at the descriptive statistics we noticed it had a maximum value of 6. In order to correct this, we replaced all 6's by the maximum value that the scale allowed, 5.

Color_Intensity	Cloudiness	Alcohol
129881.000000	129881.000000	129881.000000
3.472086	3.466743	2.838575
1.305573	1.273503	1.393000
0.000000	0.000000	0.000000
2.000000	3.000000	2.000000
4.000000	4.000000	3.000000
5.000000	4.000000	4.000000
5.000000	6.000000	5.000000

Table 5 - Descriptive statistics of Color_Intensity, Cloudiness and Alcohol variables

By observing table 4, we noticed that the Sulphites variable has 4 unique types when it could only have 3 (Present, Not Present and Very Few). After discovering that the fourth type was "0", and if 0 represented that the wine does not have the presence of Sulphites, we changed all "0"s to "Not Present".

	Type	Sulphites	Barrel	Grapes
count	129881	129881	129825	129873
unique	2	4	2	2
top	White	Present	Aluminium	Mixed
freq	65900	62160	106045	89685

Table 6 - Descriptive statistics of Type, Sulphites, Barrel and Grapes variables

We also found duplicate rows in our dataset, as table 5 shows, and since it is not correct to have multiple entries for the same wine, we dropped the repeated records.

WineID	Aging_Time	Litters_Barrel	Type	Magnesium	Residual_Sugar	Sulphites	Barrel	Grapes	Acidity	Floral	...
354	33	3384	White	0	0.0	Very Few	Aluminium	Single	5	5	...
354	33	3384	White	0	0.0	Very Few	Aluminium	Single	5	5	...

Table 7 - Duplicate data

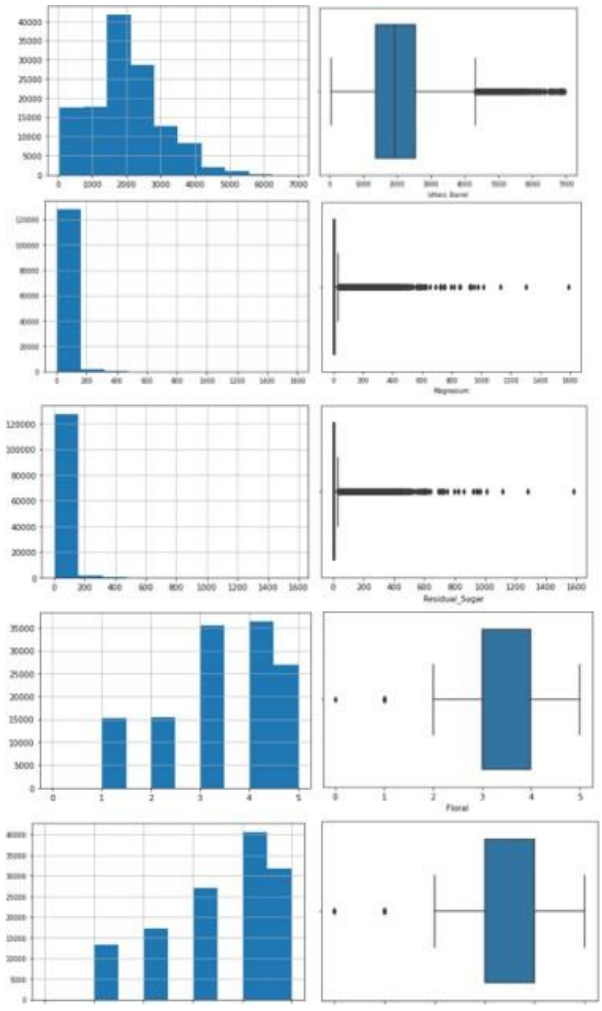
Outliers

Moving on from incoherences, we started detecting and treating outliers that can harm and invalidate our conclusions. We first analyse the skewness which, from the table below, we can see that we have two highly skewed variables, Magnesium and Residual_Sugar, all others have a normal distribution.

Aging_Time	-0.003606
Litters_Barrel	0.466748
Magnesium	6.821980
Residual_Sugar	6.670170
Acidity	-0.496440
Floral	-0.392442
Wood	-0.116813
Sweetness	-0.053064
Red_Fruit	-0.191123
Citric	-0.604828
Density	-0.575365
Color_Intensity	-0.491720
Cloudiness	-0.505270
Alcohol	-0.091861
Astringency	-0.252282
Satisfaction_France	-0.743037
Satisfaction_Spain	-0.756001
Satisfaction_Portugal	-0.366496
dtype: float64	

Table 8 - Initial skewness

Next, to observe the outliers, we created boxplots and histograms.



Graph 1 - Histograms and Boxplots graphs

Having in mind that it is incorrect to delete **more than 3%** of our data, 3896 rows in this case, we started by studying the inner and outer fences in each of the variables. To do this, we calculate the IQR that is the difference between quartile 3 (Q3) and quartile 1 (Q1), and then the fences. The inner fence is calculated by subtracting (IQR*1.5) to the Q1 and adding (IQR*1.5) to the Q3. As for the outer fence, it is equal to the inner fence, except we subtract and add (IQR*3). With these measures we can detect two types of outliers, the **mild** ones that are between the inner and outer fence and the **extreme outliers** that are beyond the outer fence.

Then we started treating the outliers. For the Litters_Barrel variable, we counted how many outliers were present (higher than the inner fence), with the result being 2581 rows, which is below 3% of the data. For this reason, we decided to **delete** these records. For the Magnesium and Residual Sugar variables, we first thought about replacing all outliers by the median, but because both medians had the value 0 (also their minimum was 0) and the outliers range lied between 30 and 1592 and 32.5 and 1584, respectively, it would not be the best practice to replace them with 0. So, our solution to better represent our data without the outliers was to replace all of them with the upper adjacent value (Q3 + (IQR*1.5)), which were 30 and 32, respectively. This way, we treat the outliers, but we still preserve the fact that all of them were bigger than 0.

For the last two variables, Floral and Cloudiness, we also counted the outliers for ratings 1 and 0. We notice that, for rating 1, both variables had a significant quantity of rows, 15369 and 13275, respectively. Since for these variables we only have 6 possible ratings (0-5), we considered that replacing the rating 1 by any other would not be the most correct approach, so we decided to **keep** the outliers present in the inner fence because they were relevant for our clustering study. As for the rating 0 we decided to **delete** those rows, that corresponded to 1 and 5 outliers, respectively. So, in total we dropped 2587 rows, which is still **below the 3%**.

Finally, we reviewed our descriptive statistics, table, and we can see an improvement in our standard deviation value compared with the ones in our initial descriptive statistics.

	Count	Mean	Standard Deviation	Min	25%	50%	75%	Max
Litters_Barrel	127296	1922.08421	946.06197	50	1340	1904	2496	4321
Magnesium	127296	7.316664	11.163035	0	0	0	12	30
Residual_Sugar	126917	7.801878	11.793437	0	0	0	13	32
Floral	127296	3.341433	1.260217	1	3	3	4	5
Cloudiness	127296	3.469339	1.269995	1	3	4	4	5

Table 9 - Descriptive statistics after outlier treatment

And for the skewness analysis, we can also see a huge improvement of our two highly skewed variables – Magnesium and Residual_Sugar.

Aging_Time	-0.028114
Litters_Barrel	0.150718
Magnesium	1.232166
Residual_Sugar	1.231915
Acidity	-0.502009
Floral	-0.392412
Wood	-0.116881
Sweetness	-0.053757
Red_Fruit	-0.182791
Citric	-0.602666
Density	-0.574380
Color_Intensity	-0.490106
Cloudiness	-0.509505
Alcohol	-0.095569
Astringency	-0.255591
Satisfaction_France	-0.742026
Satisfaction_Spain	-0.757545
Satisfaction_Portugal	-0.361418
dtype: float64	

Table 10 - Skewness after outlier treatment

Dummy variables

To use the K-Nearest Neighbors algorithm as well as the clustering algorithms later, we needed to encode our categorical variables. For that, we used the ***get_dummies*** pandas' function, which allowed us to perform one-hot encoding in a variable that assumed more than two values (Sulphites). The *drop_first* parameter was set to true, to have k-1 new columns, where k is the number of unique values the variable has. In the case of variables that only assumed two different values (Type, Barrel and Grapes), we created new dummy columns where the values of each variable were encoded as 0 and 1.

Missing values

After treating our outliers, it is time to fill in the missing values in our dataset. We noticed that there were 3 variables with missing values – Residual_Sugar, Barrel and Grapes.

Aging_Time	0.000000
Litters_Barrel	0.000000
Type	0.000000
Magnesium	0.000000
Residual_Sugar	0.297724
Sulphites	0.000000
Barrel	0.043991
Grapes	0.006284
Acidity	0.000000
Floral	0.000000
Wood	0.000000
Sweetness	0.000000
Red_Fruit	0.000000
Citric	0.000000
Density	0.000000
Color_Intensity	0.000000
Cloudiness	0.000000
Alcohol	0.000000
Astringency	0.000000
Satisfaction_France	0.000000
Satisfaction_Spain	0.000000
Satisfaction_Portugal	0.000000

Table 11 - Percentage of missing values

In the case of Barrel and Grapes, since they were categorical variables, we decided to use **SimpleImputer** with the strategy "most_frequent", so that all missing values are replaced with the most frequent value in that variable.

In the case of Residual_Sugar, which was the only numerical variable of the three and the one with more missing values (~0.3%), we decided to impute its missing values using **KNNImputer**, which uses the k-nearest neighbors of the record with the missing value to compute the most adequate value. To do that, we first studied what was the best number of neighbors to consider in the algorithm using **KNeighborsClassifier**. We trained the model using our dataset without the missing values, tried multiple values between 1 and 100, and found that the best value for k was 18 (it was the value with less error and more accuracy, according to our results). Then, we used the KNNImputer with n_neighbors=18 as a parameter and filled in all our missing values.

New variables

To better segment the wines, several variables were created. First, one that provides the wines' aging time in years, being more intuitive to interpret. Also, a variable that returns the overall average satisfaction level, taking in consideration all the three countries (France, Spain, and Portugal).

