University of Pisa

Department of Computer Science

Data Mining
Group 18

Supermarket Analysis

Authors:

Donato Meoli
Enrico D'Arco
Luigi Quarantiello

November 5, 2020

Contents

1 Data Understanding

The dataset contains informations about the purchases of each customer of the supermarket. The dataset contains **471910** rows, each of one divided into **8** columns; each entry represents the purchase of a product by a customer, and it comprehends also informations about the date of the transaction, the price of the product and the quantity purchased.
A key information we extracted from the dataset, based on the products descriptions and the frequency of purchases per customer, is that the supermarket in question is not a food store, but instead a grocery store, that sells in particular fast-moving consumer goods.

1.1 Data Semantics

In this section, we describe the semantic of each attribute, providing also some statistics about them.

BasketID
A code that identifies a shopping session for a customer. It remains the same for all the rows that indicate different products inside a single session.
We have **22190** distinct Basket IDs.
We define *good* **BasketID** as a six digit code, instead a *bad* **BasketID** is composed by a character, C or A, followed by a six digit code.

BasketDate
The date and time of a transaction. Basket Dates that correspond to the same Basket ID have the same value.
The dates range from 2010 to 2011, while the times range from 6 to 21.

Sale
A numeric attribute that indicates the price of a product. Since there is no currency associated with the sale price, we assume that it is always the same.

CustomerID
A code that identifies a customer. It is a five-digit code.
We have **4372** unique customer IDs, with some *NULL* values.

CustomerCountry
A categorical attribute that indicates the country of origin of each customer.
We have **37** different countries.

ProdID
A code that identifies a product.
It most cases, it is a five-digit string; sometimes, it has a letter at the end, while other times it is a string of characters.
We have **3953** distinct Product IDs. We define *good* **ProdID** as a five digit code, possibly followed by a variable number of characters; everything else, which are strings followed by some numbers, is labelled as a *bad* **ProdID**.

ProdDescr
A string with a description of a product.
We have **4097** unique Product Descriptions, with some *NULL* values.

Qta
A numeric attribute that represents the purchased quantity for each product.

1.2 Assessing Data Quality

Now, we describe a deeper analysis to assess the quality of data.
First, we checked for duplicate rows, and we found a total of **5232** duplicate records. Since each row represents a single purchase from a customer in a specific date, we have considered these duplicates as errors, and so we decided to drop them. After this correction, we have a dataset consisting of **466678** records.

Then, we focused on the missing values; in the dataset, we have only two columns with some *NULL* values; in particular, we have **CustomerID**, with **65073** null objects, and **ProdDescr**, with **753**.

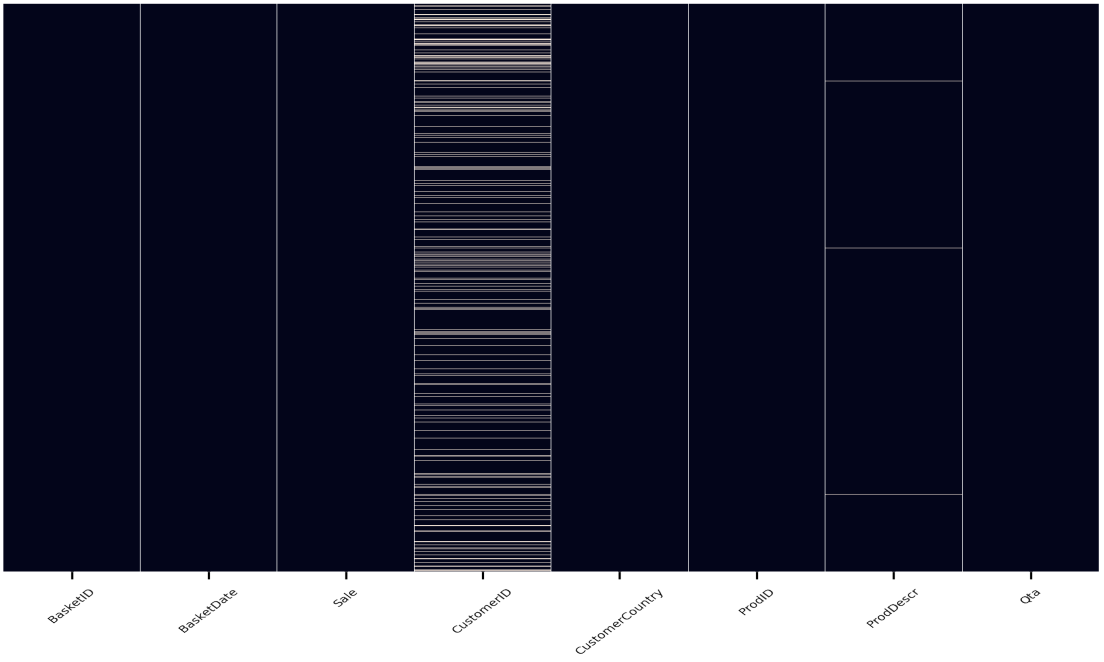


Figure 1: Missing values

We checked attributes like **BasketDate**, **Sale** and **Qta** to see if there were some syntactic errors, but we did not find any of them.

