Analysis of cocktail recipes

Thomas Bordes, Damien Willett

2020-05-06

1 Introduction

The purpose of this report is to give an account of the work we have carried out as part of the SY09 UV project. One of the aims of the project was to apply the methods we studied during the semester to real datasets.

The datasets we are dealing with here come from the Tidy Tuesday GitHub, which was put online in 2020 (2020-05-26 Cocktails). There are two of them. The first dataset consists of cocktail recipes that were retrieved from the Web as part of a hackathon. It includes cocktail ingredients and the type of glass in which the cocktails are served. The second is a list of cocktail recipes taken from a popular American bartender's guide called *Mr Boston Bartender's Guide*.

We will first clean the two datasets, perform an exploratory analysis, and then use unsupervised learning techniques to see what patterns we can find to classify the cocktails according to their ingredients. Finally, we will apply supervised classification methods, the aim of which will be to learn a model that can be used to determine the class of a cocktail.

2 Exploratory analysis and data cleaning

Let's start with the first dataset `cocktails`.

Before processing, it contains the recipes for 2104 cocktails made up of 333 different ingredients. We can study the qualitative variables *category* and *glass* (figure 1). The most frequent modality for the category is undoubtedly *ordinary drink*. Similarly, only a few glasses are used for the vast majority of cocktail recipes. This dataset also gives us an indication of the type of cocktails, with or without alcohol, thanks to the modalities taken by the qualitative variable *alcoholic*. Most cocktails contain between 2 and 5 ingredients. To go a step further, we can determine that the most important common ingredients between cocktails with and without alcohol are sugar, lemon juice and orange juice.

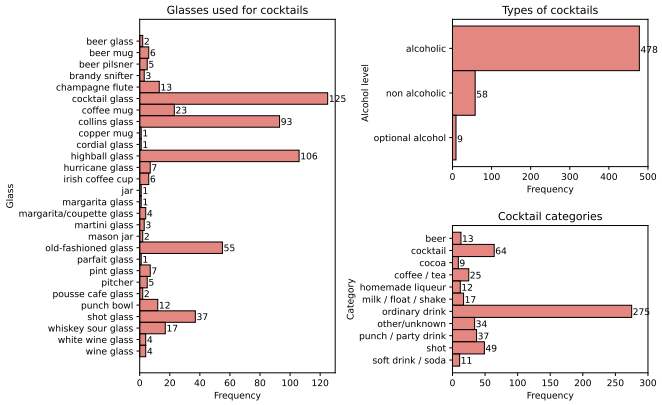

FIGURE 1 – Dataset `cocktails`

Now let's look at the second dataset `boston_cocktails`.

Each record contains the name of the cocktail (in the form of a character string), its category (character string), then each ingredient (character string) it contains and the quantity required (character string made up of a number and a unit).

We note that some categories have very small numbers, just a few cocktails (figure 2). In our cleaning process, we need to remove the cocktails belonging to these categories from the data.

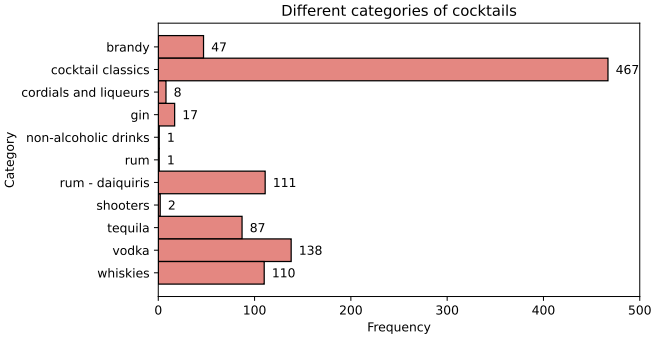

FIGURE 2 – Cocktail categories

In total, there are 989 drinks in this dataset containing 554 unique ingredients. Each of the cocktails is therefore made up of several ingredients, the distribution of which is given by the histogram in Figure 3.