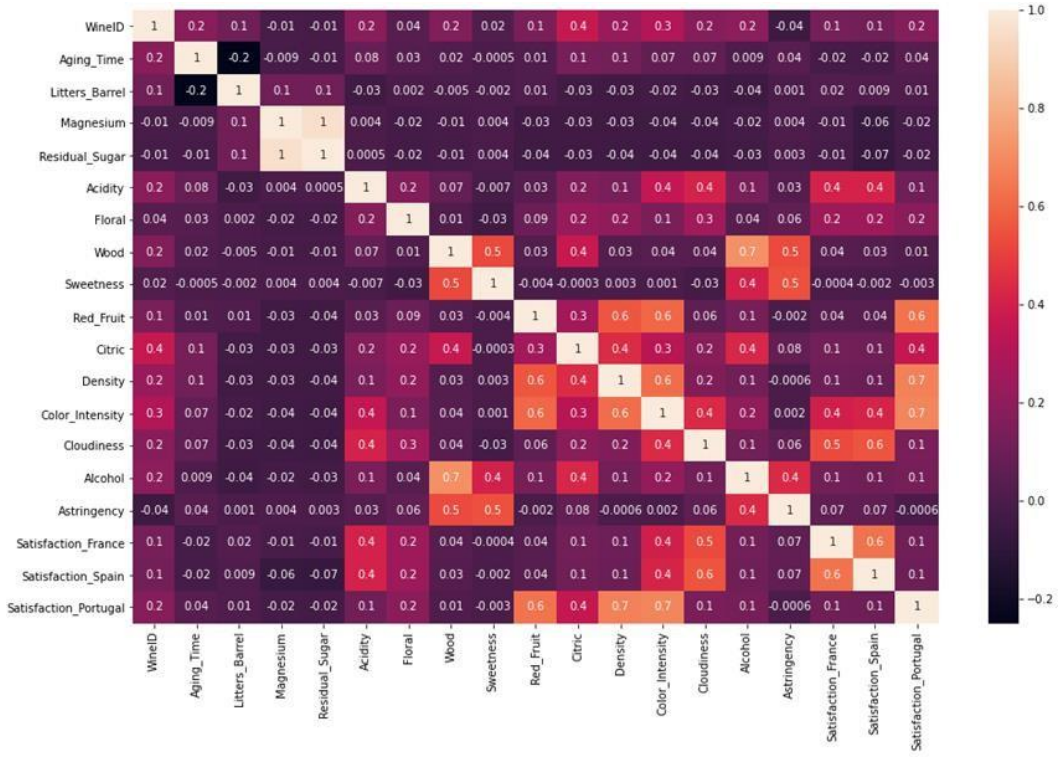
Since the original variables Magnesium and Residual_Sugar represents the grams per 1000 liters of wine, we wanted to have the grams per the number of liters in each barrel. Likewise, we created two variables that store the concentration of residual sugar and magnesium per liter of wine (%).

Variable	Description	Formula
Aging_Time_Years	Aging time of the wine in years and rounded to the unit.	=round(WineMatters2['Aging_Time']/12)
Satisfaction_overall	Overall average satisfaction level of customers in the 3 countries: Portugal, France, and Spain.	=WineMatters2[['Satisfaction_France','Satisfaction_Spain', 'Satisfaction_Portugal']].mean(axis=1)
Flavors_avg	Average rating of all flavors.	= WineMatters2[['Acidity','Floral', 'Wood','Sweetness', 'Red_Fruit', 'Citric','Density', 'Color_Intensity','Cloudiness','Alcohol','Astringency']].mean(axis=1)
Magnesium_grams_barrel	Grams of Magnesium per the number of liters in each barrel.	= WineMatters2['Magnesium'] /1000 * WineMatters2['Litters_Barrel']
Residual_sugar_barrel	Grams of Residual sugar per the number of liters in each barrel.	= WineMatters2['Residual_Sugar'] /1000 * WineMatters2['Litters_Barrel']
Magnesium_perc	Concentration of Magnesium per liter in percentage	= WineMatters2['Magnesium'] * 0.1
Residual_sugar_perc	Concentration of Residual Sugar per liter in percentage	= WineMatters2['Residual_Sugar'] * 0.1

Table 12 - New variables

Correlations

Observing the heatmap we can detect a high correlation between Satisfaction_Portugal and Density, Satisfaction_Portugal and Color_Intensity, Alcohol and Wood.



Graph 2 - Correlation between all original variables

Scaling

To finish preprocessing our data we also need to scale it. This step is necessary because the algorithms used to identify clusters in each segment depended on the distance between each observation.

To avoid giving more weight to some observations we applied the **minmax** scaler algorithm to the production characteristics segment, since our flavor segment already had all variables in the same scale (0-5). This normalizes the data to a scale between 0 and 1, preserving its shape and making it possible to analyse this data together with categorical variables encoded as 0s and 1s.

WineID	Aging_Time	Litters_Barrel	Type_Dummy	Magnesium	Residual_Sugar	Sulphites_Present	Sulphites_Very Few	Barrel_Dummy	Grapes_Dummy
1	0.743590	0.050339	1.0	0.0	0.0	0.0	0.0	0.0	1.0
2	0.512821	0.565207	0.0	0.0	0.0	1.0	0.0	0.0	1.0
3	0.102564	0.488878	1.0	0.0	0.0	0.0	0.0	0.0	1.0
4	0.679487	0.134161	1.0	0.0	0.0	0.0	0.0	0.0	1.0
5	0.807692	0.071178	1.0	0.0	0.0	0.0	0.0	0.0	1.0
...	...	...	...	...	...	...	...	...	...
129876	0.282051	0.393585	1.0	0.0	0.0	0.0	0.0	1.0	1.0
129877	0.717949	0.476937	0.0	0.0	0.0	1.0	0.0	1.0	1.0
129878	0.794872	0.531491	0.0	0.0	0.0	0.0	0.0	1.0	1.0
129879	0.756410	0.561929	0.0	0.0	0.0	0.0	0.0	1.0	1.0
129880	0.397436	0.996722	1.0	0.0	0.0	0.0	0.0	1.0	1.0

127299 rows × 9 columns

Table 13 - Final dataset after pre-processing of the data

Modelling

Perspectives

To begin the clustering algorithms, we first divided our data into two segments that we considered the most important to this study.

Our first perspective is the **Flavor/Felling** of the wine where we used 11 variables: *Acidity, Floral, Wood, Sweetness, Red_Fruit, Citric, Density, Color_Intensity, Cloudiness, Alcohol, and Astringency*. In this segment we aim to understand the different presence/degree of flavors in each cluster.

Our second perspective is the **Production Characteristics** of the wine where we used 9 variables: *Aging_Time, Litters_Barrel, Type_Dummy, Magnesium, Residual_Sugar, Sulphites_Present, Sulphites_Very Few, Barrel_Dummy, Grapes_Dummy*. In this segment, our goal is to understand the different types of production characteristics of each cluster.

For each perspective we used the following clustering algorithms: Hierarchical Clustering, K-Means, Self-Organizing Maps (SOM) with K-Means, BIRCH with agglomerative, BIRCH with K-Means and DBSCAN.

Clusters Evaluation

To evaluate the quality of our clusters, we used the Silhouette Score. It measures the goodness of a clustering algorithm and ranges from -1 to 1, where 1 means the clusters are well-defined, cohesive and separated, 0 means the distance between clusters is not significant and -1 means some observations weren't correctly assigned to the clusters.

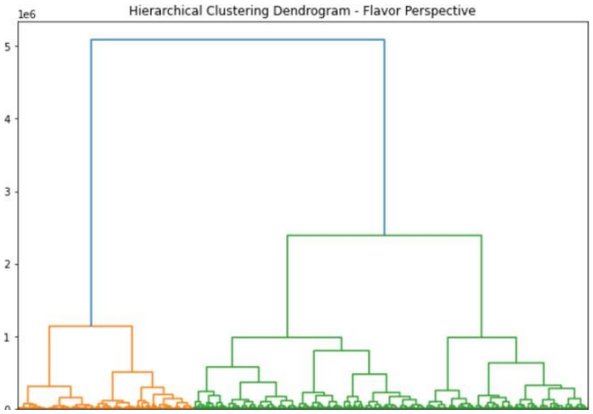
RESULTS

Hierarchical Clustering

In order to avoid memory errors during the agglomerative clustering method, we decided to create a sample of the dataset, for both perspectives, that consists of 25% of the entire dataset while not losing information, using the sample() function.

Perspective: Flavor/Feeling

Initially, we created a dendrogram by using the linkage function which calculates the distances between every combination of data points within the dataset, using the ward method. After analysing the dendrogram, we agreed that the best option would be 2 clusters, since it represents the largest vertical distance between nodes.



Graph 3 - Dendrogram of the flavor/feeling perspective

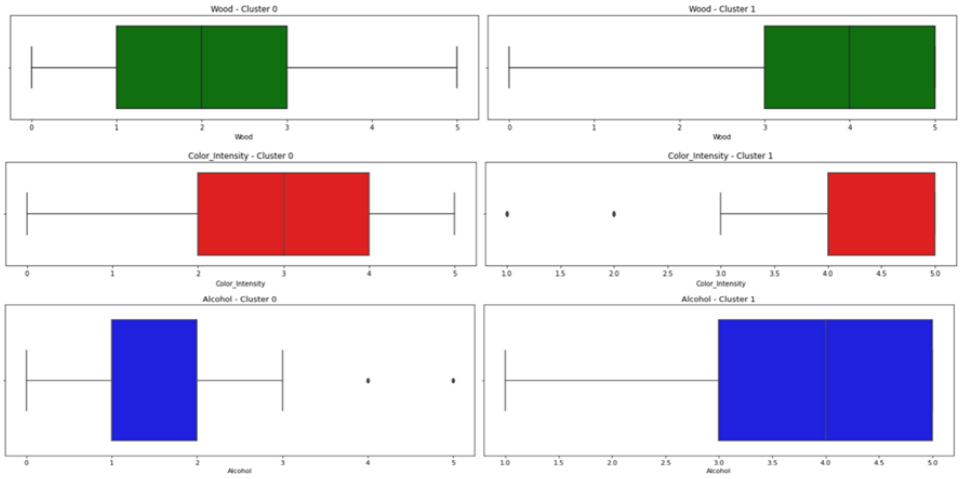
We also decided to confirm it with the silhouette score values for different numbers of clusters, being the highest score the best option, which in this case is also 2 clusters.

```
For n_clusters = 2 The average silhouette_score is : 0.15914260823794996
For n_clusters = 3 The average silhouette_score is : 0.13161623793997923
For n_clusters = 4 The average silhouette_score is : 0.11042817606973278
For n_clusters = 5 The average silhouette_score is : 0.11065321148038162
```

Table 14 - Silhouette scores for Hierarchical Clustering algorithm in flavor/feeling perspective

Then, we applied the agglomerative clustering algorithm with the 2 clusters and

decided to analyse the clusters with the descriptive statistics table and boxplots. Below there are some of the boxplots to give an idea of what the clusters represent.



