Giulio Federico

Data Mining and Machine Learning – Artificial Intelligence and Data Engineering
University of Pisa

Introduction

The application was designed to be used by **supermarket employees** to facilitate the choice of *products to showcase* and to constantly monitor *customer segmentation* to better target promotions. The extraction of the necessary information will be done with association rule algorithms (comparing Apriori and FP-Growth) and with clustering algorithms (comparing K-Means, DBSCAN and AGNES)

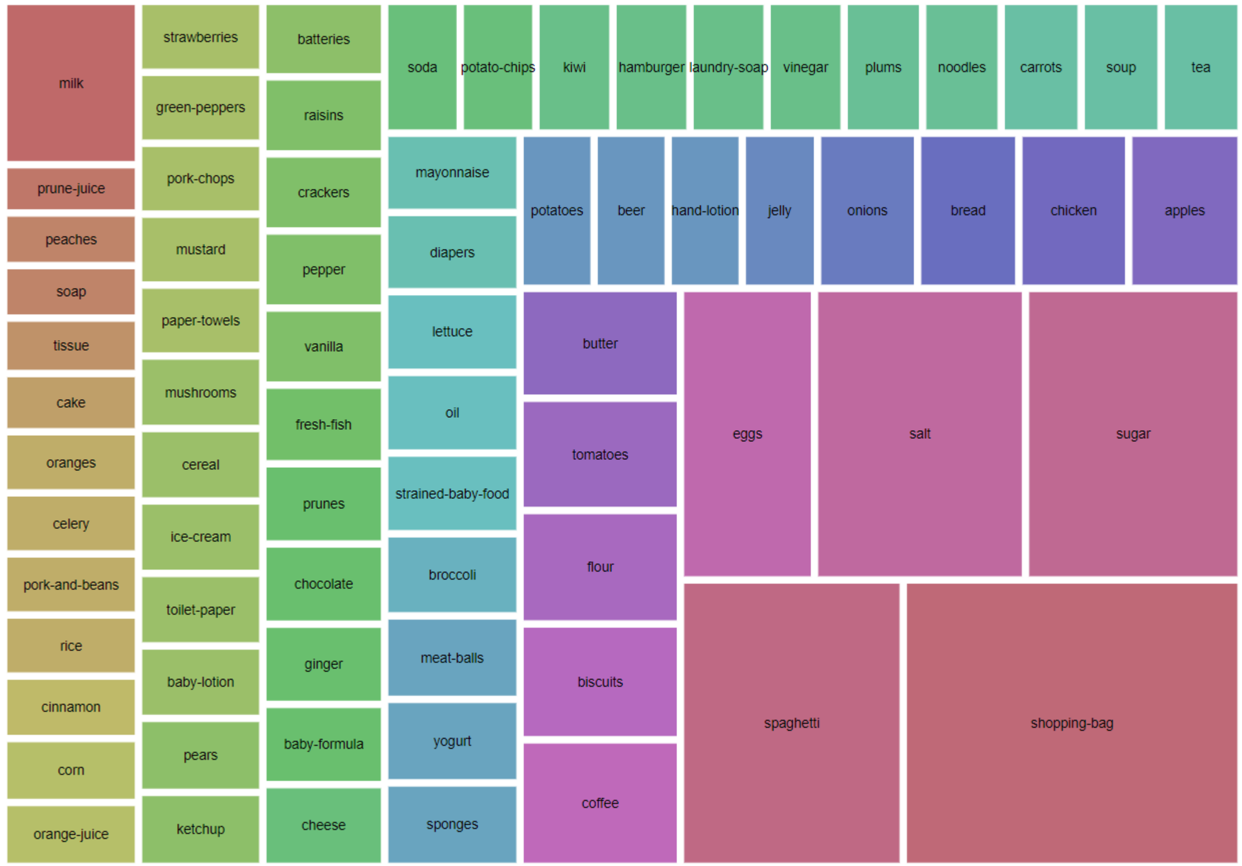
DATA UNDERSTANDING

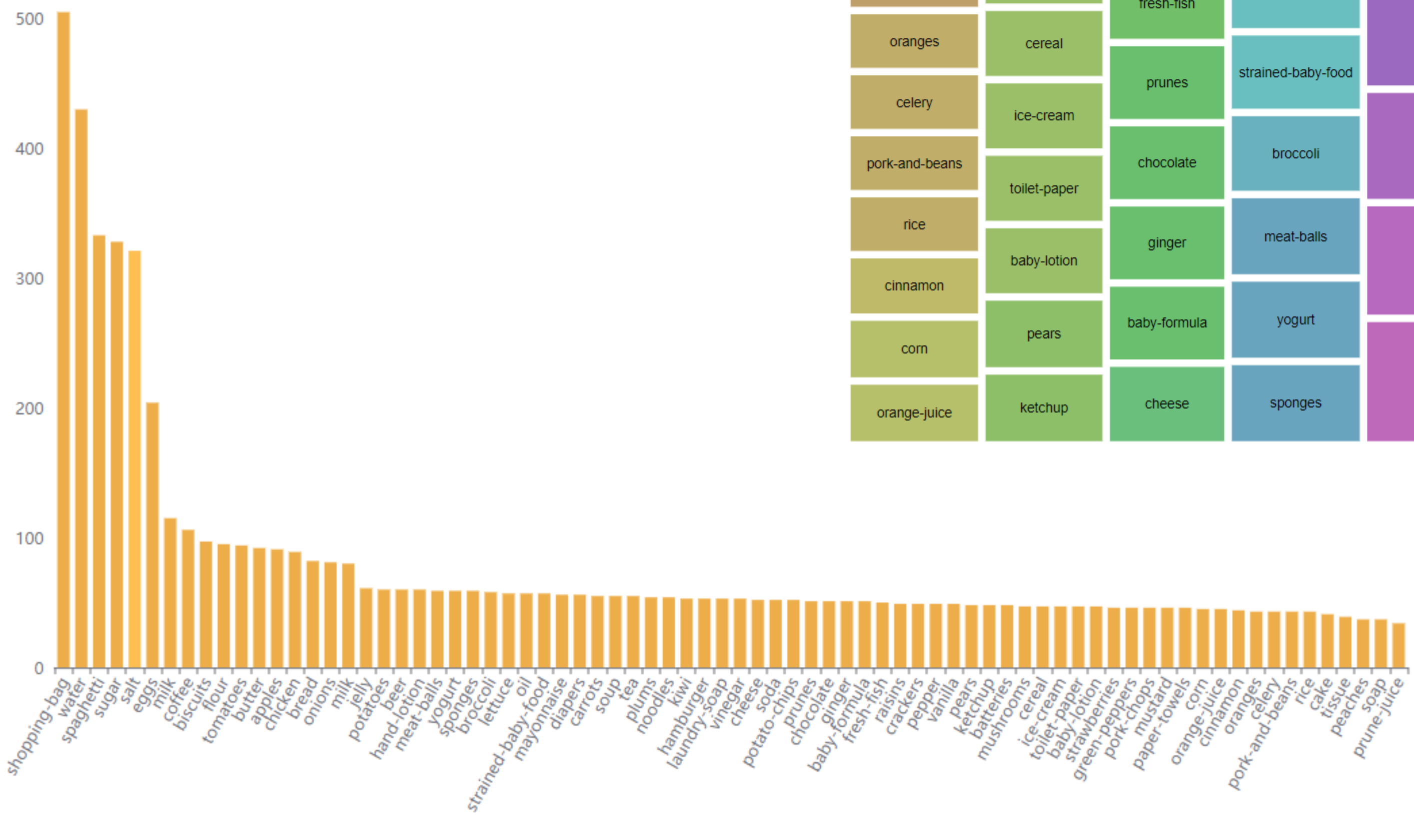
1.1 Basket analysis

row_id	tid	custId	itemPurchased	price	lot	Timestamp
...	...	...	...	...	...	...
19222	2817	98	water	2.80	4	2021/02/12
19223	2817	98	milk	0.70	1	2021/02/12
19224	2818	119	eggs	3.40	1	2021/02/12
...	...	...	...	...		...

- The dataset with which the application will extract the information and then generate the suggestions has the structure like the table above.
- Only the data of the last week will be taken, because the only ones to grasp the trend of the last week.