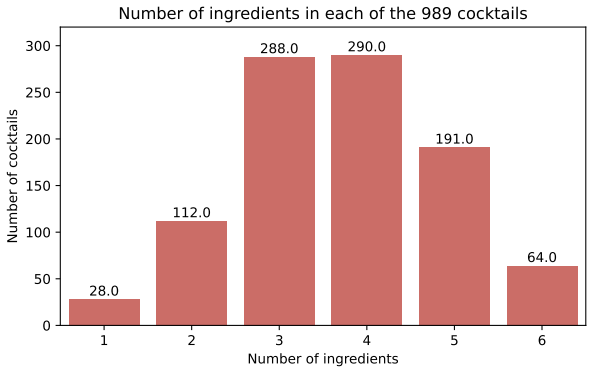

FIGURE 3 – Number of ingredients per cocktail

Pre-processing of the *measure* column is necessary. The cleaning strategy implemented essentially involves separating the numerical value of the measurement from the unit. The measurements are expressed in fractional writing ; we convert them to decimal writing. In addition, to make the latter values commensurable, we decided to keep only those expressed in *oz* (ounces), which represent the majority of measurements.

Furthermore, when we look at the list of ingredients, we notice that some ingredients are sometimes found under different names. For example, we can find the terms *fresh lemon juice* or *lemon juice* or *juice of a lemon*, which in themselves designate the same ingredient. We need to clean up the data accordingly and modify the *ingredient* column. After grouping, we are left with 263 unique ingredients. We then decided to keep only the most frequent ingredients, in this case those that appear in the recipes of at least 15 cocktails.

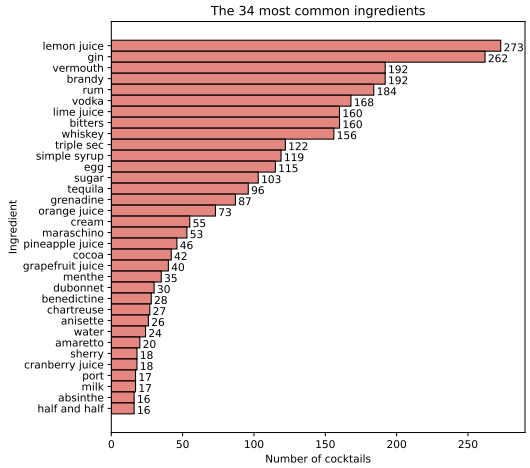

FIGURE 4 – Most common ingredients

As the histogram illustrates (figure 4), we now count 34 different ingredients in our dataset that appear at least 15 times in the recipes of 963 cocktails. We note that *lemon juice* tops the list of most frequent ingredients, while the most common alcoholic ingredient is *gin*, followed by *vermouth*, *brandy*, *rum* and *vodka*.

Let's look at the correlation between these 34 ingredients. The correlation matrix (*heatmap*) is shown in figure 5. It allows us to visually observe the ingredients often associated with cocktail recipes. We can see that the explanatory variables are very poorly correlated overall.

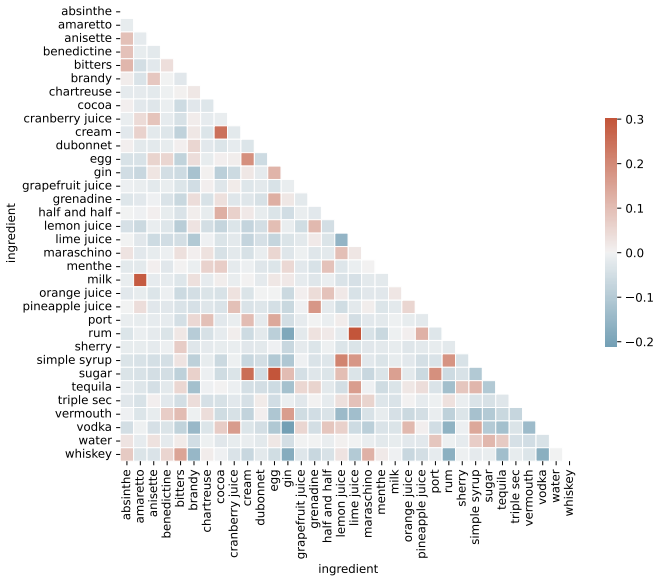

FIGURE 5 – Correlation between explanatory variables

For the modelling, we need the data in a large format. We therefore use the Python function `pivot_table`. The idea is to obtain an individual-variable table in which each of the 963 individuals represents a cocktail and where the 34 quantitative variables are the 34 ingredients found previously.