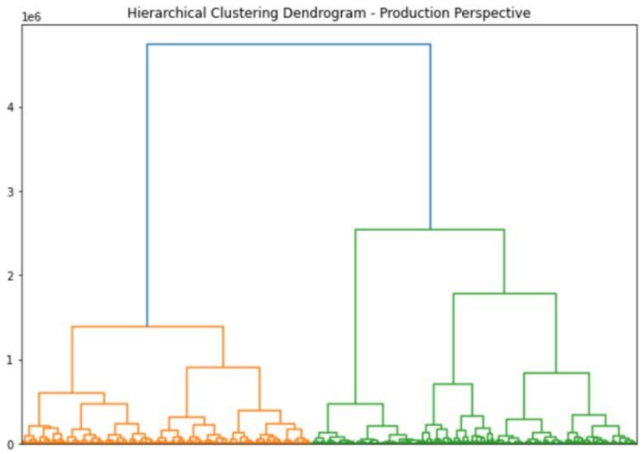
Graph 4 - Distribution of the cluster data in Wood, Color_Intensity and Alcohol variables

Cluster 0 represents wines with lower values of wood tones, sweetness, red fruit tones, citric tones, density, color intensity, alcohol taste and astringency.

Cluster 1 represents wines with higher values of wood tones, sweetness, red fruit tones, citric tones, density, color intensity, alcohol taste and astringency.

Perspective: Production Characteristics

After creating the dendrogram, we, again, agreed that the best option for the production perspective would be 2 clusters.



Graph 5 - Dendrogram of the production perspective

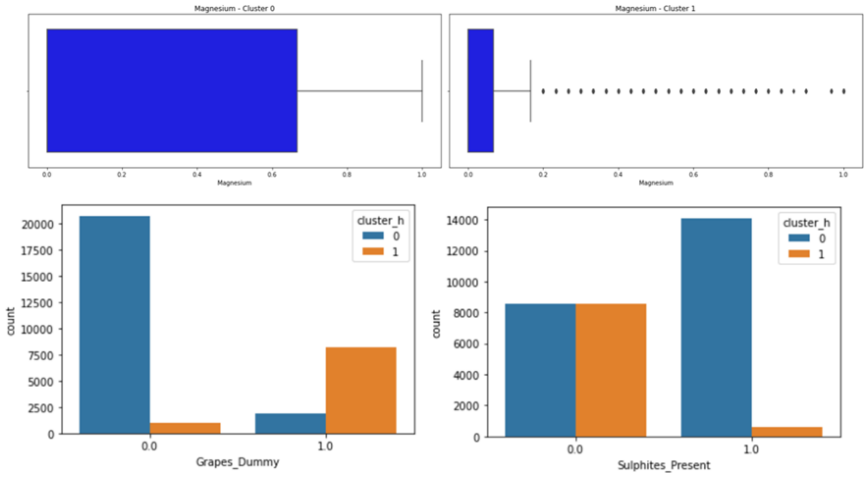
We also decided to confirm it with the silhouette score values for different number

of clusters, being the highest score the best option, which in this case is also 2 clusters.

```
For n_clusters = 2 The average silhouette_score is : 0.32024730207195573
For n_clusters = 3 The average silhouette_score is : 0.319577419258001
For n_clusters = 4 The average silhouette_score is : 0.3007506495707087
For n_clusters = 5 The average silhouette_score is : 0.2967960012755483
```

Table 15 - Silhouette scores for Hierarchical Clustering algorithm in production perspective

Then, we applied the agglomerative clustering algorithm with 2 clusters and analyzed them with the descriptive statistics table, boxplots and countplots for categorical and dummy variables. Underneath, there are some of the boxplots and countplots of the analysis in order to give an idea of what the clusters represent.



Graph 6 - Distribution of the cluster data in Magnesium, Grapes_Dummy and Sulphites_Present variables

Cluster 0 represents wines with a high presence of sulphites and low presence of grapes.

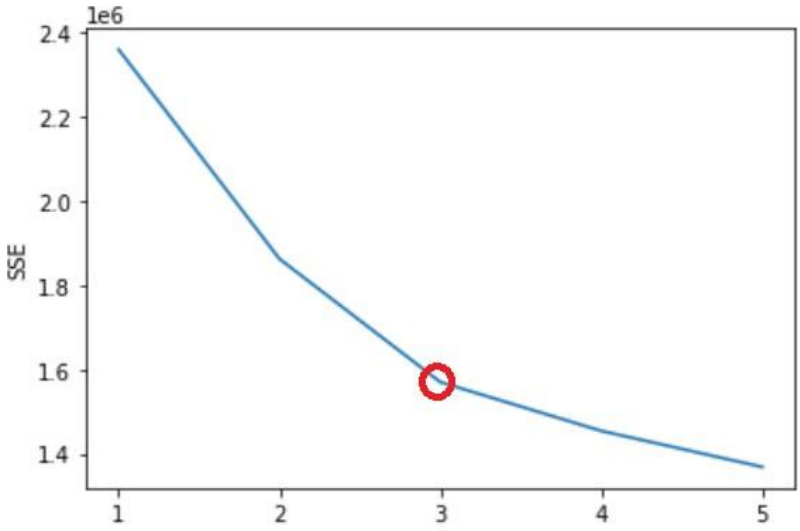
Cluster 1 represents wines with a high presence of grapes and low values of sugar, magnesium and sulphites.

K-Means

Perspective: Flavor/Feeling

First, we started by applying the Elbow method. By analysing the Elbow graph, we considered that the point where the SSE stops decreasing a lot is when K = 3 (three

clusters), which means that we will not have much more information if we add one more cluster.



Graph 7 - Elbow graph for flavor/feeling perspective

These results are confirmed with the silhouette score values for various numbers of clusters, although these are very low.

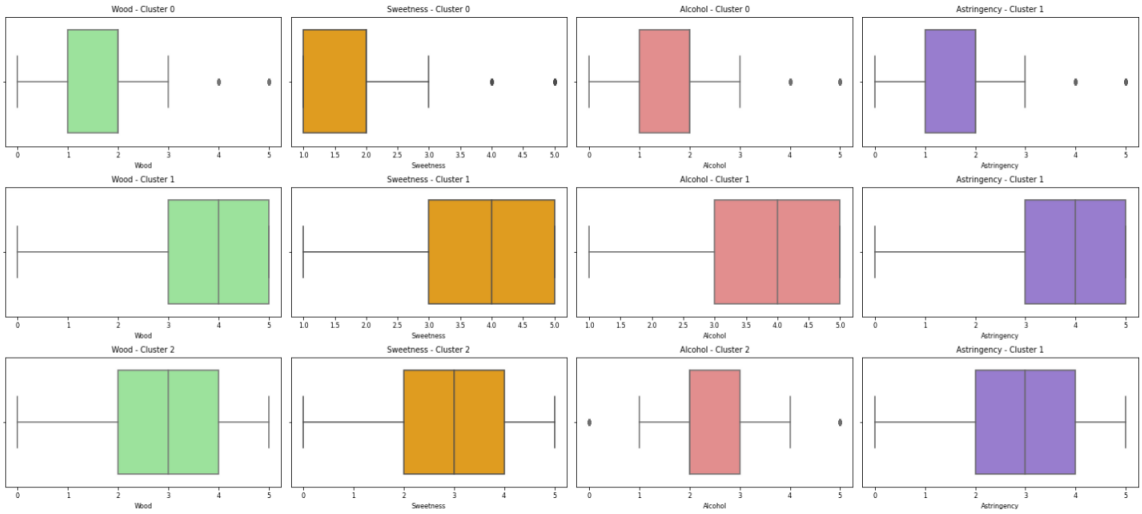
```
For n_clusters = 2 The average silhouette_score is : 0.1808529310565072
For n_clusters = 3 The average silhouette_score is : 0.18803647451727312
For n_clusters = 4 The average silhouette_score is : 0.1580363975542093
For n_clusters = 5 The average silhouette_score is : 0.15760951158499859
For n_clusters = 6 The average silhouette_score is : 0.1403441490520828
```

Table 16 - Silhouette scores for K-Means algorithm in flavor/feeling perspective

Our conclusion based on the Elbow Method and the silhouette score is that 3 clusters is the perfect fit for our data.

Next, we applied the algorithm with the correct number of clusters and started analysing the clusters using boxplots. We concluded that the customers were divided into 3 clusters based on the variables we identified earlier.

Looking at these variables, we were able to identify the clusters as follows:



Graph 8 - Distribution of the cluster data in Wood, Sweetness, Alcohol and Astringency variables

Cluster 0 is characterized by wines with **low** values of wood tones, sweetness, alcohol, and astringency.

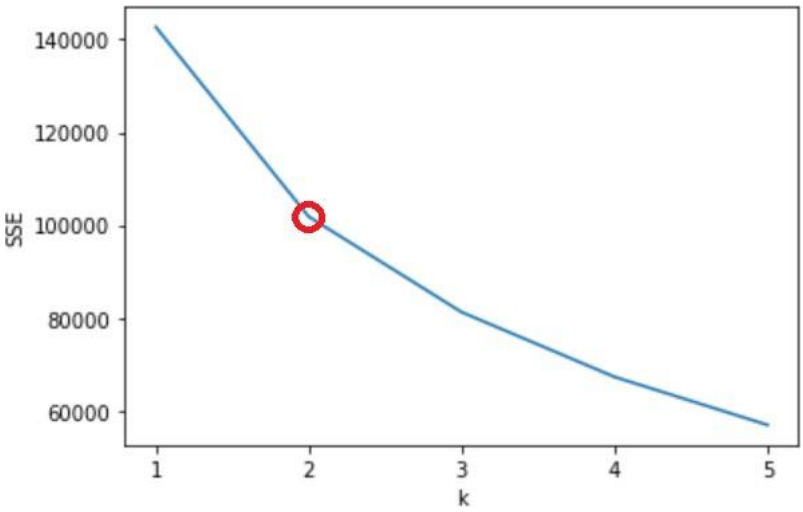
Cluster 1 is characterized by wines with **high** values of wood tones, sweetness, alcohol, and astringency.