Afterwards, we checked the *semantic accuracy* of the entries in the dataset, to see if some value was not coherent with the logic of its attribute.
In particular, we found:

- 2 records with a negative value for **Sale**;
- Some rows with "*strange*" **ProdDescr**, not referring to a particular product
  - Some of them describe the product's conditions, *e.g.* '*Damaged*', '*wet rusty*', '*Unsaleable destroyed*'. We noticed that these records have all *NULL* **CustomerID**, so we consider these objects not as actual purchases, but instead as internal operations of the supermarket staff;
  - Other ones are referring to some online buying, *e.g.* '*amazon*', '*ebay*', '*dotcom sales*', but we couldn't be able to understand the real meaning of these objects, since they are very few in the dataset;

  - Others, like '*Manual*' and '*Next Day Carriage*', are related to some particular situations, that are not interesting in relation to our analysis;

- Some rows have negative values for **Qta**; here, we discover an interesting pattern: in fact, all the rows with a **BasketID** starting with **C**, have a negative **Qta**. We interpret these records as refunds.

For what regards the outliers, we used a box plot to visualize the two continuous attributes that we have, that are **Sale** and **Qta**.
From these plots, we found that, for both attributes, the box is very flat, meaning that the vast majority of the values fall in a small range. Nevertheless, there are several outliers, someone with a value really far away from the median; these values could represent an issue for the analysis we will perform. Plus, here we can notice the negative values we already found during the semantic analysis.

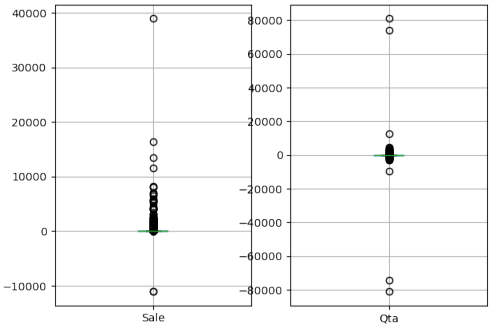


Figure 2: Box Plot for outliers detection

1.3 Variables Transformations

We chosen to categorize the attribute **BasketDate**, by first splitting the date information and the time information, and then

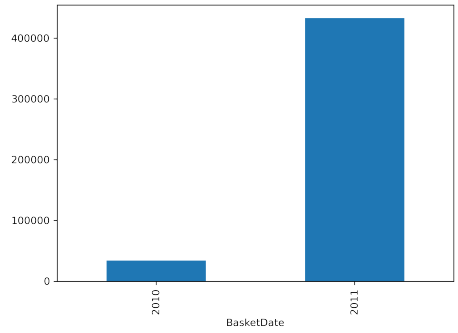
- for the time, we decided to divide the day hours into 5 categories:
  - *Early morning*, from 6 to 9;
  - *Morning*, from 9 to 12;
  - *Lunch time*, from 12 to 15;
  - *Evening*, from 15 to 18;
  - *Late evening*, from 18 to 21.
- for the date, we chosen to keep the week of the transaction.

The rationale behind these choices is that we are more interested in the period in which a transaction occurs, rather that its precise date and time, to be able to cluster customers with a similar shopping behavior; furthermore, since the dataset is not from a grocery store, the customers with several shopping sessions per day or per week are really rare, and so we decided to use a broader partitioning.
After these categorization, we ended introducing **DayTime** and **Week**.

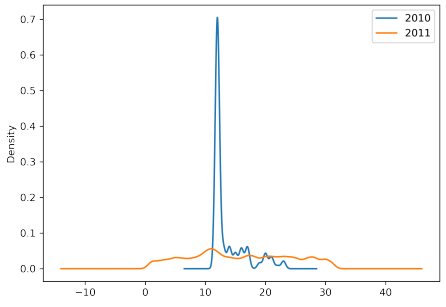
We also decided to create the attribute **TotalPrice**, which is simply the product of **Sale** and **Qta**, and represents the total amount spent by a customer for each record. We made this choice mainly to have a clearer look to the dataset, emphasizing an important information that was not explicit in the original table, and also to simplify the extraction of some features and statistics.

1.4 Variables Distribution

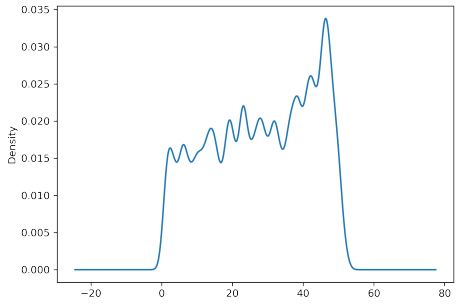
In this section, we will study the distribution of various attributes, by plotting some interesting properties about them.