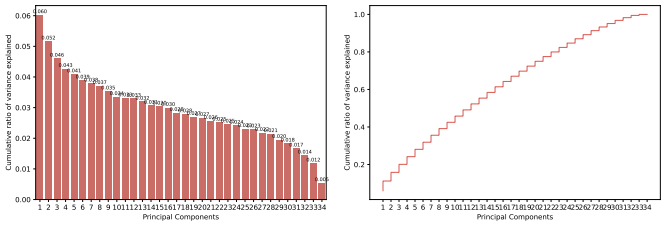

FIGURE 6 – Main axes of inertia

Note As we have just seen, cleaning work was carried out on the `cocktails` dataset as well as applied to the `boston_cocktails` dataset. The first dataset – due in particular to its extraction technique (*web scraping*) – presents a certain amount of folklore in the data. After the cleaning process, half as many individuals (cocktails) remained. The number of individuals in certain categories or certain types of cocktails was therefore very low. This is why in this report we have chosen to focus on the `boston_cocktails` dataset, which is a little more substantial.

3 Unsupervised methods

3.1 Principal component analysis (PCA)

With the dataset `boston_cocktails`, we now want to visualise cocktails according to the different ingredients they contain.

We apply a PCA to the individual-variable table. We obtain the principal axes represented in figure 6. The first axis carries 6.03% of the total inertia, the second 5.17% and the third 4.61%. The respective eigenvalues (variance explained by each principal component) are 2.05%, 1.76% and 1.57%. Note that the first 16 eigenvalues, which are greater than 1, together explain 61.42% of the total inertia. To exceed the threshold of 70% of the total inertia explained, we need to retain the first 20 principal components (72.43%). We can therefore see that a limited number of principal axes will not provide a good approximation of the variation present in the original dataset of dimension 34, and it will be difficult to visualise them all at the same time ; it will be necessary to visualise them in pairs.

3.1.1 Variable representation

To interpret the axes, let's look at the correlation of the variables with the principal axes and calculate the correlation matrix. The first two principal components are used to obtain the representation shown in figure 7. The aim is to understand which ingredients in the cocktail contribute most in the positive and negative directions of these axes.

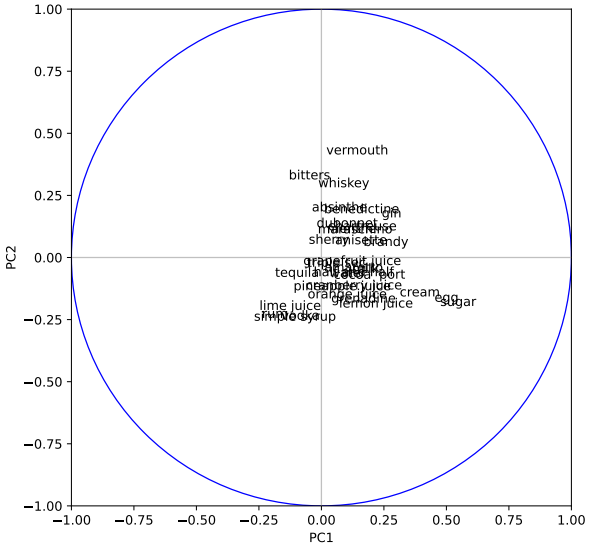

FIGURE 7 – Correlation circle

With regard to the quality of representation, it should be noted that certain variables such as *vermouth*, *bitters*, *sugar* or *simple syrup* are better represented than others in this first factorial design. However, none is perfectly well represented because none is really close to the circle of correlations. Still using the correlation matrix, we can plot the *barplot* in figure 8.