The distribution of items in transactions is uniform except for the *shopping-bag*, *water*, *spaghetti*, *sugar*, *salt* and *egg* items, that are really frequent.





DATA UNDERSTANDING

1.2 Customer analysis

This is the table from which to extract information for clustering.

cust_id	name	email	marital_status	sex	date_of_birth	age	job	job_category	Status
…	…	…	…	…	…	…	…	…	…
720	Darrel Canet	DarrelCanet@gmail.com	unmarried	Male	Fri Oct 23 00:00:00 CET 1931	89	Recruiting Manager	Retail	Mass Customer
721	Katlin Creddon	KatlinCreddon@yahoo.com	married	Female	Thu Aug 22 00:00:00 CET 1935	85	VP Quality Control	Retail	High Net Worth
…	…	…	…	…		…	…	…	…

income	annual_expenses	beverage	bakery	canned_foods	dry_foods	frozen_foods	meat	produce	cleaners	others
…	…	…	…	…	…	…	…	…	…	…
12000	2350	0.8	0.1	0.5	0.4	0.5	0.8	0.3	0.2	0.4
78500	10119	0.3	0.6	0.1	0.2	0.2	0.3	0.8	0.7	0.3
…	…	…	…	…		…	…	…	…	…

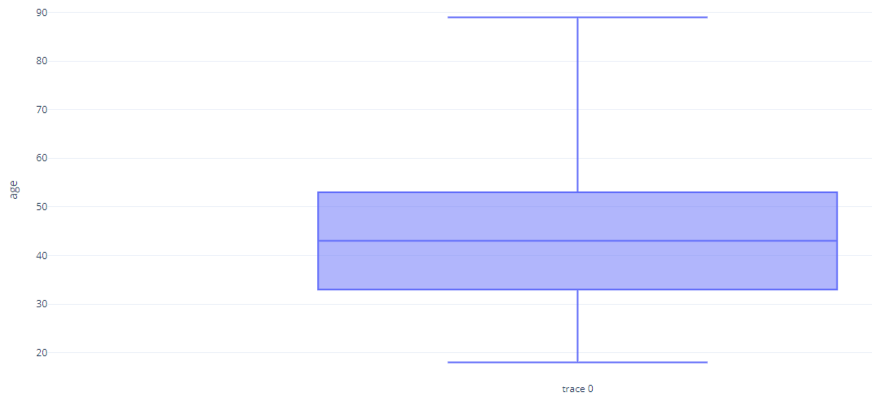
In addition to basic information, the interest for each category of the supermarket is also recorded for each customer (in possession of a points card) with a score from 0 (no interest) to 1 (high interest).

label	Number of instances
unmarried	2954
married	955
Missed	0

Figure 1: pie chart of **marital status**.

label	Number of instances
Female	2038
male	1871
Missed	0

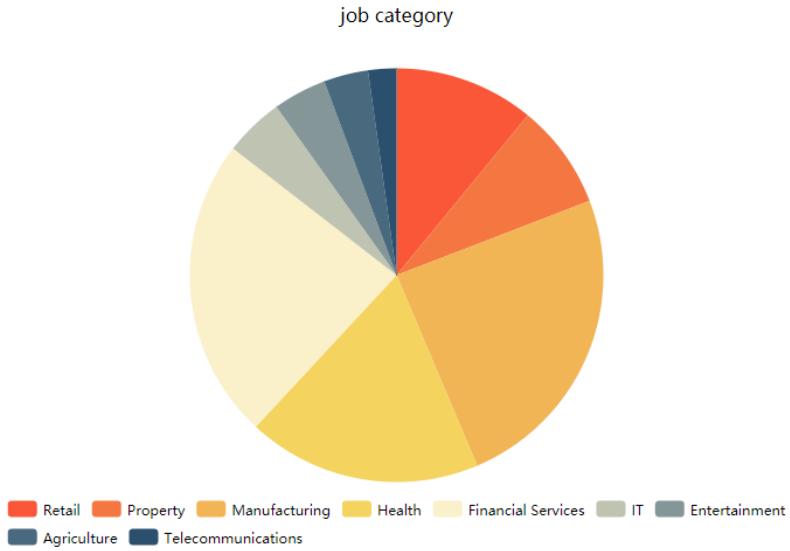
Figure 2: pie chart of **sex**.



N	min	Q1	Median	Q3	Max	Mean	Missed
3909	18	33	43	53	89	43	0

Figure 3: box plot of **age**.

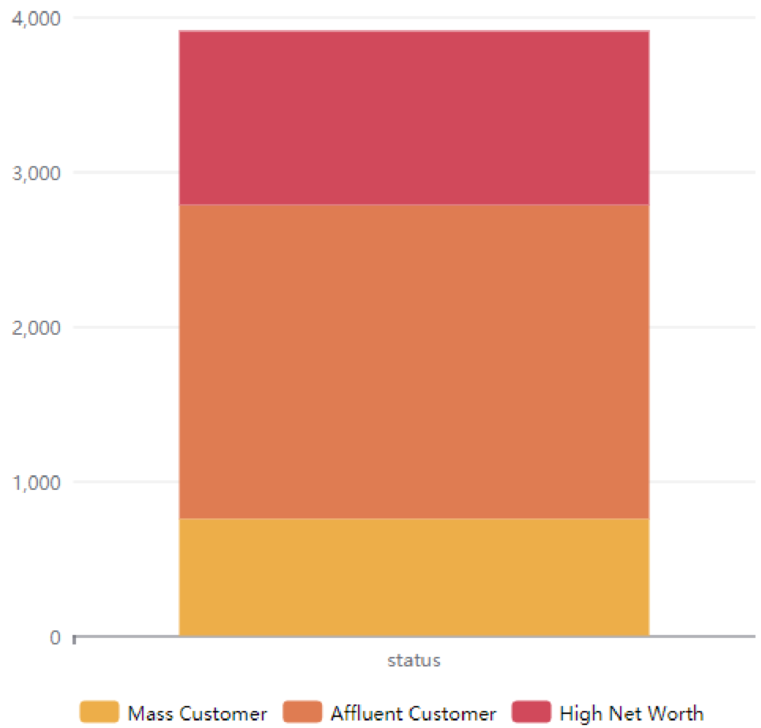
Figure 4: word cloud of **job category.**



Missing!

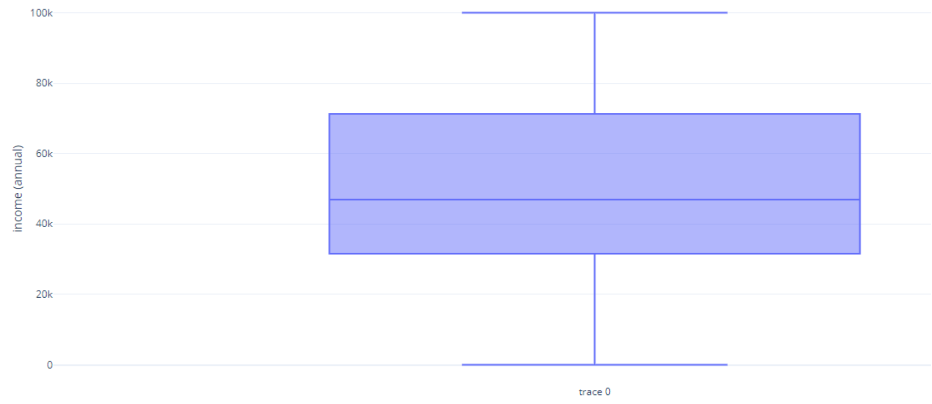
label	Number of instances
Retail	356
Property	267
Manufacturing	796
Health	595
Financial Services	767
IT	151
Entertainment	136
Agricolture	113
Telecommunications	72
Missed	656

Figure 5: pie chart of **job category**.



label	Number of instances
Mass Customer	1124
Affluent Customer	2028
High New Worth	757
Missed	0