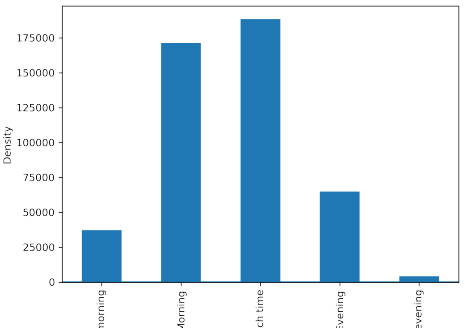
(a) Year



(b) PDF by Year



(c) PDF by Week

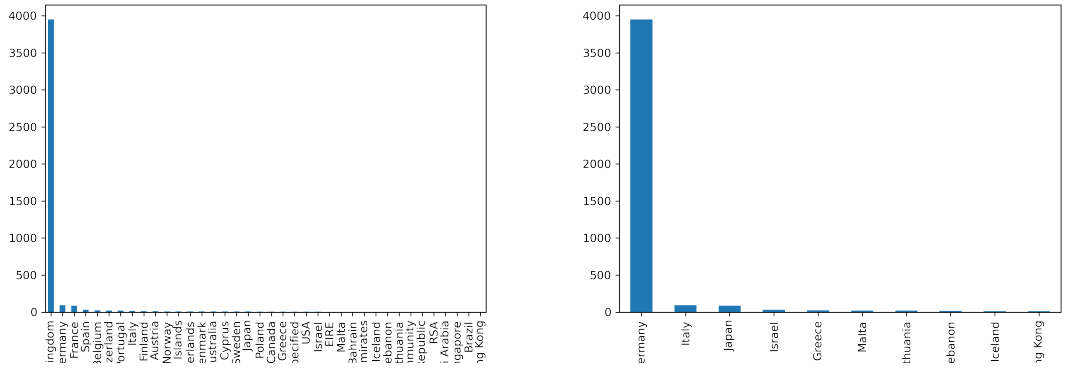


(d) Day Time

Figure 3: BasketDate Distributions

First, we analyse the **BasketDate**.
From the Figure 3a, we can see that the attribute is highly unbalanced; in fact, we have that almost all the records are related to transaction of the 2011, while the objects from 2010 are very few. Indeed, the rows of 2011 represent about the 93% of the whole dataset.
Furthermore, from the Figure 3b, that represents an estimation of the probability density function of **BasketDate** divided by year, we can appreciate the different distributions for the two years.
In fact, for the 2010, we have a very uneven plot, which indicates that the records are not uniformly distributed with respect to the days in a month. That is justified by the fact that, for the majority of the months in 2010, there were registered only transactions from a single day; this day, that is the 12th, is the value for which the plot shows the peak.
On the other hand, the distribution for the 2011 is much more homogeneous, meaning that the transactions were registered for most days in the months of that year.
Another interesting distributions are plotted in Figure 3c and 3d.
In the first one, we can see that the last weeks of the year are the one with more purchases; that is consistent with our knowledge, since those are the weeks closest to Christmas time, that typically represents a great period of shopping.
In the second one, the focus is instead on the hours in a day; we found that, unsurprisingly, the

most popular hourly range goes from 9 to 18.



(a) (b)

Figure 4: CustomerCountry Distributions

Some others statistics can be visualized for the attribute **CustomerCountry**.
In Figure 4a, we can see the distribution of the CustomerID with respect to the country; from the plot, it is clear that the most frequent country is the United Kingdom, that is present in about the 90% of the rows.
To see also some properties of the other countries, in Figure 4b, we plotted the most frequent countries, excluding the UK. Here, we can see that the second most frequent country is Germany, while the other ones are almost irrelevant, since they are in very few objects.

Finally, we see some informations about the correlation of the attributes, to see if some of them are redundant.
From the Figure 5, we can see that almost all the attributes are uncorrelated, except for **TotalPrice**, that shows ah high correlation with **Qta**; that follows what we expected, since **TotalPrice** is, by construction, dependent on **Qta**. So, we conclude that all the original columns are independent, and so we don't need to perform any further manipulation.

Figure 5: Correlation Matrix

2 Data Preparation

2.1 Cleaning the Dataset

In this section, we are going to manipulate our dataset, in order to improve its quality and to facilitate the next analyses.

First of all, we focused on the *NULL* values.
For what concerns the **CustomerID**, since our main goal is to outline the customer behavior, we decided to drop all the missing values for this attribute; therefore, we have eliminated **65073** rows.

Instead, regarding the **ProdDescr**, we notice that all the *NULL* **ProdDescr** have also the **CustomerID** equal to *NULL*; for that, we already eliminated these rows with the previous operation. From the data understanding regarding this attribute, we noticed some values that are not relevant for our analysis; so we decided to drop these rows, being particularly careful to not delete useful objects.

Dealing with the **Sale** attribute, we can perform some kind of data integration.
In the dataset, we have **34** elements with a price equal to 0; that, of course, would represent a problem in the following phases of our work, since the information about the amount spent by a customer is crucial to understand its behavior. For that, we can correct those rows by extracting the informations we need from the rest of the dataset, checking by the **ProdID**. Here, we noticed that the price of a single product can show several variations, maybe due to the market fluctuations; we chose to take the mean of those values to integrate the rows in question.