Cluster 2 is characterized by wines with **medium** values of wood tones, sweetness, alcohol, and astringency.

Regarding the correlation of the variables, we didn't find any kind of correlation.

Perspective: Production Characteristics

Like the previous perspective, we started by applying the Elbow method, by analysing the Elbow graph, we considered that the point where the SSE stops decreasing a lot is when K = 2 (two clusters).



Graph 9 - Elbow graph for production characteristics perspective

These results are confirmed with the silhouette score values for various numbers of clusters, although these are very low.

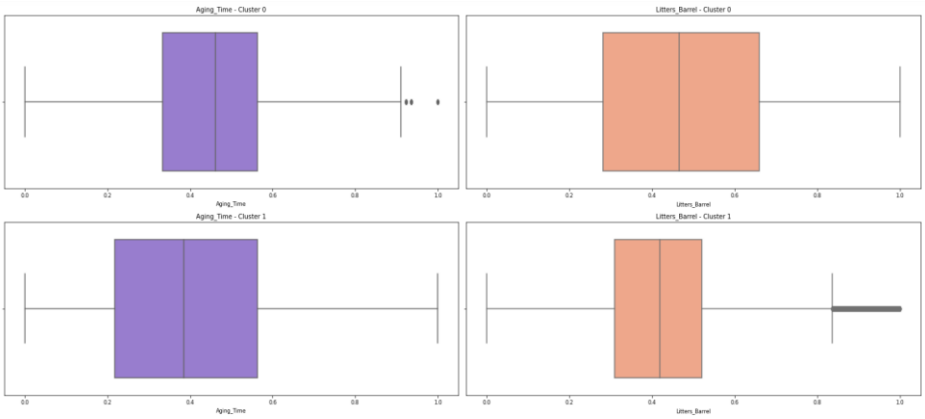
```
For n_clusters = 2 The average silhouette_score is : 0.3587239290490054
For n_clusters = 3 The average silhouette_score is : 0.3402840598753255
For n_clusters = 4 The average silhouette_score is : 0.31785554329774285
For n_clusters = 5 The average silhouette_score is : 0.32741571068724346
For n_clusters = 6 The average silhouette_score is : 0.35141639628089943
```

Table 17 - Silhouette scores for K-Means algorithm in production perspective

Our conclusion based on the Elbow Method and the silhouette score is that 2 clusters are the perfect fit for our data.

Next, we applied the algorithm with the correct number of clusters and started analysing the clusters using boxplot. We concluded that the customers were divided into 2 clusters based on the variables we identified earlier.

Looking at these variables, we were able to identify the clusters as follows:

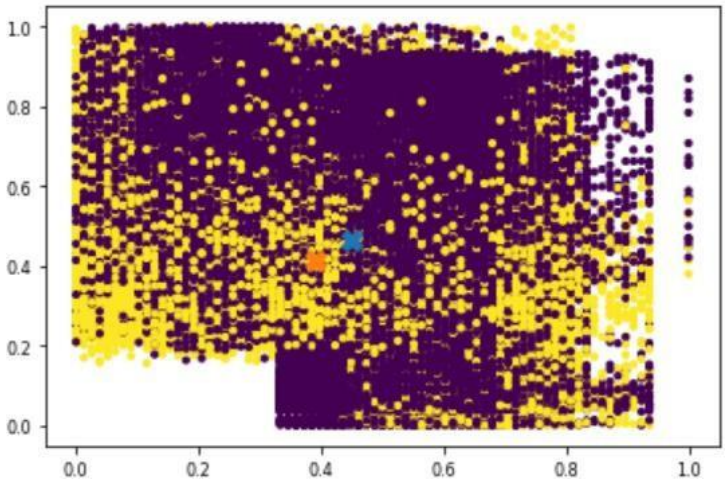


Graph 10 - Distribution of the cluster data in Aging_Time and Litter_Barrel variables

The **cluster 0** is characterized by wines with a medium aging time and a medium litters barrel.

The **cluster 1** is characterized by wines with a medium-low aging time and a medium-low litters barrel.

Regarding the correlations, it was very difficult to analyse the results, as can be seen in the graph below.



Graph 11 - Correlation between Aging_Time and Litters_Barrel variables

SOM

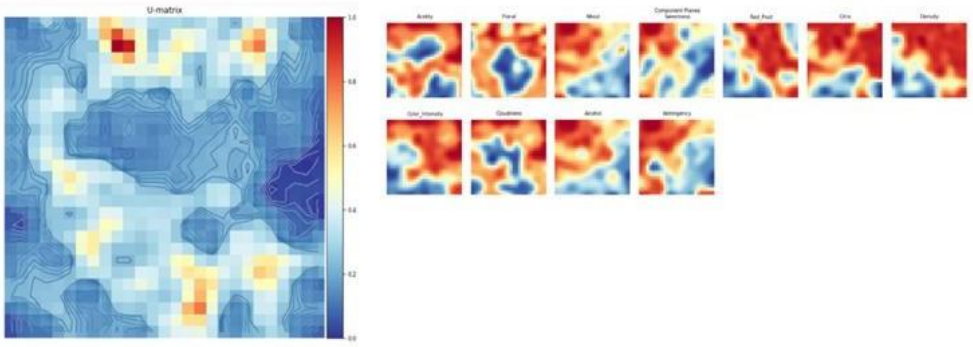
Perspective: Flavor/Feeling

First, we started by choosing the SOM grid size, for which we used a heuristic defined in the *som_make* function documentation. It states that the number of units in a map should be around 5*dlen^0.54321, where dlen is the number of records in the dataset. After some experimentation, we found that the best map size was around 27x27.

For the training part, we used train_rough_len=15 and train_finetune_len=25 as parameters, which represent the number of epochs by which the model will be trained with a high learning rate and a smaller one, respectively. These parameters were finetuned using a rule of thumb found in the documentation that stated: “For a topographic error very near to zero, the quantization error should be as low as possible”.

Next, we built the U-Matrix, using this map size, where we should be able to get an intuition on the possible number of clusters. We also computed the component

planes for each variable, where each unit represents a unit of the SOM grid and is colored according to the weight of each variable in the SOM.



Graph 12 - U-Matrix and Component planes

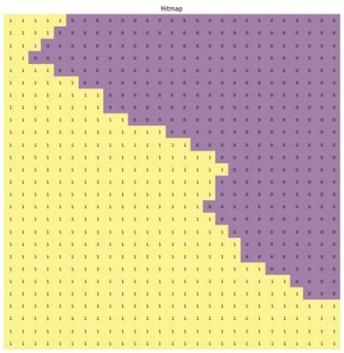
From these visualizations, we could get a sense of which variables were more related in terms of the density of higher/lower values.

After getting the BMUs (best matching units) of the SOM for each observation, we decided to use K-Means on top of our already trained data for clustering. With some exploration and the analysis of the silhouette scores for various numbers of clusters, we concluded that the best option was 2 clusters, although the score values are all too low.

```
For n_clusters = 2 The average silhouette_score is : 0.27248665367904334
For n_clusters = 3 The average silhouette_score is : 0.1809703845960016
For n_clusters = 4 The average silhouette_score is : 0.11511036462088076
For n_clusters = 5 The average silhouette_score is : 0.06250817114578017
For n_clusters = 6 The average silhouette_score is : -0.05575378023004999
```

Table 18 - Silhouette scores for SOM + K-Means algorithm in flavor/feeling perspective

Then, we produced a hitmap where we could see the match between each unit of the SOM grid and the correspondent cluster label.

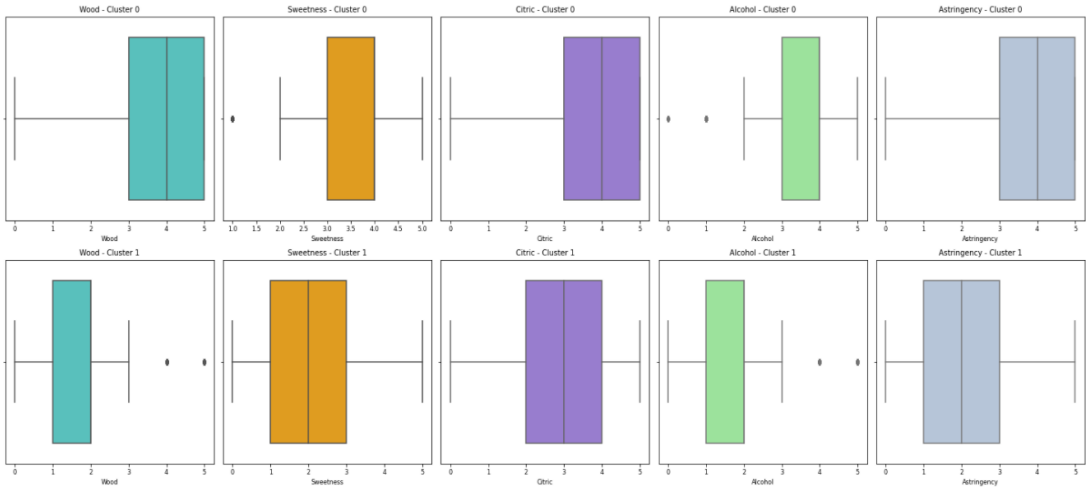


Graph 13 - Hitmap for two clusters in flavor/feeling perspective

Finally, by getting the cluster each observation belongs to using the BMUs, we were now ready to analyze the descriptive statistics of each cluster and produce some visualizations.