Figure 6: stacked column chart of **status**

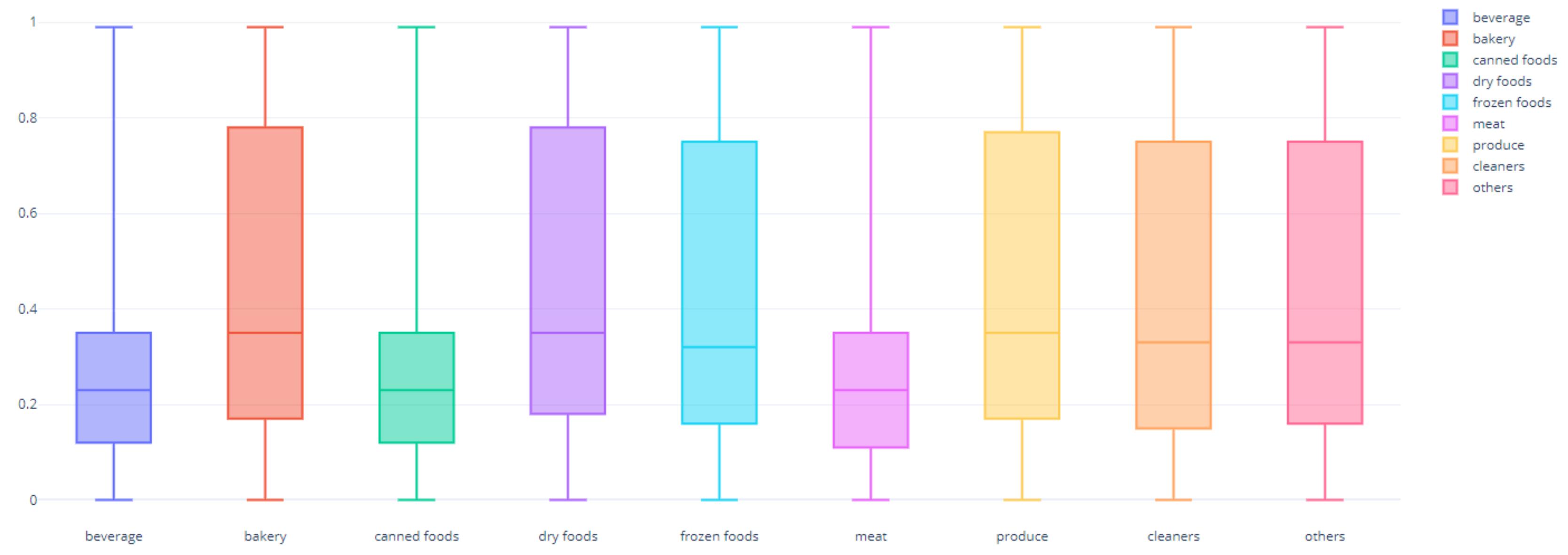


N	min	Q1	Median	Q3	Max	Mean	Missed
3909	0	31k	45k	53k	71k	42k	0

Figure 7: box plot of **income** (*annual*).

N	min	Q1	Median	Q3	Max	Mean	Missed
3909	200	1466	2836	6182	11.986	3173	0

Figure 8: box plot of **annual expenses.**



Attribute	min	Q1	Median	Q3	Max	Mean	Missed
beverage	0	0.12	0.23	0.35	0.99	0.31	0
Bakery	0	0.17	0.35	0.78	0.97	0.36	0
Canned foods	0	0.12	0.22	0.35	0.98	0.31	0
Dry foods	0	0.18	0.35	0.76	0.91	0.37	0
Frozen foods	0	0.16	0.32	0.74	0.98	0.50	0
Meat	0	0.11	0.23	0.36	0.99	0.31	0
produce	0	0.17	0.35	0.77	0.97	0.36	0
Cleaners	0	0.15	0.33	0.75	0.94	0.49	0
others	0	0.16	0.33	0.75	0.99	0.51	0

Figure 9: box plot of **food categories.**

Data cleaning

- check each record to identify any **missing values**.
- check only the presence or absence of the purchased product. If the value exists, also check that it belongs to the domain of products that can be sold by the supermarket, otherwise **discard** the record.

Data reduction

- remove all attributes (except that of the purchased product) as they are useless for the purpose of the analysis to be done (**dimensionality reduction**).
- remove all records containing the product "shopping-bag" and "water" (*is there really a relationship between product X and one of the two?* No) (**numerosity reduction)**.

Data transformation

- Transform the input data into a **binary matrix** to be manipulated by *Apriori* and FP-Growth.
  1) Group all records together if they belong to the same transaction obtaining a **transactional database**.
  2) Create a column for each possible item and indicate with ‘1’ the presence of that item in that transaction, otherwise indicate it with ‘0’.

row_id	tid	custId	itemPurchased	price	lot	Timestamp
...	...	...	...	...	...	...
19222	2817	98	water	2.80	4	2021/02/12
19223	2817	98	milk	0.70	1	2021/02/12
19224	2818	119	eggs	3.40	1	2021/02/12
...	...	...	...	...		...

tid	Items purchased
...	...
1023	milk, sugar, bread, salt, kiwi
1024	meat, green peppers, broccoli, soda
1025	eggs, apple, salt, hamburger, mushrooms
...	...

milk	meat	bread	apple	pork chops	spaghetti	pears	...	soda
...	...	...	...	...	...	...	...	...
1	0	1	0	0	0	0	...	0
0	1	0	0	0	0	0	...	1
...	...	...		...	...	...	...	...

Figure 10: from raw dataset to binary matrix.

Data cleaning

- if the marital status is missing then the status will be replaced with the ***mode*** status of the costumers.
- if the sex is missing, then the tuple is ***deleted***.
- if the age and date of birth are inconsistent, the age will be replaced with the ***mean*** age of the costumers.
- if the job category is not specified then the tuple will be ***deleted***.
- if the social status is missing then it will be ***deleted***.
- if the annual income or expenses are missing then they will be replaced with the ***mean***.
- all the values for the categories, if missing or out of range [0,1] (perhaps caused by noise) will be replaced with the ***mean***.

For all the **nominal attributes** it will also be checked if the assumed value falls within those allowed, otherwise they will be treated as missing, and therefore, depending on the attribute, it will be *deleted* or replaced with the *mode*.

Data reduction

- Reduce the **dimensionality** by eliminating the following attributes (none are useful for clustering):

  1) cust_id

  2) name

  3) email

  4) dateOfBirth attribute can also be dropped as we have the same information with the *age* attribute.

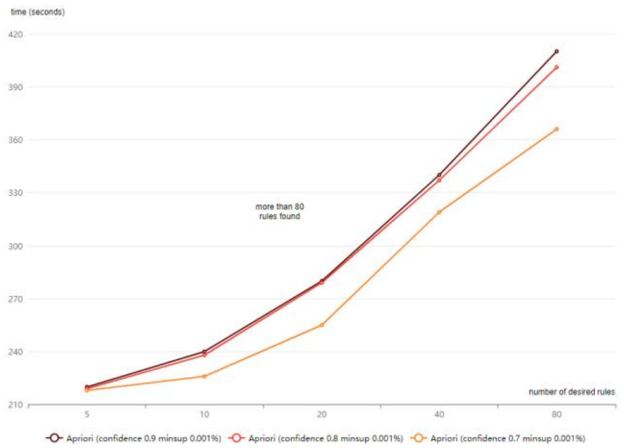
  5) job attribute can be deleted leaving *jobCategory* instead as it guides better clustering.

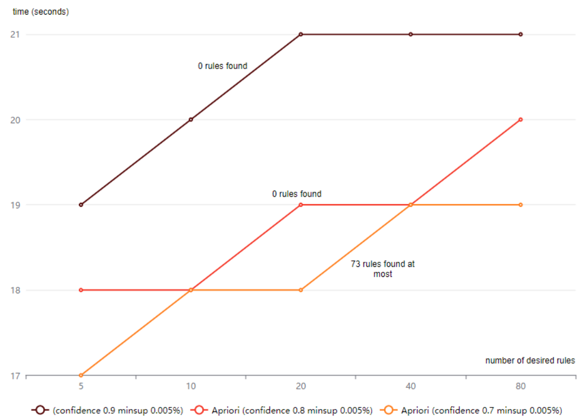
- A **min-max normalization** was done on all numeric attributes.