About **BasketDate**, we decided to remove rows of the 2010, since the data were not taken regularly, and so they could alter our next study. On the contrary, we focus on the 2011 records, which were registered much more uniformly and are more useful and representative.

We also decided to delete the rows with a *bad* **ProdID**, that we defined earlier as not a five digit code plus a variable number of characters. As a matter of fact, at this stage of data cleaning, the dataset contains only 4 unique *bad* IDs, that are *POST*, *C2*, *PADS* and *DOT*; it is evident that they have an ambiguous meaning and so, they are not interesting in our context. As a result,we dropped **1273** elements.

We checked the attribute **CustomerCountry**, especially to see if some customer is associated to more than one country; in that case, we select, for each customer, the country registered with the most number of purchases.

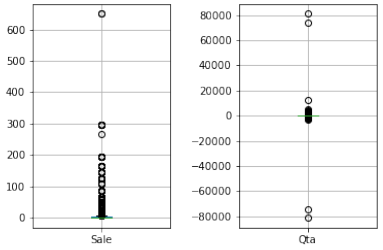
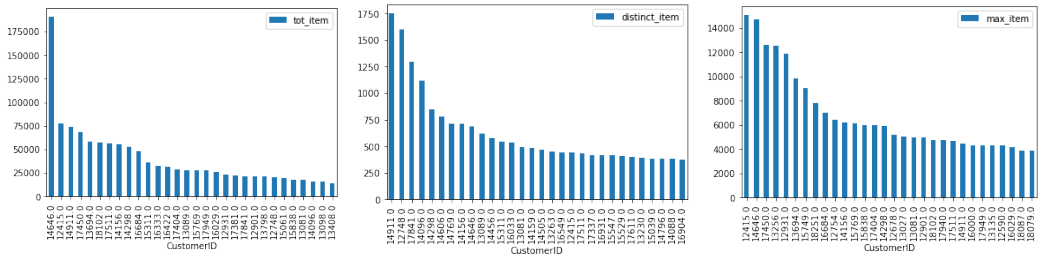


Figure 6: Box plots after the data cleaning

We also looked at the outliers we already detected with the box plots in the previous section.
As we can see in Figure 6, the previous operations already cleaned some of the outliers we had in the original dataset; now we manually check them, to determine if they are errors or not.
In the case of **Sale**, we have that the maximum value is 649.5, which is quite high; in reality, we discovered that all the outliers for this attribute are referring to purchases of furniture or objects sold in big quantities, *e.g.* 60 picnic baskets. So we consider these rows as correct.
In the case of **Qta**, we see just two values that are really away from the others, and they represent huge purchases, with quantities equal to 74251 and 80995; we have also the same quantities in negative. Since this is a very strange case, we thought that they could be the results of an error and so we decided to drop these 4 rows.

We ended up with a cleaned dataset, consisting of **373364** entries.



(a) Total number of items per customer

(b) Number of distinct items per customer

(c) Maximum number of items per customer

Figure 7: Visualization for the extracted features

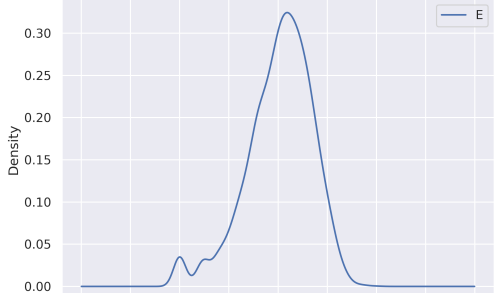
2.2 Feature extraction

Here, we extract new features from the dataset, in order to describe the customers behavior.
First of all, we defined some basic features, like:

- The total number of items purchased by a customer
- The number of distinct items bought by a customer
- The maximum number of items purchased by a customer during a shopping session

In Figure 7, we can see some visualization for the features we just extracted; in particular, they represent the first 30 customers with the biggest values for each feature.
An interesting information is clear from the plot 7c, where we can see that the maximum quantities purchased in a single shopping session are very big; they are all above 3500, with the maximum equal to 15049. These are very high values, unlikely for a retail customer; this led us to think that the supermarket in question also sells wholesale.

In Figure 8, we can evaluate the distribution of the entropy. In particular, we are focusing



on the attribute **TotalPrice**. This value of the entropy represents the variability of the customer's spending habits; a bigger value means that the customer did not have a regular behavior, instead he spent always different amount of money.
We have a small peak corresponding to 0, meaning that there are some customers with very specific habits, but the maximum is reached for ~ 4, which means that the majority of the clients have a quite unpredictable behavior.