Note that the first axis mainly informs us about the presence of *suggar*, *egg* or *cream* on the one hand, and *simple syrup*, *lime juice* and *rum* on the other. The second axis differentiates cocktails based on *vermouth*, *bit-*

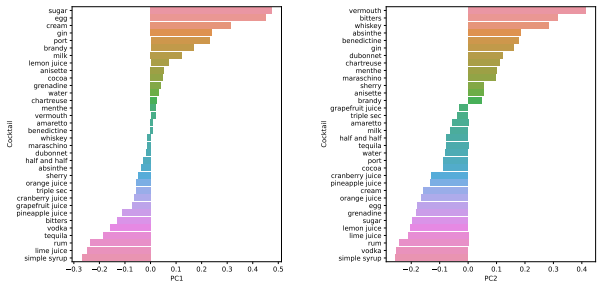

FIGURE 8 – Correlations of variables on the first (left) and second (right) principal axes

ters and *whiskey* on the one hand, and based on *simple syrup* and *vodka* on the other. In practical terms, cocktail recipes are not likely to contain *gin* or *whiskey* and *rum* at the same time (which seems rather logical).

3.1.2 Representation of individuals

We can represent the data in the first factorial plane (figure 9). Cocktails containing *vermouth* are at the top, those containing *rum* on the left and those containing *sugar* on the right.

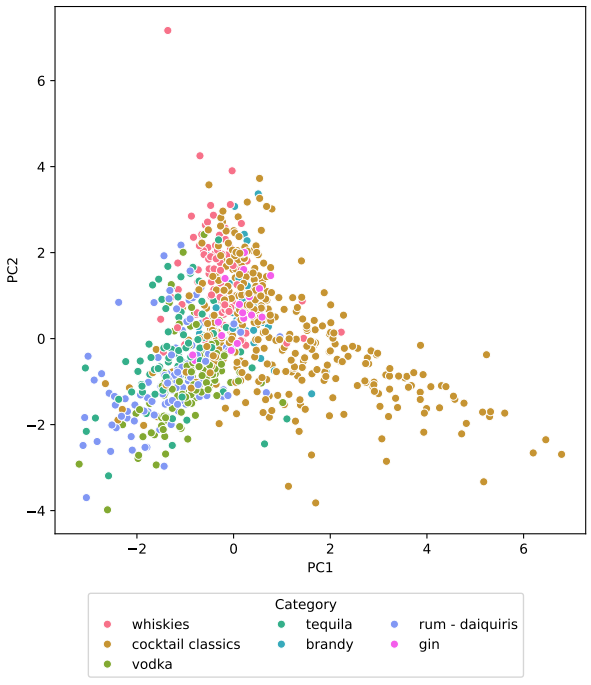

FIGURE 9 – Representation of cocktails in the first factorial plane

Finally, in the light of this analysis, we can see that, although the first factorial design does not allow us to clearly identify the original categories, it does allow us to visually distinguish the cocktails classified in the *cocktail classics* category from the others. It should also be noted that the *cocktail classics* category, which has the largest number of members, nevertheless seems to include many drinks that could possibly be placed in other categories.

With this in mind, we created a new individual-variable table in which the 963 cocktails still represent the 963 individuals, but where the explanatory variables are now the first 2 principal components since we are more interested in the first factorial design. We can also establish two categories of cocktails : *cocktail classics* and *other* (encompassing the 6 other categories).

3.2 K-means method

The objective here is to find clusters from our continuous quantitative variables (main axes of inertia). The method of moving centres (K-means) is suitably associated with our table of individuals-variables with weights $\frac{1}{n}$ (with $n = 963$) and the Euclidean distance as a measure of dissimilarity.

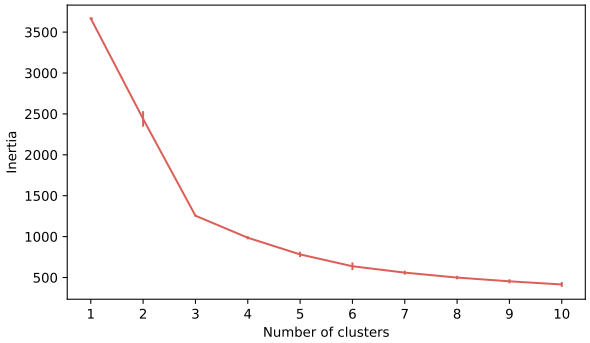

FIGURE 10 – Elbow method

First, we want to visualise the initialisation inertias *randomom* as a function of the number of clusters and select the most likely number of clusters ». According to the elbow rule (figure 10), a good number of clusters would be 3 clusters. We know the dataset and we know that cocktails are classified according to 7 categories (2 in our new classification).