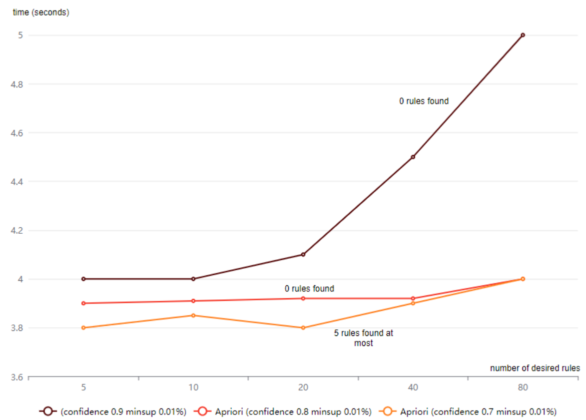
  So we finally reduced the size from 21 to 16. Furthermore, no significant correlation between attributes was found to reduce the size even further.

Apriori

- The 3 Apriori charts show how a minimum support of 0.001% is too low and causes a large number of candidates to be tested bringing the user's waiting time on average 5 minutes, which obviously is unthinkable.
- We note how to increase the support very little, the performance increases a lot going from minutes to (few) seconds.
- We also note how by increasing confidence the time increases as Apriori returns the first desired rules it finds, but which satisfy the indicated minconfidence.

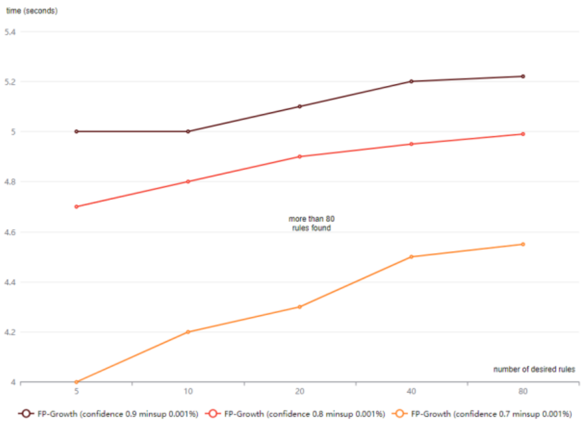


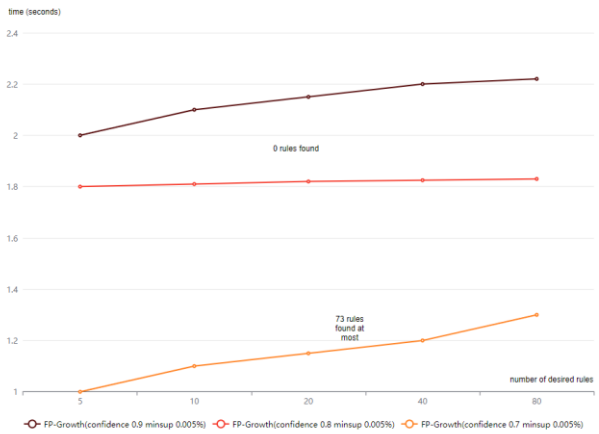


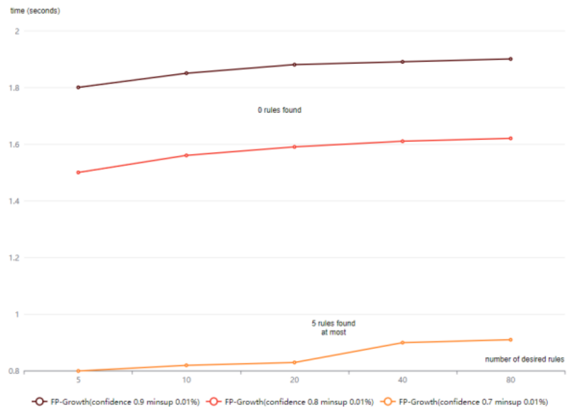


FP-Growth

- where Apriori took minutes, here FP-Growth takes a maximum of 5.3 seconds. This is because FP-Growth only needs a complete scan of the dataset 2 times, while Apriori does it at each iteration. Also no candidates are generated, but mining is done by recursively building a tree that represents smaller and smaller and compactly saved database projections, so it is faster to mine.
- FP-Growth may not be effective on large datasets as it works a lot on memory, but for the small-medium supermarket for which the application is designed it is not a problem.





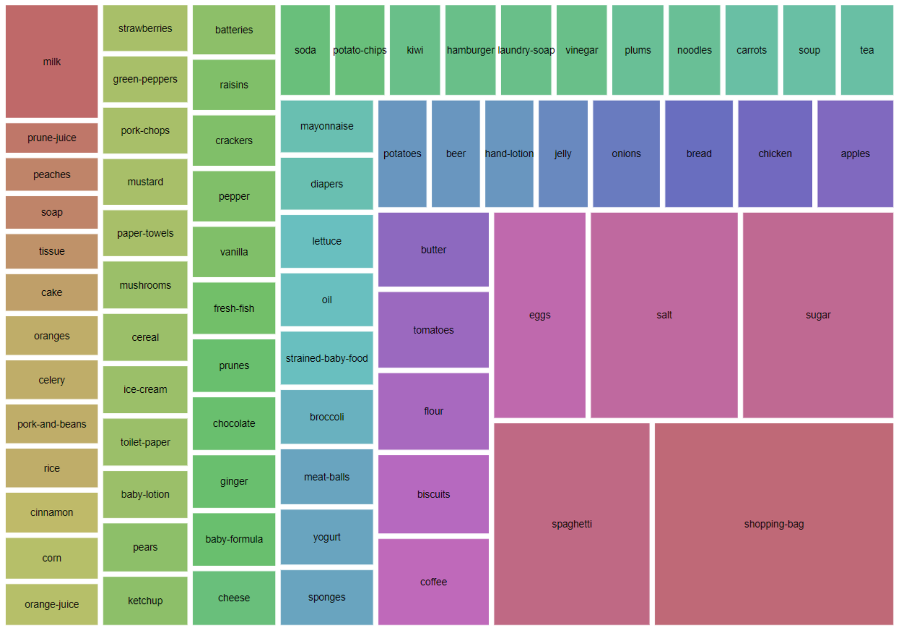


- FP-Growth will be chosen as the mining algorithm of the application.

An example of rules found with a confidence of 0.7% and a minimum support of 0.7%

Association rule	support	confidence	lift	Conv
apples, flour, butter → eggs	76	0.75	1.42	1.79
meat balls, soup → beer	115	0.73	1.39	1.71
hamburger, cheese → chicken	117	0.71	1.35	1.61
pizza, snack, potato-chips →onions	71	0.71	1.36	1.62
broccoli, eggs, bread → orange-juice	70	0.71	1.35	1.57

- *Confidence* indicates a good implication.
- The *lift index* indicates that the rules are enought independent. This 'enought' is because items such as eggs, beers, chicken, onions and orange-jouice are among the most frequent items.



Hopkins statistic

Does the dataset tend to be clustered?

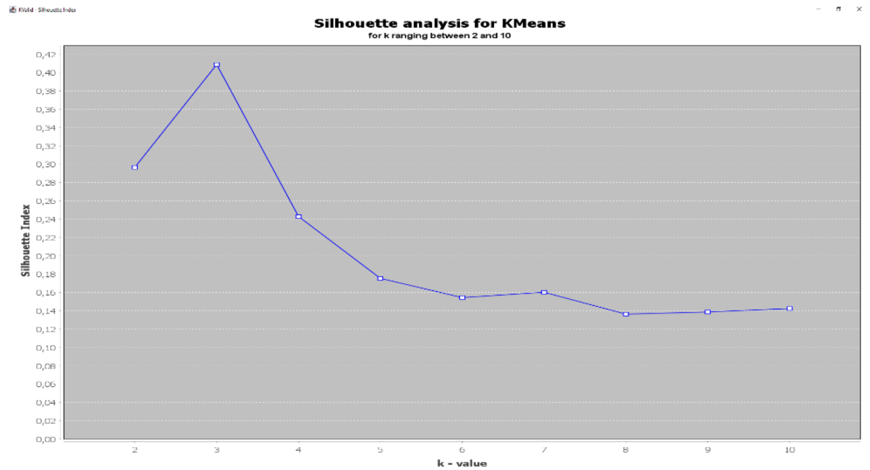
$$H = \frac{\sum_{i=1}^{s} y_i}{\sum_{i=1}^{s} y_i + \sum_{i=1}^{s} x_i}$$