Since the focus is on the purchasing behavior of the customers, we decided to study other features, in order to deeply understand the information we have. We chose to extract information about:

- The number of basket per customer
- The number of *bad* basket per customer
- The number of purchased products
- The number of returned products
- The amount of money spent for each customer
- The amount of money refunded per customer

Thanks to these features, we were able to see better the kind of customers registered in the dataset. First, we checked if there were customers with a number of returned products bigger than the purchased ones; as a matter of fact, we found **18** rows of this kind. We consider these as *bad* customers, since they have made more returns than purchases; this is clearly an impossible situation, maybe due to the fact that some transaction was not registered in the dataset. Anyway, we decided to delete these customers.
Plus, we inserted two other columns:

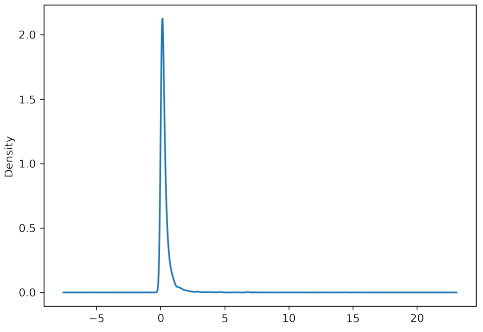
- **ActualQta**, which is the algebraic sum between the positive and the negative quantities
- **ActualSpent**, equal to the difference between the money spent and the one refunded

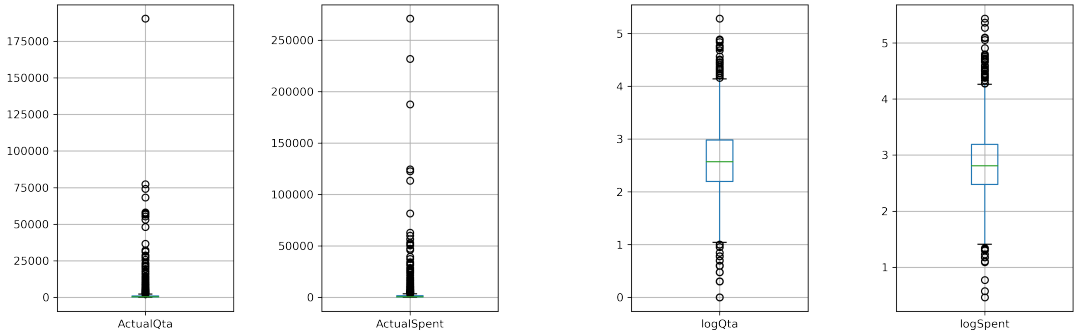
We also deleted all the customers with the **ActualQta** equal to zero, since, in the end, they didn't buy anything.

Now we extract some information about the average values, as:

- **AvgPrice**, the ratio between **ActualSpent** and **ActualQta**, that represents the average price of the products bought by a customer
- **AvgBaskValue**, the ratio between **ActualSpent** and the number of *good* baskets, which represents the average amount of money spent for each basket per customer

Another interesting feature is the **MonthFreq**, that we computed as the ratio between the number of *good* baskets and 12, the number of months. We can visualize this attribute in the Figure 9, where we can clearly see that the





(a) (b)

Figure 10: Box plots for ActualQta and ActualSpent, before and after the log

peak is really close to 0; in fact, the mode of this column in equal to **0.083**, which means that the majority of the customers went to this supermarket just once in the year.

Analyzing the Figure 10a, we can see that the attributes **ActualQta** and **ActualSpent** are really spread; for this reason, we consider their logarithm in base 10, so that the difference between high values has a different weight. We can appreciate the changes in the Figure 10b, where the two distributions are much more compact.
After the transformations of these attributes, we saw that also **AvgPrice** and **AvgBaskValue** were quite spread, and so we decided to compute the same operation to them.

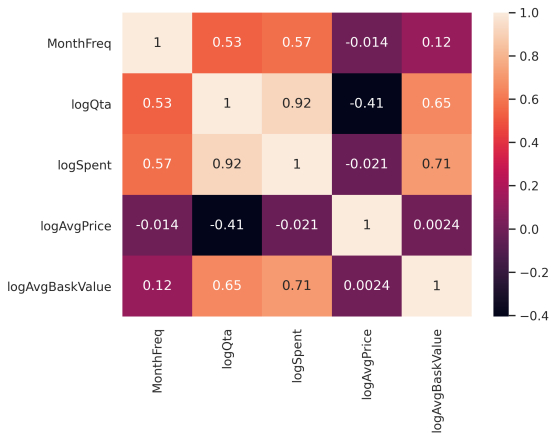


Figure 11: Correlation Matrix

Now, we can visualize the correlation between the attributes. In Figure 11, as expected, we find that the average basket value is highly correlated with the total quantity purchased and the amount of money spent. Furthermore, we can see that also the monthly frequency is correlated with the amount spent and the total quantity. In the end, of course, the quantity purchased is very highly correlated with the total amount spent.

In Figure 12, we can visualize the Pair Plot for the attributes we have.
On the diagonal, we have the density plots, where we can appreciate the distribution of the features. Instead, the other scatter plots show the relationship between two variables. By analyzing those, we can have a confirmation of what we already found thanks to the previous plot.