Using the K-means method, we obtain the graph shown in figure 11 (left) for $K = 3$ with an initialisation method *random*. The K-means algorithm confirms our observations following the PCA that the cocktails are classified according to 3 directions in the first factorial plane. We also test the adaptive K-means method

using the Mahalanobis distance instead of the Euclidean distance. The graph for $K = 3$ with an initialization method *random* is given in figure 11 (right).

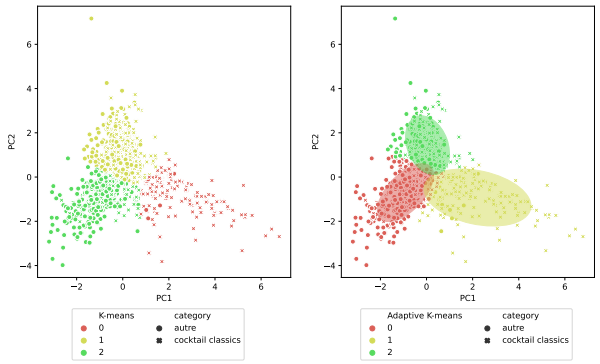

FIGURE 11 – K-means (left) and adaptive K-means (right) ($K = 3$, initialization *random*)

To measure the adequacy of the partition found with the real partition, i.e. to compare the partitioning obtained by the automatic classification with the partitioning corresponding to the categories defined at the outset, we calculate the adjusted Rand index. The values are recorded in table 1. We also see that consistency is stronger when the number of groups requested from the K-means algorithm corresponds to the number of natural « categories » in the data.

TABLE 1 – Indices de Rand ajustés

# groups	2	3
K-means	0.1787	0.1406
Adaptive K-means	0.2549	0.1478

4 Supervised methods

In this section, we use the first factorial design in which each point represents a cocktail, characterised by its coordinates on the first and second factorial axes respectively. We keep the indication of the 2 classes corresponding to the 2 categories of cocktails (*cocktail classics* and *other*).

4.1 K nearest neighbours method

Model selection .

To determine the optimal number of neighbours K, we first adopt the so-called cross-validation strategy. We randomly separate the available data set to form a training set and a test set. The results can be seen in figure 12. The optimal number K is 67.

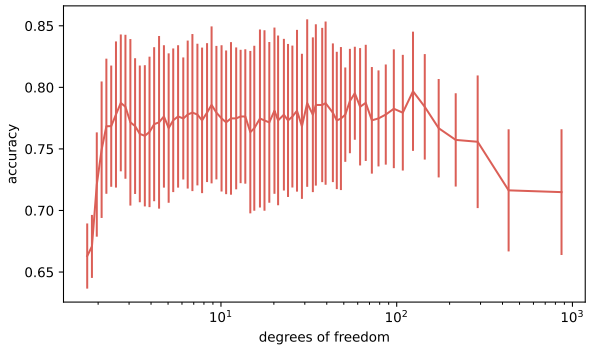

FIGURE 12 – Determining the optimal number K of neighbours by cross-validation

We therefore use the 67 nearest neighbours method on the previous data. We can visualise the decision frontier in figure 13.

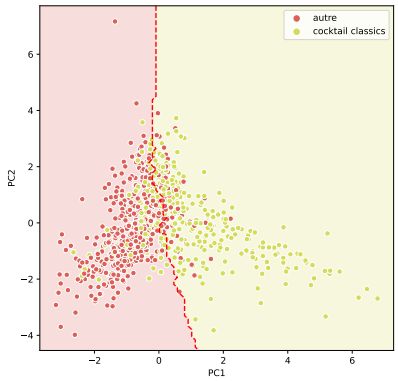

FIGURE 13 – Decision frontier obtained using the 67 nearest neighbours method

Estimating performance Let us now give an unbiased estimate of the model's accuracy. The *accuracy* is 78.65%. In other words, more than 78% of the predictions on the test set are correct ; the model's performance is good.