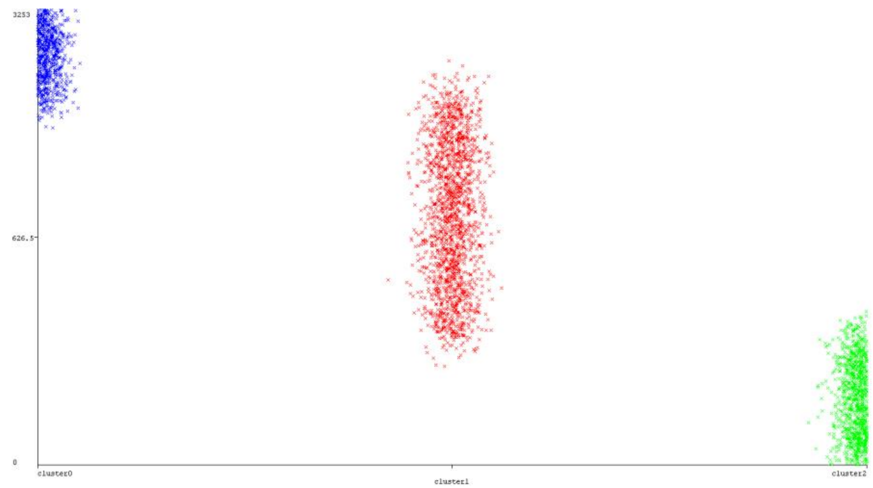
A java code was written to simulates this calculation. If the behavior of the samples taken is the same as those generated synthetically then the distribution is uniform and there are no significant clusters.

This method has been repeated several times. The final average is: *0.685*.
It therefore seems that a certain tendency to be clustered exists.

K-MEANS

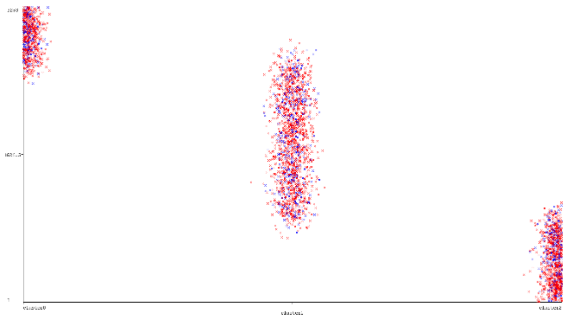
- Compute the value of the *silhouette index* for several k (using KValid package). The maximum k will be 10, this is because over ten clusters the supermarket employees will find it difficult to identify significant aspects from the results that will be shown to them.





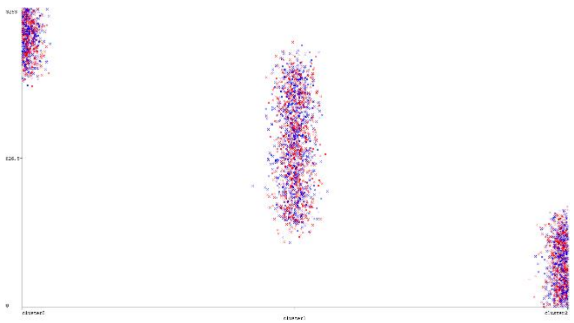
- The best number of clusters is 3, with an average silhouette index of **0.4090** which, as Weka himself suggests, is a **weak structure**.
- **3 main clusters** have been identified: 627 instances (20%) inside cluster0, 1686 instances (52%) inside cluster1 and finally 931 instances (29%) inside cluster2.

What relevant information can be extracted?



married unmarried

Figure 19: distribution of **social status** instances in the 3 clusters.



male female

Figure 20: distribution of **male/female** instances in the 3 clusters.

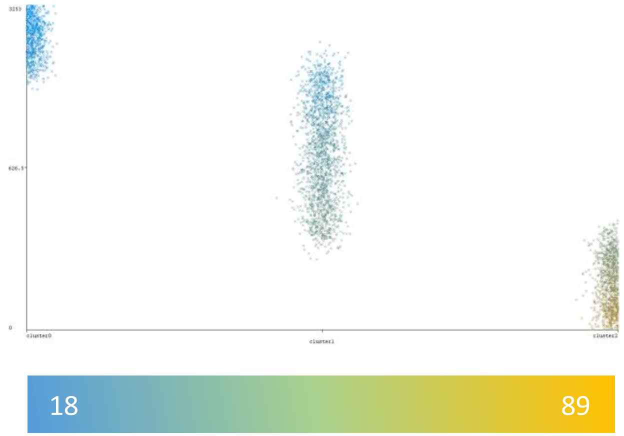
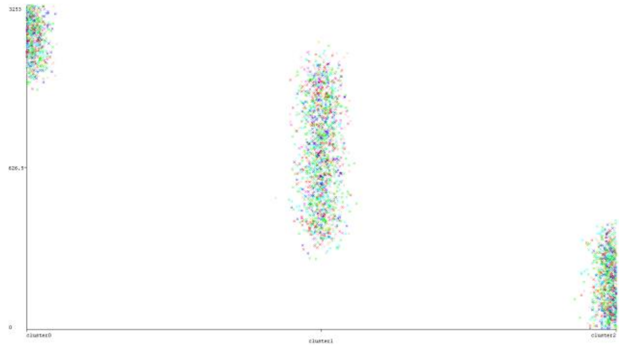
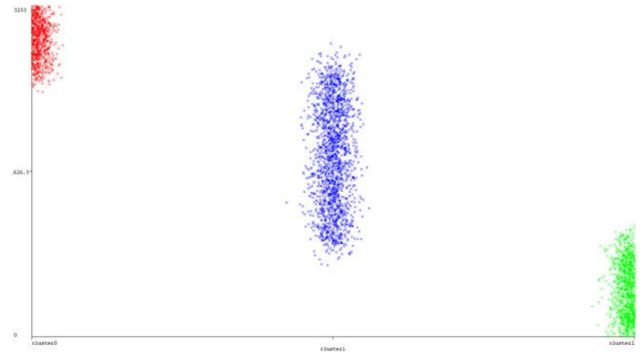


Figure 21: distribution of **age** in the 3 clusters.



Retail Property Manufacturing Health FinancialServices IT Entertainments Agricolture Telecommunications

Figure 22: distribution of values of **job** category in the 3 clusters.



Affluent Customer Mass Customer

High Net Worth

Figure 23: distribution of the values of **social status** in the 3 clusters.

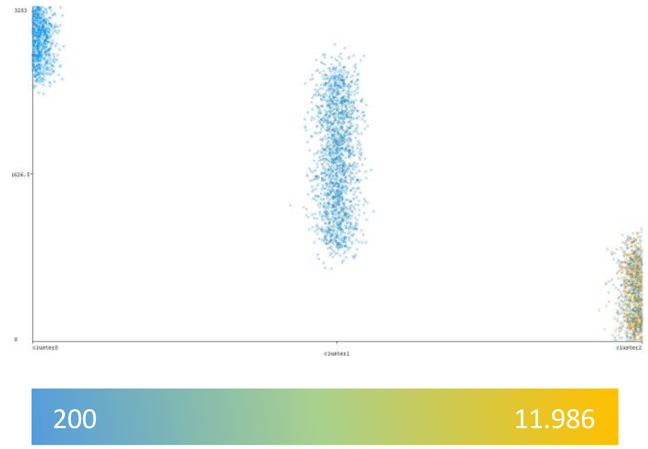


Figure 24: distribution of **annual expenses** in the 3 clusters.