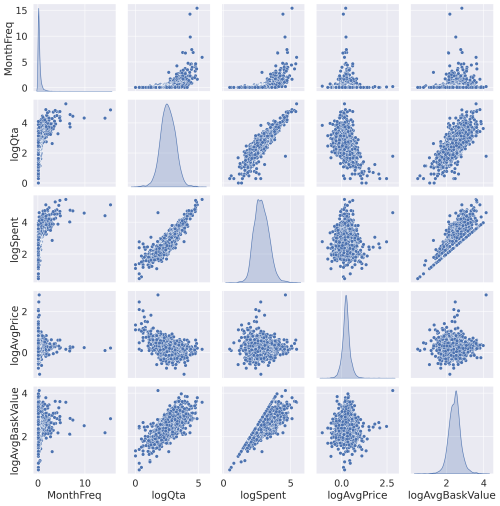
3 Clustering

We now run and compare some clustering algorithm in order to find some structures among the data. First we start with a basic *K-Means* followed with some *Hierarchical clustering techniques* and *DB-SCAN*.

3.1 Preprocessing

The data matrix was first standardized and then the two principal components were extracted. The choice was between two and tree principal component since they retain respectively ∼0.75 and ∼0.85 of the variance of the data.

We selected the two principal components because it performed better on K-Means and in this way a visual inspection was possible.



12: Pair Plots

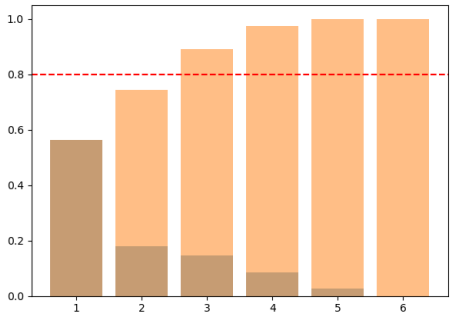


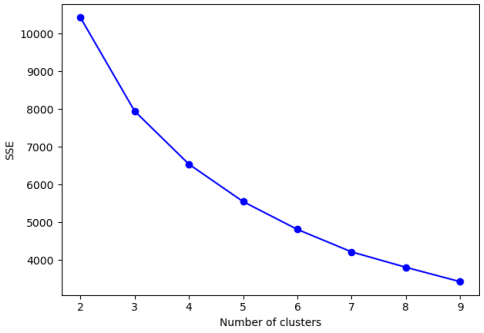
Figure 13: Explained variance ratio

3.2 K-Means

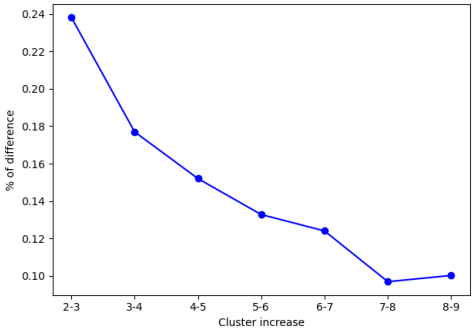
The algorithm run for K ranging from two to ten clusters. For each iteration the SSE and the average silhouette value were computed.

By looking at the plots in Figure 14 we can see that there isn't a prominent *elbow shape* so to get the optimal number of clusters the analysis of more metrics was necessary. At first we saw that for K moving from two to three the difference of the SSE in percentage was quite high (∼0.24) while for all the other values the difference was below the 0.2, meaning that the most of the decrease in the SSE was in moving from 2 to 3 clusters. By inspecting the silhouette score it's possible to see that it has its maximum for K equal to two, followed by K equal to 3. In conclusion we opted of K equal to three to get a trade off between high silhouette and low SSE.

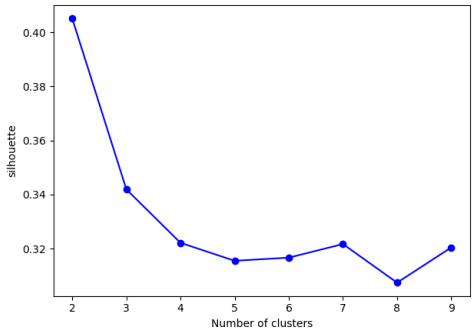
The resulting clusters have a comparable number of points. In particular *Cluster 0* has 990 points, *Cluster 1* 1137 and *Cluster 2* 2072. By looking at the plot of the Silhouette score (Figure 15) we