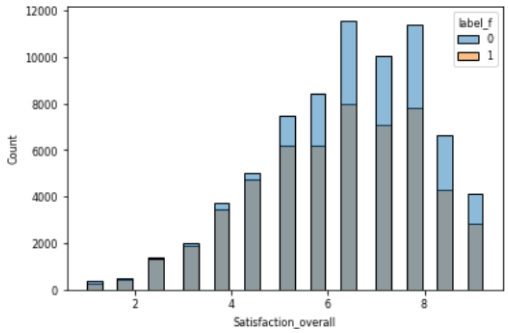
Cluster 0: Is composed by 54677 observations and is generally characterized by **high** values of wood tones, sweetness, citric tones, alcohol level and astringency.

Cluster 1: Is composed by 72619 observations and is generally characterized by **low** values of wood tones, sweetness, citric tones, alcohol level and astringency.



Graph 14 - Distribution of cluster data in Wood, Sweetness, Citric, Alcohol and Astringency variables

By analysing the clusters in the perspective of the *Satisfaction_overall* variable, we were also able to conclude that **cluster 0** has more wines with **higher** levels of average **satisfaction**.

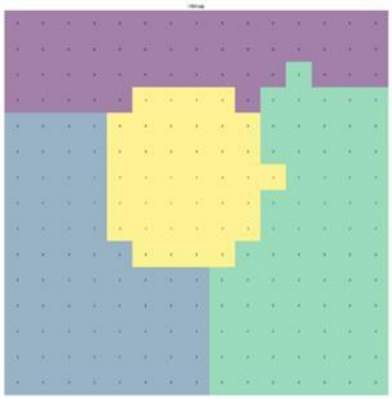


Graph 15 - Distribution of wine satisfaction in the two clusters (flavor/feeling perspective)

Perspective: Production Characteristics

For this perspective, we did the same steps, only this time the variables we used needed to be standardized, because they were not on the same scale and we had numerical variables mixed with categorical variables. The map size used for the SOM grid was also different, because we found that our data worked best with a smaller map this time.

The U-Matrix and silhouette scores indicated that the best number of clusters was 4, although the score was really low – approximately **0.11**.



Graph 16 - Hitmap for four clusters in production characteristics perspective

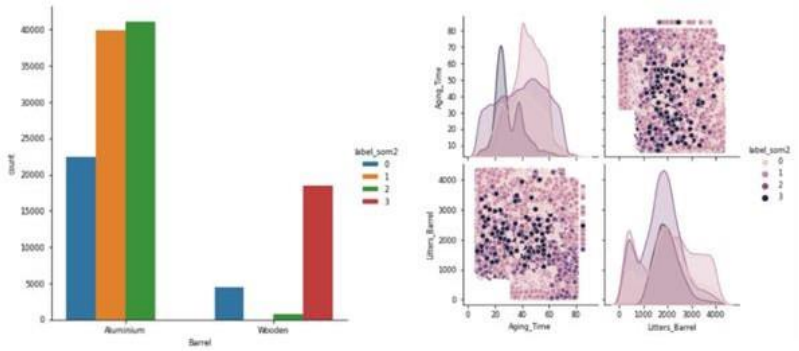
By plotting our data and exploring some visualizations we arrived at some conclusions about our clusters, although they were not very well defined.

Cluster 0: Has predominantly wines in aluminum barrels but has some in wooden barrels too; has the highest values of residual sugar.

Cluster 1: Has mostly wines with sulphites present in them; only has wines in aluminum barrels; all of its wines have mixed grapes.

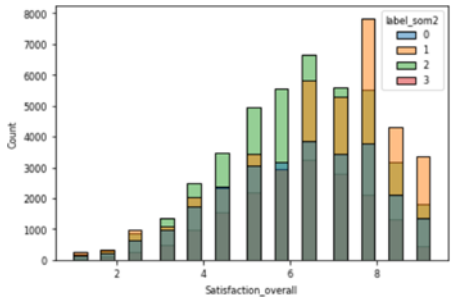
Cluster 2: Wines with big aging times are mostly the ones with low values of litters per barrel; most of its wines have no sulphites or very few; most wines are in aluminum barrels; more wines with single grapes than mixed

Cluster 3: Wines tend to have less aging time; only wines in wooden barrels; almost all wines have mixed grapes.



Graph 17 - Distribution of cluster data in Barrel variable and correlation between Aging_Time and Litters_Barrel for the four clusters

In terms of overall satisfaction, cluster 3 has the wines with the lowest scores, and clusters 1 and 2 have the highest.



Graph 18 - Distribution of wine satisfaction in the four clusters (production characteristics perspective)

BIRCH with agglomerative

Perspective: Flavor/Feeling

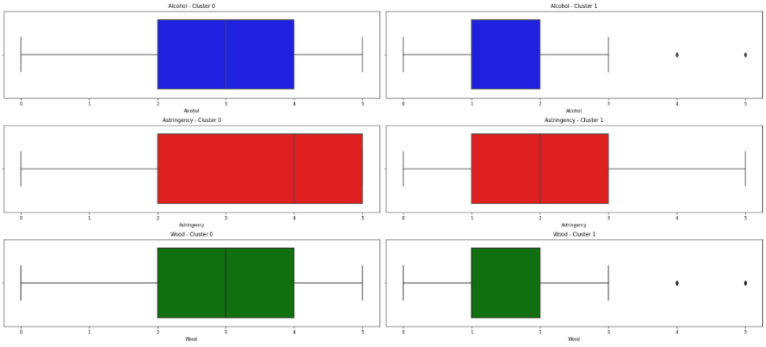
To apply this method, as mentioned before, we had to perform many times the algorithm to reach a good silhouette score. After some tries, our values for the parameters were: branching_factor=100 and threshold=.8. Additionally, we run the algorithm with these parameters for n_clusters = [2, 3, 4] where we obtained a

silhouette score of 0.143, 0.114, and 0.101, respectively. This indicates that the best number of clusters is 2.

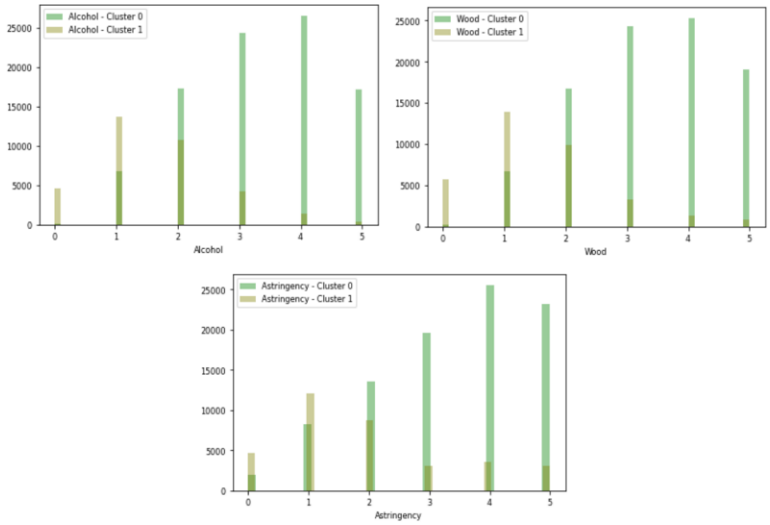
So, after plot two types of visualizations, boxplots, and distplots, although we had variables, we had similar clusters we were able to draw some conclusions with the other variables where their clusters were a bit more different.

Cluster 0: Opposite to cluster 0, the flavor of the wine is characterized by having high alcohol and astringency taste and high wood tones.

Cluster 1: The flavor of the wine is characterized by having low astringency and alcohol taste as well as low wood tones.



Graph 19 - Visualization of the most relevant clusters using boxplot



Graph 20 - Visualization of the most relevant clusters using distplot

Perspective: Production Characteristics

For the second perspective, we also try different values for the parameters for different numbers of clusters to find the best silhouette scores. So, for our

parameter, we choose the following: branching_factor=100 and threshold=.8. For our number of clusters, the best silhouette score was 4 with a score of 0.283.

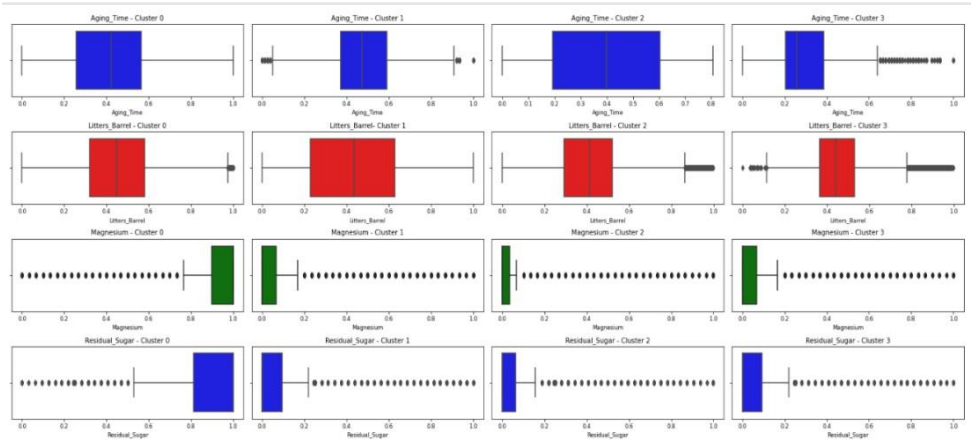
From both visualizations, we were able to draw some conclusions.

Cluster 0: Has a medium aging time and medium volume of litter in the barrel, with a high concentration of magnesium and residual sugar. It contains wines with both varieties of grapes and both types of wine (red and white).

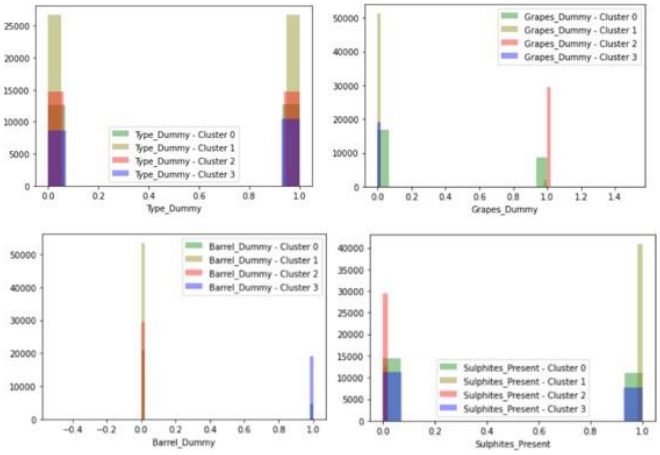
Cluster 1: Has a low concentration of magnesium and residual sugar, both types of wine, made with mixed grapes, aluminum barrels, and sulphites.