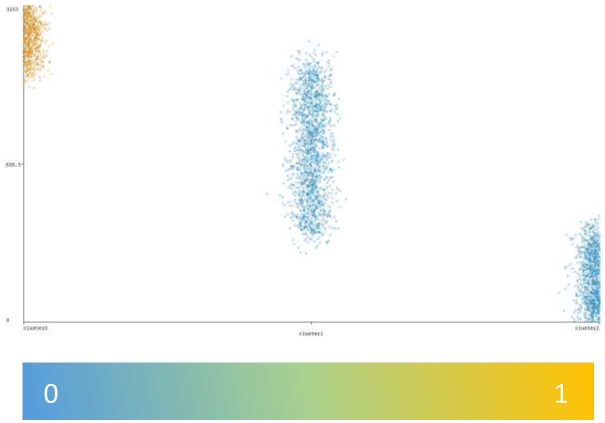


Figure 25: score distribution in the **Beverage category** in the 3 clusters.

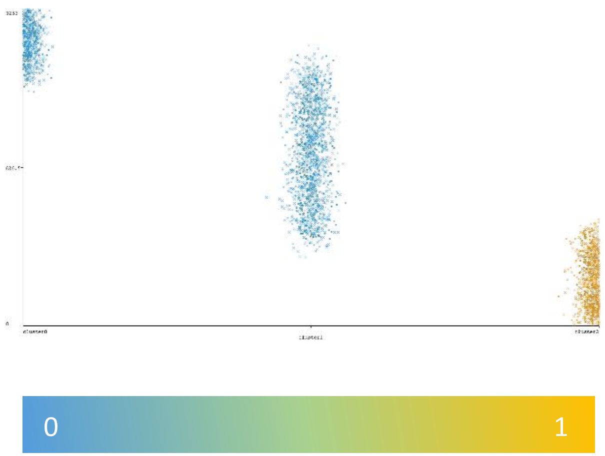


Figure 26: score distribution in the **Bakery category** in the 3 clusters.

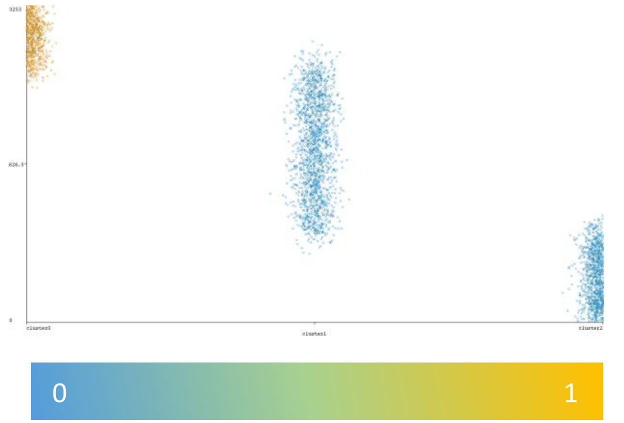


Figure 27: score distribution in the **Canned Foods category** in the 3 clusters.

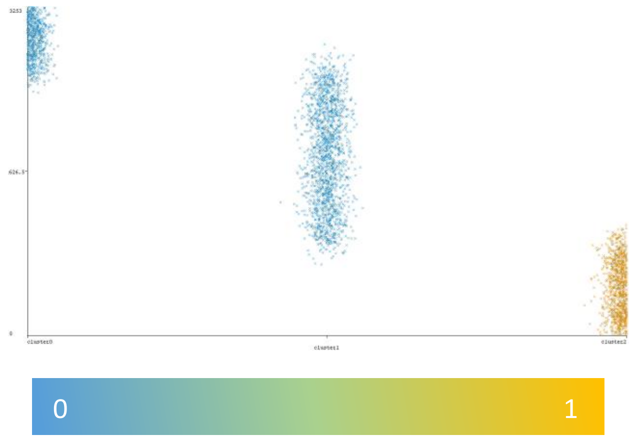


Figure 28: score distribution in the **Dry Foods category** in the 3 clusters.

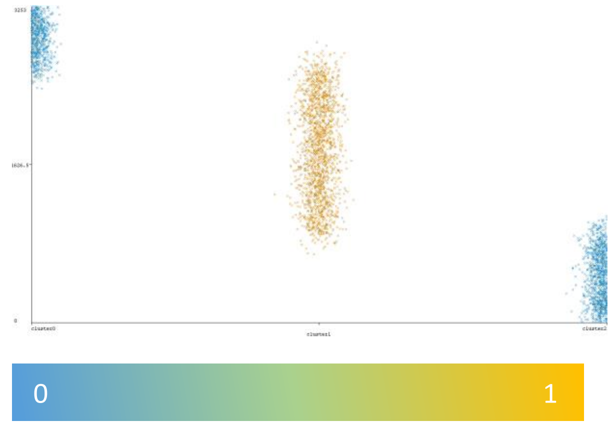


Figure 29: score distribution in the **Frozen Foods category** in the 3 clusters.

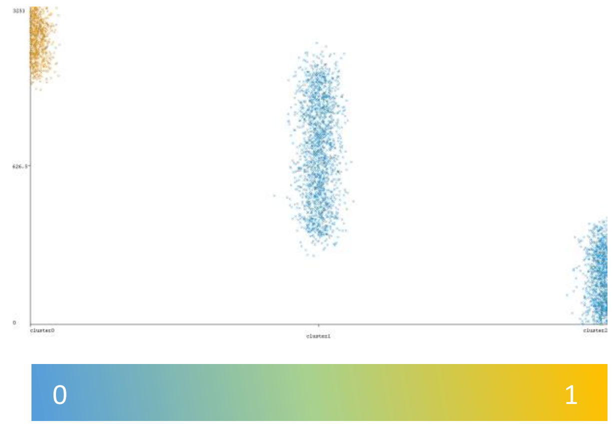


Figure 30: score distribution in the **Meat category** in the 3 clusters.

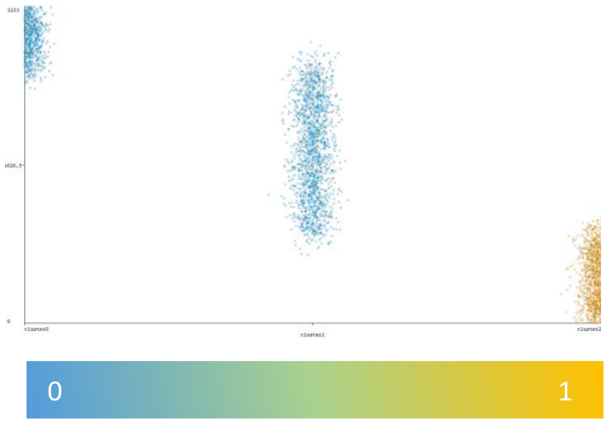


Figure 31: score distribution in the **Produce category** in the 3 clusters.

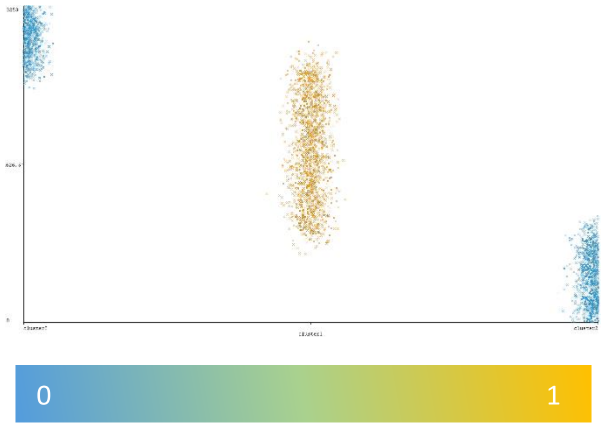


Figure 32: score distribution in the **Cleaners category** in the 3 clusters.

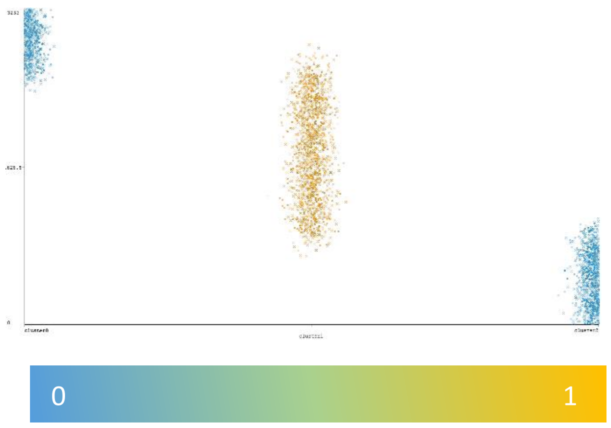


Figure 33: score distribution in the **Others category** in the 3 clusters.