4.2 Discriminant analysis

We are interested in being able to create a classifier which, based on the cocktail category, can define decision boundaries. For example, we can use a binary classification to determine whether a cocktail is of the

cocktails classic type (coding « 1 ») or not (coding « -1 »). To do this, we compare three discriminant analysis models (figure 14) (linear discriminant analysis, quadratic discriminant analysis and naive Bayesian classifier under the assumption of class normality).

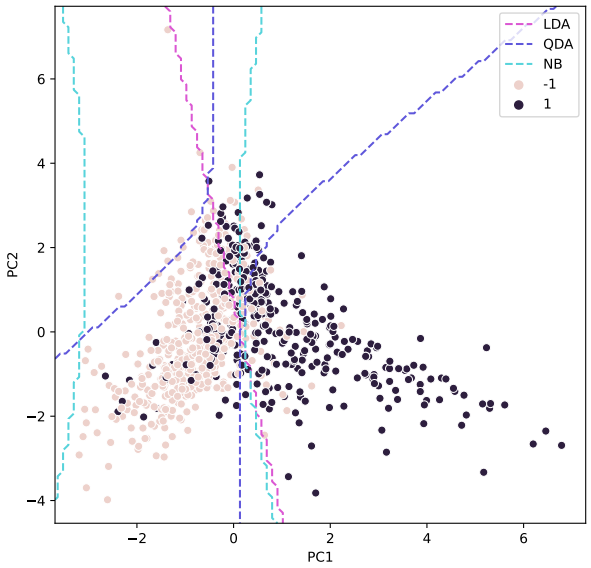

FIGURE 14 – Decision frontiers obtained using the ADL, ADQ and NB models

We can see visually that the three models give fairly satisfactory results. The data along the *PC1* and *PC2* axes are fairly evenly distributed in space.

Let's now compare the performance of each method (figure 15). To do this, we calculate the 10-fold cross-validation errors for the three algorithms.

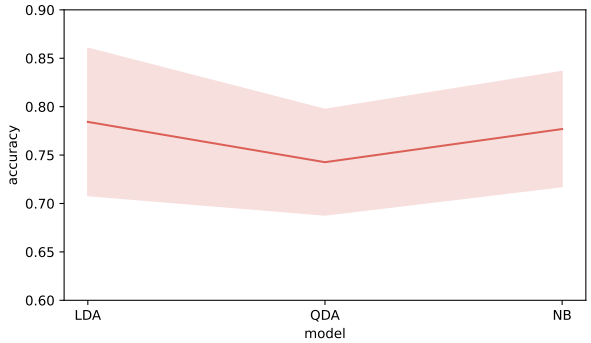

FIGURE 15 – *Accuracy* of the different discriminant analysis models

We note that the highest *accuracy* comes from a linear separation. The linear model has an average *accuracy* of 0.78 over the 10 folds. This is visually explained by a linear decision boundary separating the two groups.

4.3 Logistic regression

4.3.1 Binary logistic regression

Instead of making assumptions about conditional distributions, logistic regression attempts to estimate class membership probabilities directly.

We can see the linear decision frontier obtained by logistic regression in figure 16 (left). Also shown are the level lines for the a posteriori probabilities (0.3, 0.5 and 0.7 respectively).

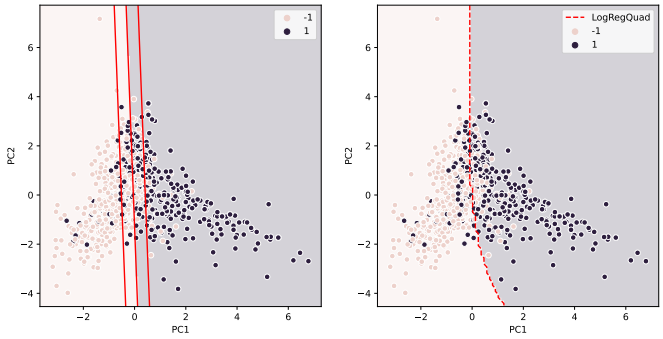

FIGURE 16 – Logistic regression (left) and quadratic logistic regression (right)

The logistic model is therefore written :

$$0.0868 + 1.8235 \times pc1 + 0.0667 \times pc2$$

.