(a) SSE



(b) Percentage of decrease in SSE



(c) Average silhouette

Figure 14: *K-Means* metrics

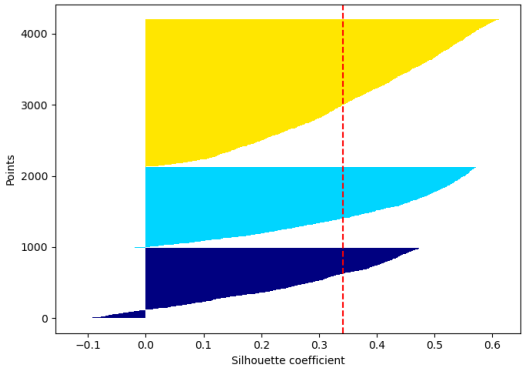
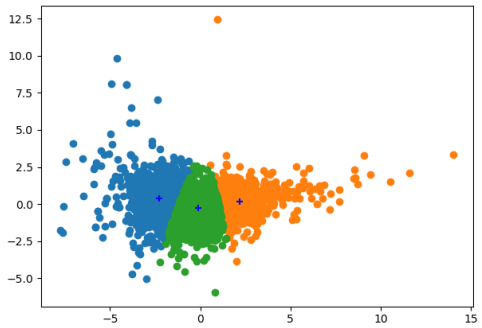


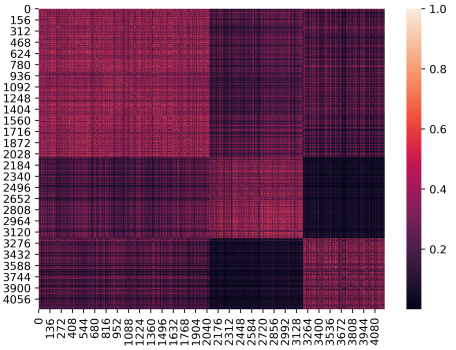
Figure 15: Silhouette score for each data point. Each different color represent a different cluster, the dotted red line is the average silhouette coefficient.

see that a small fraction of points in *Cluster 1* and *Cluster 0* have a negative value but overall the Silhouette suggest a good clustering.

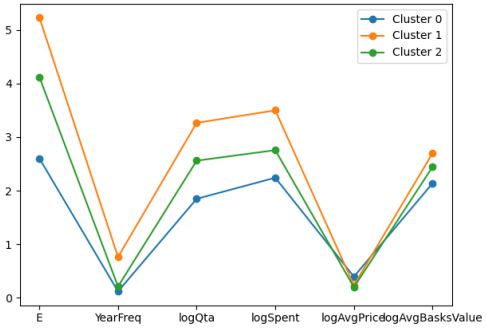
The similarity matrix (Figure 16b) shows that the points within the same cluster are similar to



(a) Clustering result

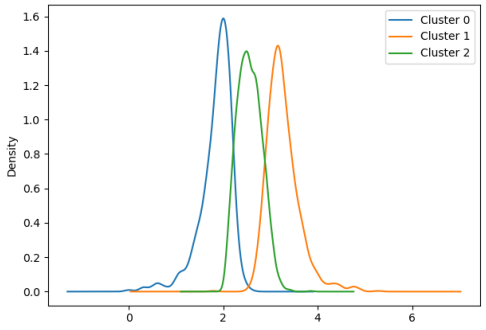


(b) Similarity heatmap

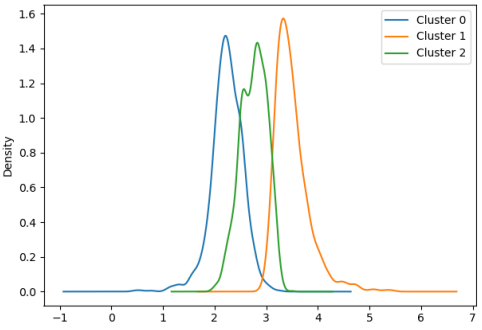


(c) The plot shows the average values for each attribute divided by the clusters. In particular is possible to see that *Cluster 0* contains the most frequent and spending customers, followed by *Cluster 2* and *Cluster 1*

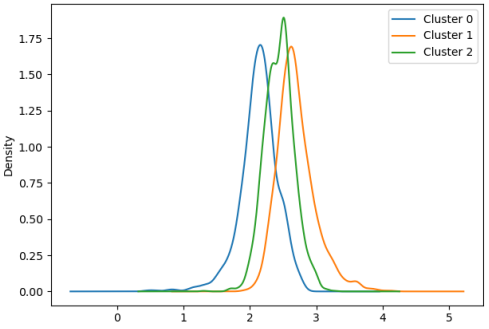
each other when similarity is measure with $e^{-d(x,y)}$ where x and y are data points and d is the euclidean distance. In particular *Cluster 2* includes points really similar to each other.