Cluster 2: Has also both types of wine, with low concentrations of magnesium and residual sugar. It is produced with single grapes, aluminum barrels and with no presence of sulphites.

Cluster 3: Has both types of wine, produced in wood barrels with mixed grapes. It presents low concentrations of magnesium and residual sugar, and like all clusters have medium litter barrels but lower aging time.



Graph 21 - Visualization of Aging_Time, Litters_Barrel, Magnesium and Residual_Sugar clusters with boxplots



Graph 22 - Visualization of the dummy variables with distplot

BIRCH with K-Means

Perspective: Flavor/Feeling

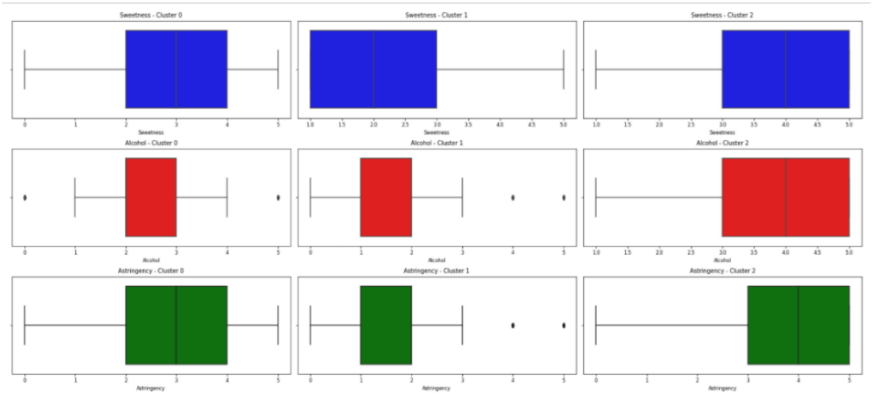
Like the BIRCH with Agglomerative algorithm, we also used the same parameters with the values that we considered to be the most accurate, branching_factor=100 and threshold=.8. The difference here was setting the number of clusters, we used the K-Means method. So, we also tried with two, three, and four clusters where we obtain a silhouette score of 0.183, 0.186, and 0.161, respectively. This indicates that the optimal number of clusters is 3.

So, to visualize the clusters we plot boxplots and distplots of the ones we considered more differently from each other and so be able to conclude something.

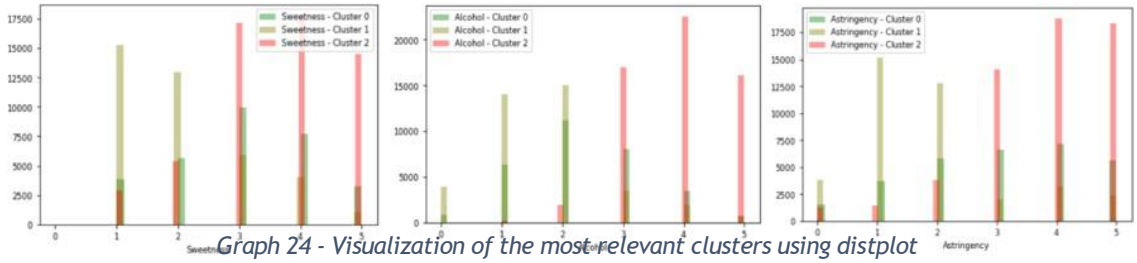
Cluster 0: Has higher levels of all three flavors, sweetness, alcohol and astringency.

Cluster 1: Has medium fallings of sweetness and medium levels of alcohol and astringency taste.

Cluster 2: Has a lower presence of sweetness and a lower presence of alcohol and astringency taste.



Graph 23 - Visualization of the most relevant clusters using boxplot



Graph 24 - Visualization of the most relevant clusters using distplot

Perspective: Production Characteristics

For this perspective, the optimal parameters were also branching_factor=100 and threshold=.8, using the K_Means algorithm to define the number of clusters. We run it for 2, 3, and 4 clusters to see which one had the highest silhouette score, resulting in 4 clusters with a silhouette score of 0.278.

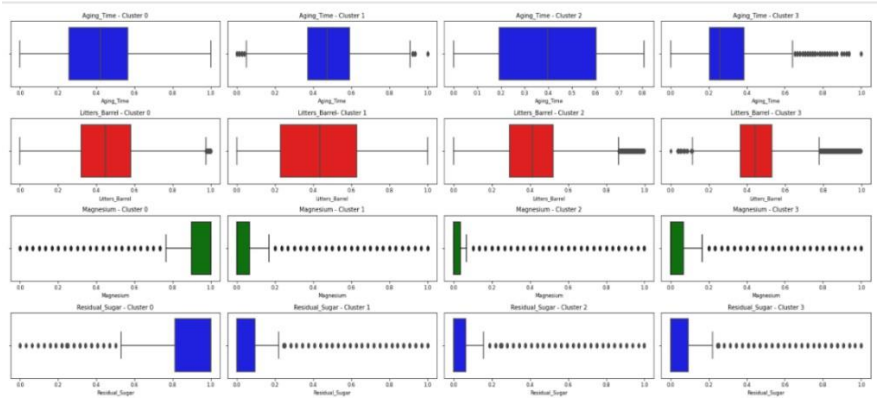
Again, by plotting the graphs we can visualize the characteristics of each cluster.

Cluster 0: Has high concentration of magnesium and residual sugar, produced with both types of barrels and grapes.

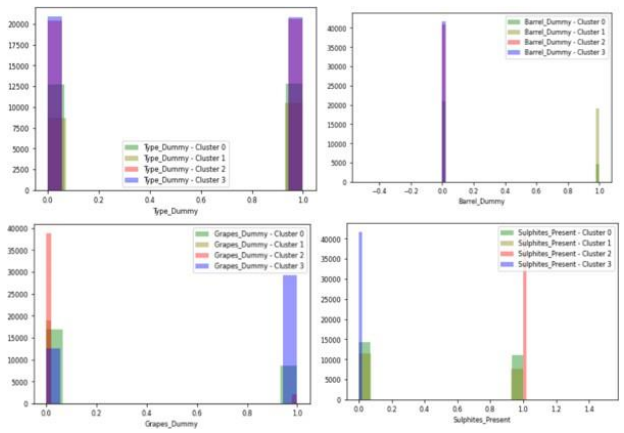
Cluster 1: Was produced predominately in aluminium barrels with mixed grapes and with the presence of sulphites.

Cluster 2: Has no presence of sulphites and a low concentration of magnesium and residual sugar. It was produced with single grapes in aluminium barrels.

Cluster 3: Has both types of wines, a medium aging time, and medium litters barrel, like all other clusters. It has low concentration of magnesium and residual sugar, produced with single grapes and in both types of barrels.



Graph 25 - Visualization of Aging_Time, Litters_Barrel, Magnesium and Residual_Sugar clusters with boxplots

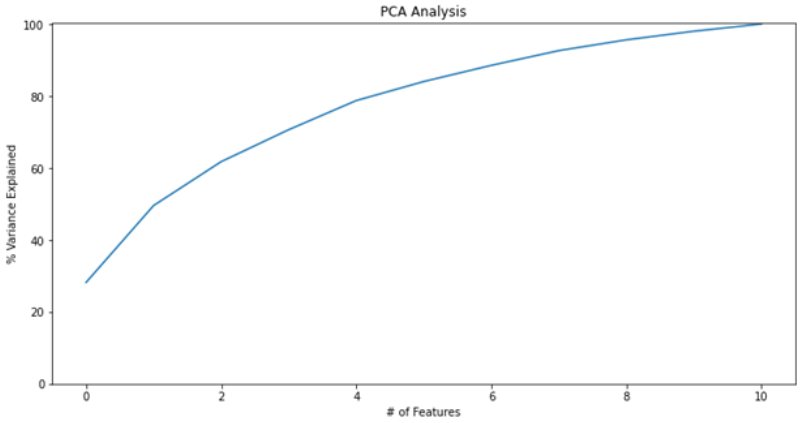


Graph 26 - Visualization of the dummy variables with distplot

DBSCAN

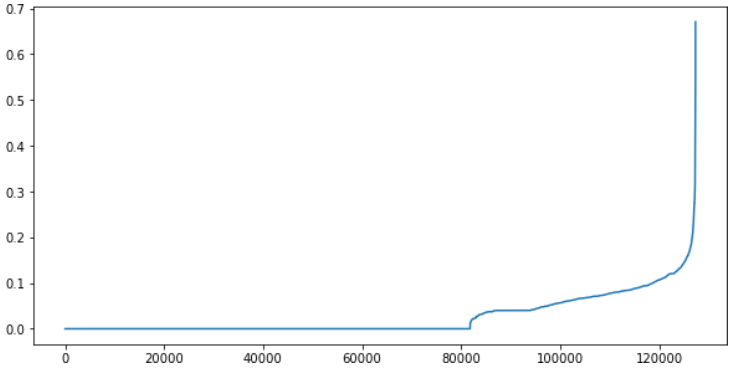
Perspective: Flavor/Feeling

To perform PCA, we standardized the data, and plotted the explained variance with different number of principal components. We chose the number of components (3), when it explained approximately 75% of the variance.



Graph 27 - Explained variance to choose the number of principal components (Flavor/feeling)

To find the best parameters, according to minpts = 2 * dimensions, minpts = 2 * 3 = 6 since we reduced the data. Regarding eps, we plotted the K-distance graph with k=minpts-1 = 5. The knee occurred approximately when eps =0.2, thus we tested several values of eps ≥ 0.2, maintaining minpts = 6. We analyzed the Silhouette score of each combination and the best eps would be 0.4 with a Silhouette score of 0.23. When eps=0.45 the score was higher, however it only formed one cluster. Consequently, a DBSCAN instance was defined with eps=0.4 and minpts=6.