From these graphs we can deduce the following final **conclusions** for each cluster:

1) *the first cluster* collects the **youngest** and **wealthiest clients** with **low-medium expenses**, of any social status, gender and job category. They seem to prefer categories like *beverages*, *canned foods* and *meat*.
2) *the second cluster* collects **mass** and **middle-aged customers** with **medium-high expenses**, of any social status, gender and job category. They seem to prefer categories like *frozen foods*, *cleaners*, and *others*.
3) *the third cluster* collects the **richest** and most **senior customers** with **high spending**, of any social status, gender and job category. They seem to prefer *bakery*, *dry foods* and *produce*.

DBSCAN

Problem: how to decide the $minPoints$ and eps parameters?

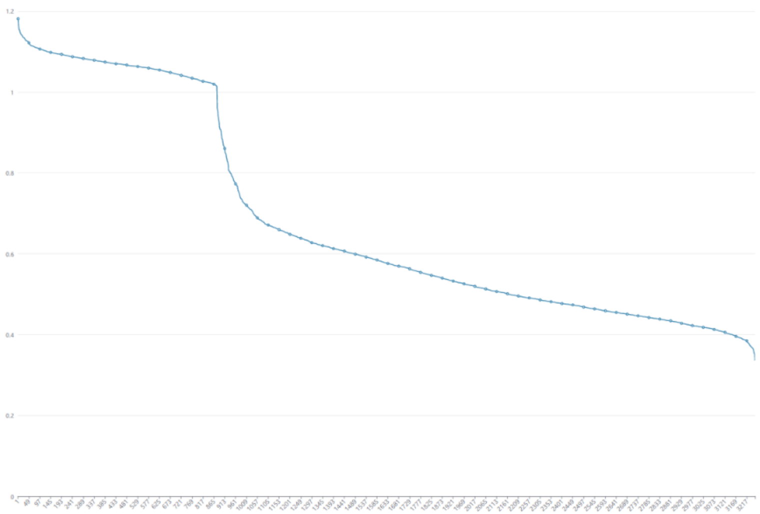
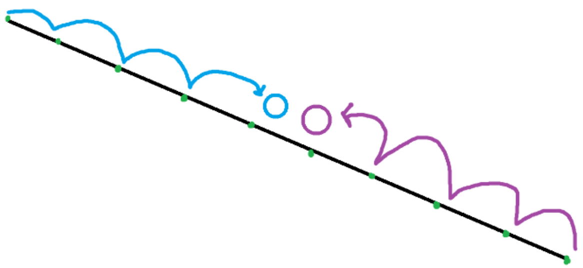


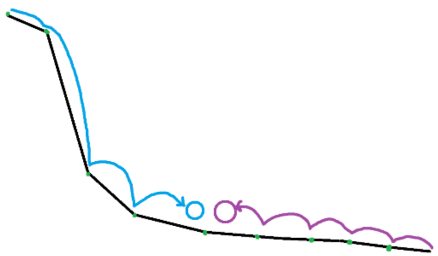
Figure 34: curve of nearest k-th instance of each instance.

- The curve was obtained by calculating for each instance the distance with the nearest k-th and ordering the points in descending order by this distance.
- eps corresponds to that y = eps for which the valley is encountered. This value is difficult to calculate.

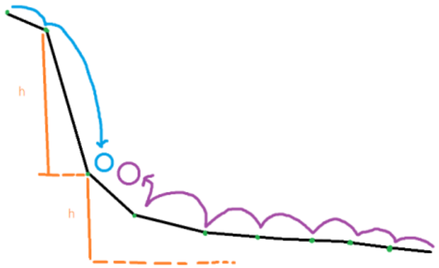
I have called my heuristic algorithm **EquitySlope**. Simple intuition:



If the blue and purple circle move alternately from one point to the next neighbor (i.e. from one x-axis unit to the next) then they will meet in the middle.



The same thing happens if the path is distorted. The blue circle will travel further, and therefore accelerates to always reach the center point "in time".



Let's change philosophy. If instead of moving the two circles *unit by unit of the x-axis*, we now move them *unit by unit of the y-axis*. In this way the purple circle will no longer move from a point to the one near it, but to the one that will allow it to "go up" as much as the blue circle has "gone down".

The algorithm returns a good approximation of eps with a complexity O (n). Ideally, iterate DBSCAN by varying eps in a range centered in the found value. Then setting:

$$minPoints = 2 * Dimension$$

we get the cluster in figure with a *silhouette index of 0.399* . DBSCAN performs well, thanks to the good density of the regions as shown in figure Y, but it takes too long, so it will be discarded.

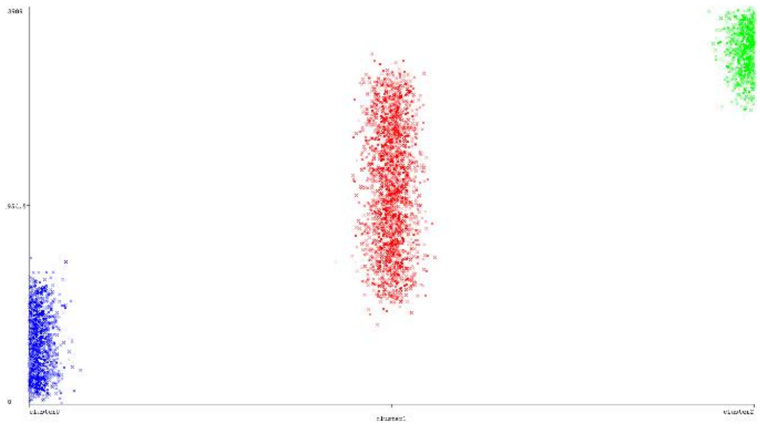


Figure 35: clustering with DBSCAN.

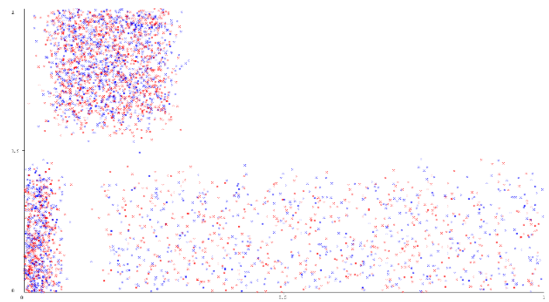


Figure 36: two-dimensional visualization of AnnualExpenses and Cleaners.

AGNES

- Similarly as done with K-Means we move k from 2 to 10 and taking advantage of my implementation of the silhouette index (since Weka does not have a package similar to KValid) we will find once again that the best k is 3.
  The type of linkage will also be changed.