4.3.2 Quadratic logistic regression

It is also possible to do a quadratic logistic regression in order to obtain a quadratic decision boundary. Our set of natural variables $\{PC1, PC2\}$ becomes $\{PC1, PC2, PC1^2, PC2^2, PC1 \cdot PC2\}$ when expanded polynomially. The decision function is then linear in the predictors which are a polynomial expansion of degree 2. The model becomes more flexible but turns out to be less robust than the classical model since it requires examining more parameters (figure 16 on the right).

The quadratic logistic model is then written :

$$\begin{aligned} &0.0083 + 1.9176 \times pc1 + 0.1698 \times pc2 \\ &+ 0.1271 \times pc1^2 + 0.2617 \times pc1 \times pc2 \\ &- 0.0182 \times pc2^2 \end{aligned}$$

4.3.3 Coefficient analysis and variable selection

Wald test. The last classification obtained calculates 6 coefficients β_j. This determines the equation of the decision boundary. A method for testing the significance of the coefficients by an approximation of the variance of the coefficient is the Wald test. This allows to arrive at the statistic W_j called Z-score following approximately a normal distribution. We then impose a critical region $RC = \left\{|W_j| > u_{1-\frac{\alpha^*}{2}}\right\}$ (with $u_{1-\frac{\alpha^*}{2}} approximately 1.96$ or $\alpha^* = 0.05$) on the coefficients and we thus obtain the non-significant β_j. In the present case, the variables considered non-significant are *intercept* ($|W_0| = 0.1025$), $PC1^2$ ($|W_3| = 1.1542$), $PC2^2$ ($|W_5| = 0.4493$).

Likelihood ratio test. We have used an algorithm seen during a SY09 tutorial which, at each step, removes each variable in turn and selects the removal causing the smallest decrease in likelihood, before testing whether this decrease is significant. The stopping condition is the critical region $RC = \left\{-2\log(\frac{L^{-J}(\beta)}{L(\beta)} > \chi^2_{J;1-\alpha^*}\right\}$ where $L(\beta)$ is the likelihood of the model including all variables and $L^{-J}(\beta)$ is the likelihood of the model omitting the subset of variables excluded. Here again, the final model is based on the descriptive variables $PC1$, $PC2$ and $PC1 \cdot PC2$.

4.3.4 Estimation of the model performance

So we train the model with these last three variables on the training set. We obtain the following final model :

$$1.7142 \times pc1 + 0.0868 \times pc2 + 0.2753 \times pc1 \times pc2$$

. Calculating the *accuracy* on the test set gives us a good score of 0.8292.

5 Conclusion

The first necessary step in our approach therefore consisted of a pre-processing of the data. The objective was to arrive at an individuals-variables table, in which the cocktails are explained by the ingredients they contain. The cleaning phase was an opportunity to reduce the number of variables for the first time. A principal component analysis then allowed us to further reduce this number of variables. Visualizing the cocktails in the first factorial plane led us to note that those classified in the *cocktail classics* category can easily be distinguished. This is not the case for the other categories. We completed these observations by applying the unsupervised method called K-means to see what model we can find to classify the cocktails. We then sought to discriminate between these two categories using supervised methods. The K nearest neighbors method allows us to obtain a decision boundary with good precision. What about other models ? We tested three different classifiers (linear discriminant analysis, quadratic discriminant analysis, naive Bayesian classifier). It turns out that a linear model performs better. Finally, a logistic regression was used to establish the values of the coefficients of the decision function. Given a cocktail, we are now able to predict whether or not it belongs to the category *cocktail classics* with good accuracy.

Finally, let us make some remarks on the dataset that we have just studied. We can report a certain lack of data due to its relatively limited size (less than 1000 cocktails). It would be interesting to have a larger number (especially in each category) in order to increase the performance of supervised learning in particular. One solution would be to apply the Data Augmentation technique to multiply the data from the initial *dataset*. This technique would also allow us to manage the balance problems within the distribution of data in the *category*. It may also be desirable to have a larger number of variables such as the skills required of the bartender to make the cocktail or its popularity or price.

A complementary analysis of the dataset could be given through the training of a neural network. For classification, neural networks generally give satisfactory results even if we lose interpretability. They allow us to predict trends on parameters that are significantly outside the learning domain. Its learning condition is a large volume of data.