Graph 28 - K-distance plot to find the best eps value (Flavor/feeling)

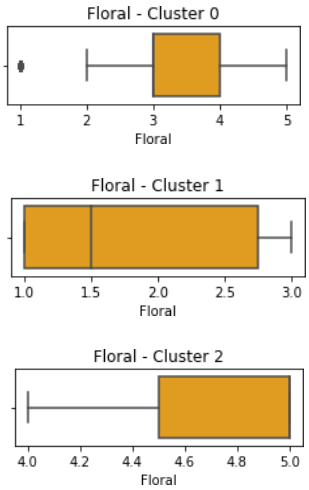
Minpts	Eps	Number of clusters	Silhouette score
6	0.2	187	-0.58159
6	0.25	67	-0.42844
6	0.3	19	-0.13558
6	0.35	8	-0.04385
6	0.4	3	0.22925
6	0.45	1	0.32909

Table 19 - Silhouette score results for different combinations of parameters (flavor/feeling)

It returned 3 clusters, assigning however the great majority of the points to cluster 0.

Variable	Cluster	0	1	2
	Count	127206	6	7
Acidity	Mean	3.49	1.33	4.86
Floral		3.34	1.83	4.71
Wood		2.85	0	0
Sweetness		2.99	1.33	1.14
Red_Fruit		3.24	3.50	1.86
Citric		3.38	1.17	0.14
Density		3.52	2.17	2.14
Color_Intensity		3.47	2.17	1.86
Cloudiness		3.47	1.17	5
Alcohol		2.84	0.17	0.14
Astringency		2.99	0	0

Table 20 Flavor / feeling cluster description

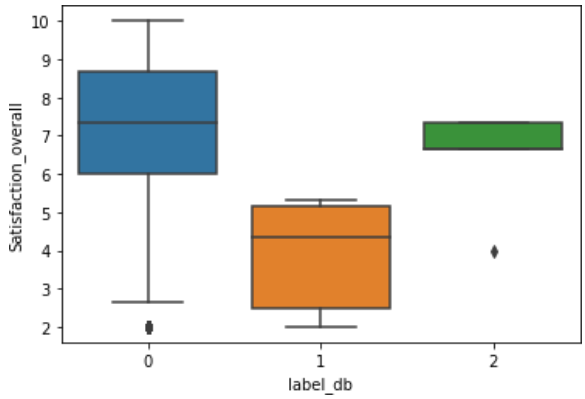


Graph 29 - Visualization of Floral flavor in the different clusters (flavor/feeling)

Cluster 0 comprises wines that have average ratings of about 3 in all the flavors. However, there are higher values of ratings in several flavors, like Sweetness, Citric, Density, Color_Intensity or Alcohol.

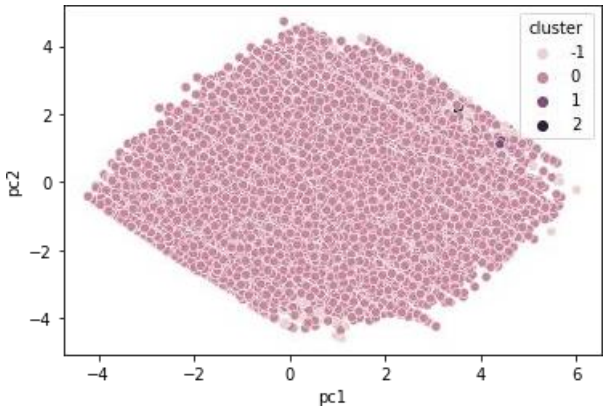
Cluster 1 has very few wines assigned, nevertheless it has the highest value of Cloudiness and Acidity. Also, we can highlight the low values in Alcohol and Citric flavors.

Cluster 2 also has few wines assigned, still we can note that the ratings are rather low in all flavors, except in Acidity and Floral that present the highest values. Regarding the overall satisfaction, cluster 0 comprises wines with higher satisfaction, followed by cluster 2, whereas cluster 1 has the lowest level of satisfaction in the three countries.



Graph 30 - Visualization of Overall Satisfaction in the different clusters (flavor/feeling)

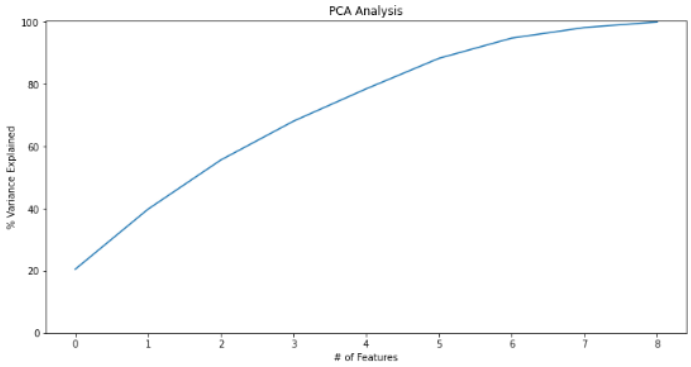
Visualizing the results in 2 dimensions, we can verify that most of the data points belong to cluster 0, being extremely difficult to point out the remaining clusters.



Graph 31 - DBSCAN results in two dimensions (flavor / feeling)

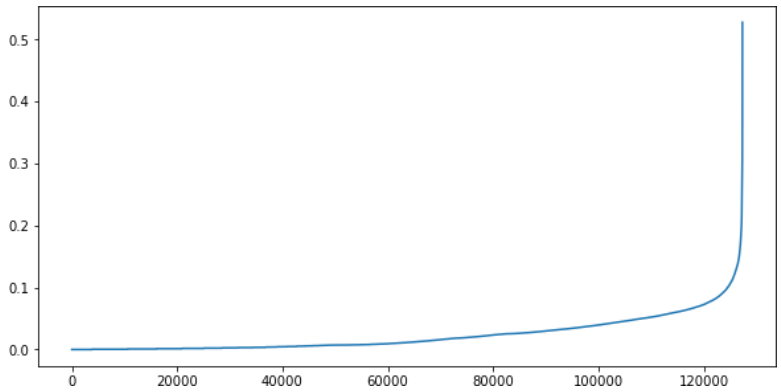
Perspective: Production Characteristics

Firstly, the data was first scaled through Min-Max scaler to have a scale between 0 and 1 in all variables. To perform PCA, we standardized the data and plotted the explained variance, concluding that having 3 principal components accounted for approximately 75% of the variance.



Graph 32 - Explained variance to choose the number of principal components(Production characteristics)

We used the same heuristic for minpts, resulting in minpts = 6. For the eps, we plotted the K-distance graph with k=minpts-1 = 5. The knee seems to be when eps =0.2, hence we tested several values of eps ≥ 0.2 and evaluated their Silhouette scores. The best combination was minpts = 6 and eps=0.35, registering a score of 0.154.



Graph 33 - K-distance plot to find the best eps value (Production characteristics)

Minpts	Eps	Number of clusters	Silhouette score
6	0.2	22	-0.10404
6	0.25	3	0.11871
6	0.3	2	0.14679
6	0.35	2	0.15350
6	0.4	1	0.26686
6	0.45	1	0.27147

Table 21 - Silhouette score results for different combinations of parameters (Production characteristics)

A DBSCAN instance with those parameters was defined, providing 2 clusters. Once again, most data points were assigned to cluster 0.

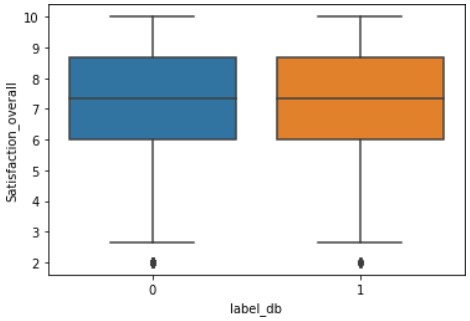
Variable	Cluster	0	1
	Count	122783	4490
Aging_Time	Mean	39.68	38.62
Litters_Barrel		1926.98	1781.77
Type_Dummy		0.51	0.51
Magnesium		7.31	7.47
Residual_Sugar		7.80	7.91
Sulphites_Present		0.49	0
Sulphites_Very Few		0.04	1
Barrel_Dummy		0.19	0
Grapes_Dummy		0.29	1

Table 22 - Production characteristics clusters description

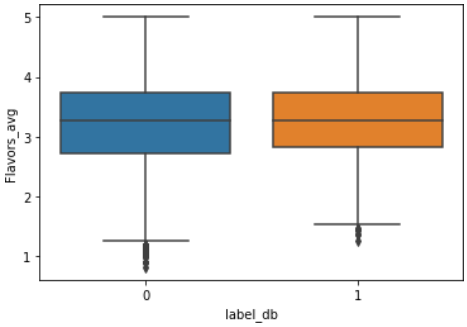
Cluster 0 comprises wines of both types (white and red) and with approximately 3 years of aging time. Also, these wines' barrels are mainly of type aluminium.

Cluster 1 can be associated to wines where the grapes variety is mixed, and there are few sulphites present.

Additionally, both clusters behave similarly regarding not only the overall satisfaction, but also the flavors average rating.

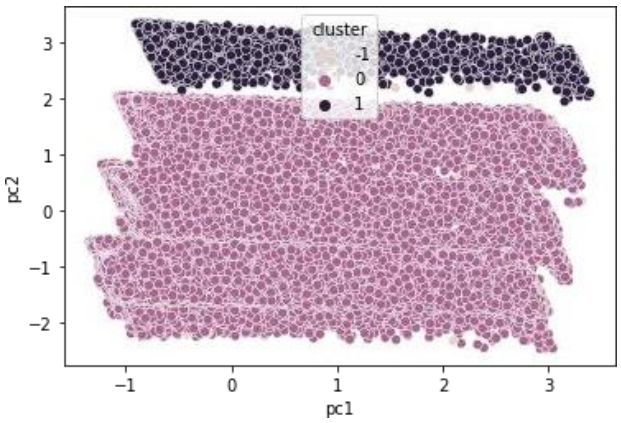


Graph 34 - Visualization of Overall Satisfaction in the different clusters (Production characteristics)



Graph 35 - Visualization of Flavors Average in the different clusters (Production characteristics)

Plotting the results in two dimensions, the majority belongs to cluster 0 and there is a well separated cluster at the top, cluster 1.