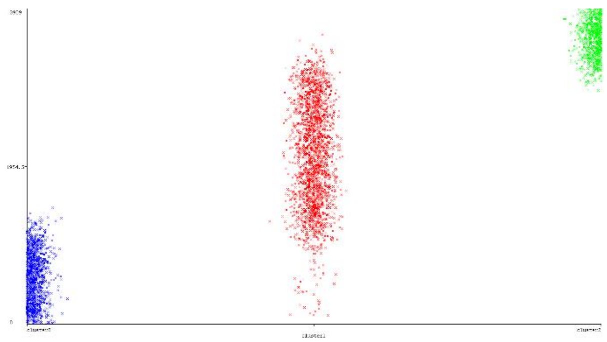
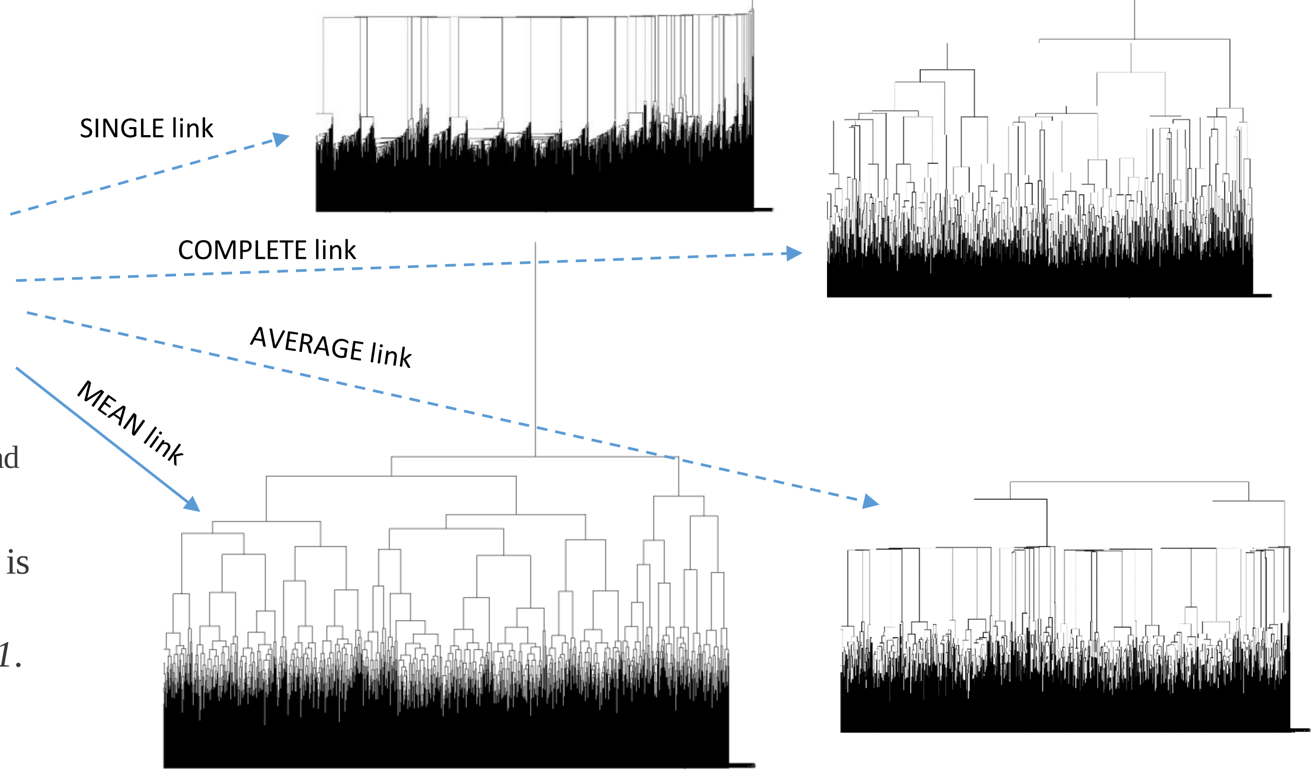


Figure 37: hierarchical clustering with k=3 and MEAN as link type.



The dendogram with MEAN as link type is clearer and less **skewed** towards one side and have a *silhouette index equal to 0.401*. But the single execution of the algorithm **takes too long**.

The algorithm that the application will use will therefore be K-Means (with KValid).

Functional & Non-Functional requirements and UML diagrams.

Functional

- suggests the most frequent product combinations purchased during the week.
- calculate the current customer segmentation.

Non-Functional

- heavy computational must be processed in background.
- intuitive and simple to use.
- the data in the two datasets that the application use must have a fixed and non-changing structure.
- the application uses only FP-Growth as the association rule mining algorithm, and only K-Means as the clustering mining.

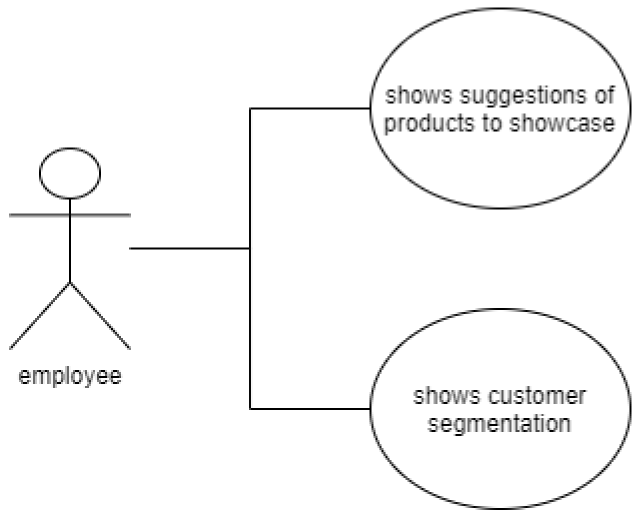


Figure 38: use case diagram.

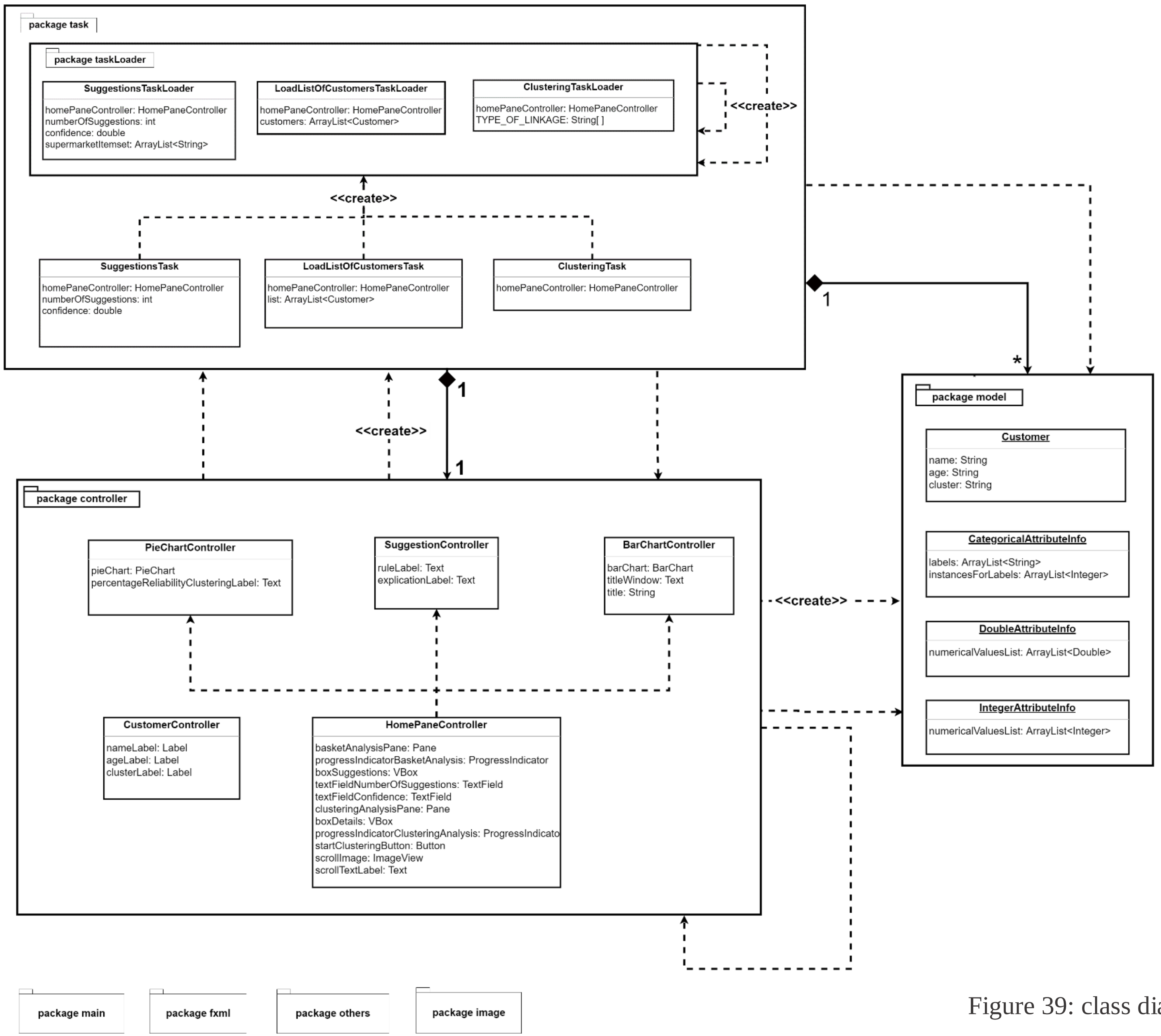


Figure 39: class diagram.

User interface