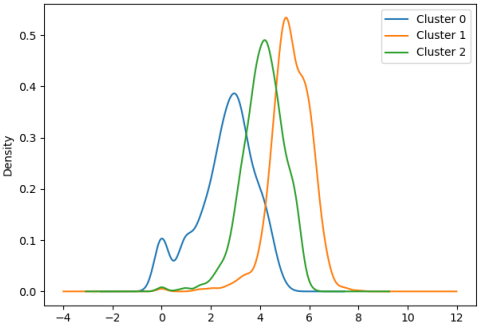
(a) Distributions of *logQta*



(b) Distribution of *logSpent*



(c) Distribution of *logAvgBasksValue*



(d) Distribution of *TPEntr*

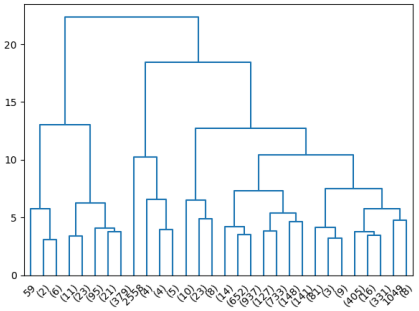
Figure 17: Distribution of some attribute based partitioned with respect the clusters

3.3 Hierarchical clustering

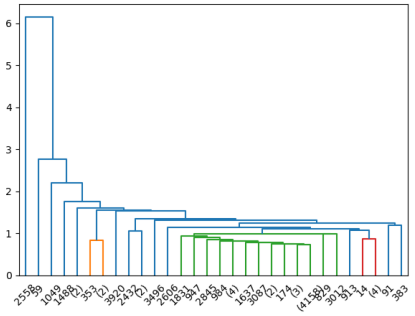
Different kinds of algorithm are used: *Complete Link*, *Single Link*, *Ward* and *Centroid*. The CPCC coefficient are respectively 0.51, 0.62, 0.44 and 0.74 meaning that the best results are obtained with *Single Link* and *Centroid*. By looking at the dendogram, if we cut the tree by selecting two clusters we see that the vast majority of the data fall into a single cluster leaving the last merge between a large cluster and a small one, in some cases even a singleton. The most *balanced* result is obtained with the *Ward* method though it has the lowest CPCC.

3.4 DBSCAN

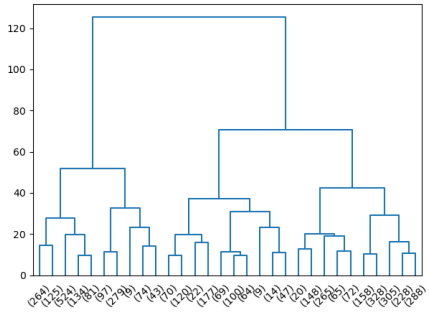
The dataset doesn't look suitable for DBSCAN since it is a large and dense area surrounded by a set of noise points. With different combination of *eps* and *MinPts* the algorithm is able to find more clusters (up to 6, including noise points, with *eps* equal to 0.3 and *MinPts* equal to 5), those clusters have ad small size compared to the central one which lead us to think that aren't representative. For a given value of *MinPts* the selection of *eps* was made by checking the *Knn* distance with K equal to *MinPts*.



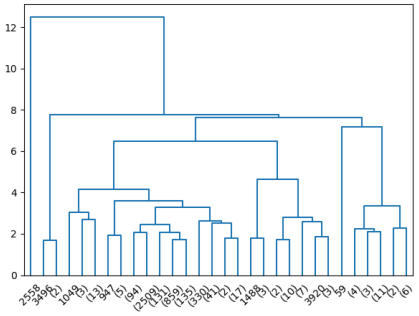
(a) *Complete Link*



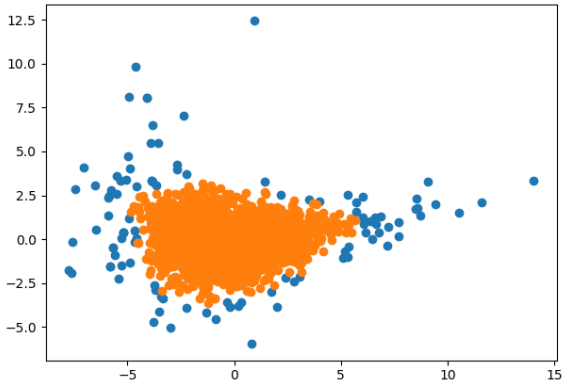
(b) *Single Link*



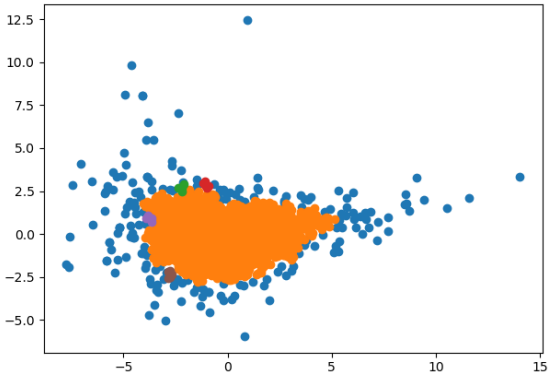
(c) *Ward*



(d) *Centroids*



(a) *DBSCAN* with *eps* equal to 0.6 and *MinPts* equal to 8. There is a large and dense cluster surrounded by noise points



(b) *DBSCAN* with *eps* equal to 0.3 and *MinPts* equal to 5. There is still a large and dense cluster surrounded by noise point but a large number a clusters are presents. Those clusters are really small compared to the central one.

Figure 19