Graph 36 - DBSCAN results in two dimensions (Production characteristics)

DISCUSSION

After exploring multiple algorithms for the two perspectives – flavor/feeling and production characteristics - and analysing the results given by the silhouette scores, it was now time to choose the best algorithm for each perspective, in order to join the clusters later.

For the flavor/feeling perspective, the algorithm that produced the best results was **SOM + K-Means** with **2 clusters**, although the silhouette score wasn't very high in general – approximately **0.27**.

Regarding the production characteristics perspective, the best results were given by **K-Means,** also with 2 clusters and a silhouette score of approximately **0.36**.

To join the perspectives and get the final clusters, we added the label columns of SOM and K-Means to the original dataset and computed the descriptive statistics for the combination of the two, ending up with **4 final wine segments** with distinct characteristics.

After carefully analyzing the mean and count of all the variables, we gathered the main characteristics of each segment. It is important to note that some variables weren't much relevant for the definition of the clusters, so they are not included in the characteristics.

label_f	label_p	Aluminium	Wooden	Mixed	Single	Red	White
0	0	28976	5547	32909	1614	17103	17420
	1	31249	6931	16130	22050	17367	20813
1	0	21568	3646	24165	1049	12548	12666
	1	21780	7599	14026	15353	15629	13750

Table 23 - Categorical variables of the final clusters

	label_f	0		1	
	label_p	0	1	0	1
Aging_Time	mean	41.9895	37.775	42.3361	36.9694
	max	85	85	85	85
	min	7	7	7	7
Litters_Barrel	mean	2051.4	1780.82	2040.16	1852.37
	max	4320	4321	4321	4321
	min	50	50	50	50
Magnesium	mean	7.09565	7.22381	7.19255	7.80357
	max	30	30	30	30
	min	0	0	0	0
Residual_Sugar	mean	7.41759	7.7893	7.53366	8.5
	max	32	32	32	32
	min	0	0	0	0
Sulphites_Present	mean	1	0	1	0
	max	1	0	1	0
	min	1	0	1	0
Sulphites_Very Few	mean	0	0.13934	0	0.13843
	max	0	1	0	1
	min	0	0	0	0
Acidity	mean	3.73302	3.3973	3.60938	3.22305
	max	5	5	5	5
	min	0	0	0	0
Floral	mean	3.54679	3.23465	3.51709	3.08812
	max	5	5	5	5
	min	1	1	1	1
Wood	mean	3.88164	3.72116	1.61295	1.5651
	max	5	5	5	5
	min	0	0	0	0
Sweetness	mean	3.7838	3.42881	1.88943	2.43112
	max	5	5	5	5
	min	1	1	0	0
Red_Fruit	mean	3.33638	3.23499	3.30626	3.08765
	max	5	5	5	5
	min	0	0	0	0
Citric	mean	3.93236	3.6406	3.46696	2.31369
	max	5	5	5	5
	min	0	0	0	0
Density	mean	3.81366	3.39015	3.72571	3.15885
	max	5	5	5	5
	min	1	1	1	1
Color_Intensity	mean	3.69108	3.37459	3.60807	3.21856
	max	5	5	5	5
	min	0	0	0	0
Cloudiness	mean	3.74203	3.29539	3.66757	3.20484
	max	5	5	5	5
	min	1	1	1	1
Alcohol	mean	3.69858	3.61991	1.54973	1.92736
	max	5	5	5	5
	min	0	0	0	0
Astringency	mean	3.79376	3.6703	1.66828	2.30045
	max	5	5	5	5
	min	0	0	0	0

Table 24 - Numerical variables of the final clusters

Segment/Catalogue 1:

This catalogue was specially created to highlight sweet and strong wines from our finest mixed grapes, both whites and reds. Produced in aluminium barrels, we have a presence of sulphites in the wine, they are also dense with an average concentration of 7.1 g per 1000 litters of wine for both magnesium and residual sugar. With an average aging time of 3 years and 6 months, you can taste a mix of citric and red fruits along with a certain acidity. Finally, you can also see floral and wood tones present in the wines of this catalogue.

Segment/Catalogue 2:

This next catalogue was specially created to highlight dense and strong wines from both our finest single and mixed grapes, where you will find a higher percentage of

white wines available. Produced in aluminium barrels, these wines have no presence of sulphites, with an average concentration of 7.5 g per 1000 litters of wine for both magnesium and residual sugar. With an average aging time of 3 years and 2 months in barrels that contain on average 1780 litters, you can taste a mixed of citric and red fruits along with a certain acidity. Finally, you can also see some cloudiness and astringency in the wines of this catalogue.

Segment/Catalogue 3:

This third catalogue was specially created to highlight acid and light wines from our finest mixed grapes, with both whites and reds. Produced in aluminium barrels, these wines have a presence of sulphites, with an average concentration of 7.2 g per 1000 litters of wine for both magnesium and residual sugar. With an average aging time of 3 years and 7 months, you can taste a mix of citric and red fruits. However, in this catalogue you have no sweetness feeling and you can barely see the presence of wood tones just floral. Finally, you can also see some cloudiness but no astringency.

Segment/Catalogue 4:

This last catalogue was specially created to highlight floral and light wines from both our finest single and mixed grapes, where you will find a higher percentage of red wines available. Produced in aluminium barrels, these wines have no presence of sulphites, with an average concentration of 8.2 g per 1000 litters of wine for both magnesium and residual sugar. With an average aging time of 3 years and 1 month, you can taste red fruits with a little acidity. However, in this catalogue you have no sweetness feeling and you can barely see the presence of wood tones nor citric flavors. Finally, you can also see some cloudiness but just a few astringencies in these wines.

Image 1 - Example of a catalog page for wine segment 0

Once a month, we want to advertise on the cover our finest wine that perfectly fits all attributes present in each catalogue created. Additionally, there will also be promotions in specific wines based on our top clients and sales (for future research).

CONCLUSION

By developing this project, we were able to put in practice many techniques learned in class to pre-process, transform, and cluster the data, as well as derive insights from and evaluate the results.

In the end of our segmentation, we were able to define four wine segments, with distinct characteristics, and provide a marketing strategy (wine catalogs) that can then be sent to the marketing department for them to take advantage of. Some future work that could be done in this area would be to segment customer data, to know how to better target our catalogs.

Overall, although our initial dataset did not have many variables and seemed relatively simple, we were able to make sense of it and derive actionable insights, helping the company's marketing strategy and, ultimately, its revenue, by better targeting the main product – wines.

REFERENCES

Birkett, A. (2019, August 24). *How to Deal with Outliers in Your Data*. CXL. https://cxl.com/blog/outliers/

Frost, J. (2019, October 23). *Guidelines for Removing and Handling Outliers in Data*. Statistics By Jim. https://statisticsbyjim.com/basics/remove-outliers/

Horsch, A. (2020, November 22). *Detecting And Treating Outliers In Python — Part 1*. Towards Data Science. https://towardsdatascience.com/detecting-and-treating-outliers-in-python-part-1-4ece5098b755

Amira, A. (2021, August 31). *Pandas Get Dummies (One-Hot Encoding) – pd.get_dummies()*. Amiradata. https://amiradata.com/pandas-get-dummies/

Band, A. (2020, May 23). *How to find the optimal value of K in KNN?* Towards Data Science. https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb

pandas documentation — pandas 1.3.5 documentation. (2021, December 12). Pandas. https://pandas.pydata.org/docs/

Version 1.3.5

scikit-learn - Machine Learning in Python. (n.d.). Scikit-Learn. https://scikit-learn.org/

scikit-learn developers. (n.d.). *1.6. Nearest Neighbors*. Scikit-Learn 1.0.2. https://scikit-learn.org/stable/modules/neighbors.html#classification

som_make. (n.d.). SOM Toolbox. http://www.cis.hut.fi/projects/somtoolbox/package/docs2/som_make.html

Sander, J., Ester, M., Kriegel, H.-P. & Xu, X. (1998). Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. Data Mining and Knowledge Discovery, 2, 169--194.

Schubert, E., Sander, J., Ester, M., Kriegel, H., & Xu, X. (2017). DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. https://www.ccs.neu.edu/home/vip/teach/DMcourse/2_cluster_EM_mixt/notes_slides/revisitofrevisitDBSCAN.pdf

Mullin, T. (2020, July 15). DBSCAN parameter estimation using Python. Medium. https://medium.com/@tarammullin/dbscan-parameter-estimation-ff8330e3a3bd

Mysiak, K. (2020, July 16). Explaining DBSCAN clustering. Medium. https://towardsdatascience.com/explaining-dbscan-clustering-18eaf5c83b31

Implementing DBSCAN algorithm using Sklearn. GeeksforGeeks. (2019, June 6). https://www.geeksforgeeks.org/implementing-dbscan-algorithm-using-sklearn/https://www.geeksforgeeks.org/implementing-dbscan-algorithm-using-sklearn/

Dobilas, S. (2021, August 22). DBSCAN clustering algorithm-how to build powerful density-based models. Medium. https://towardsdatascience.com/dbscan-clustering-algorithm-how-to-build-powerful-density-based-models-21d9961c4cechttps://towardsdatascience.com/dbscan-clustering-algorithm-how-to-build-powerful-density-based-models-21d9961c4cec

DBSCAN algorithm: Understand the DBSCAN clustering algorithm. Analytics Vidhya. (2021, June 8). https://www.analyticsvidhya.com/blog/2021/06/understand-the-dbscan-clustering-algorithm/

Jaadi, Z. (n.d.). A step-by-step explanation of principal component analysis (PCA). Built In. https://builtin.com/data-science/step-step-explanation-principal-component-analysis

Principal component analysis. Principal Component Analysis (2017, May 1). https://www.geo.fu-berlin.de/en/v/soga/Geodata-analysis/Principal-Component-Analysis/index.html

Jolliffe, I. T. (2002). Principal component analysis. 2nd ed. Springer-Verlag.

*SKLEARN.DECOMPOSITION.PCA.scikit.*https://scikitlearn.org/stable/modules/generated/sklearn.decomposition.PCA.html