3 Design and Application of Market Basket Analysis Methodology

3.1 Project Methodology

Through previous chapters, most aspect of the project were already described. However, a brief summary of the project process will be described following to have a general view of the procedure. In addition, there are some details that has to be mentioned about the design of the project that affected its procedure.

Our objective in his project was the analysis of customers purchases and its behaviour. To do it, the project was divided in two steps. The first one, was a store clustering. Group stores based on its behaviours. The second one, was the analysis of associations rules of its items for each cluster.

The first step we realized was the problem definition. Which thing the client wanted to achieve with this project. All this step was described in the "*Introduction*" chapter.

The second step was the obtention of data. Companies usually have its data in data warehouse [21] or databases [22] and the extraction of it is a difficult task that requires a huge work. In order to design a good data science project, automatic workflows of that extraction has to be done. That's because, as it was mentioned previously, machine learning models have to be retrained periodically. However, for the first model of associations rules, the client was not interested in this automatic workflow. Due that, all the data used in this project was given in a *csv* format.

Once we obtained the data we had to clean it. Throughout the project we removed data that was incorrect or invalid. Is common that data have mistakes, is impossible to have everything in order in a company, that's why a data cleaning task is performed. However, there are cases where some instances have to be removed although are correct because they are considered anomalies. Anomaly detection [23] is the identification of observations which do not conform to an expected pattern or other items in a dataset. This concept has not to be confused with data cleaning, due the cleaning data process search for invalid records. Anomalies detection just looks for records that not form part of a pattern. Depending the objective of the project, this anomalies can be noise or be exactly what you are looking for. For instance, in fraud detection problems, those instance that don't conform a regular pattern are possible fraudulent transactions [24].

In our project, the store with id *614* had *2500* m^2 when the second shop with more meters was the *525* with *1400* m^2. The store *614* was so big due the client consider as a single store an entire commercial center and rents parts of the area to different business. This traduced to patterns, means an anomaly store. Our issue was to decide if we removed this store from the clustering process. At the end, we decided to kept it due the client was interested to know in which cluster the clustering algorithm classified it. In addition, in the same process of data cleaning, we decided to remove plastics bags and parking records from the tickets historical due they didn't add valuable information to the project.

In the clustering step we had to choose between two different clustering algorithms that BigML has, K-means and G-means. Each of them has its own characteristics. In the "Clustering Models" chapters both are described. At the beginning of the project we considered to use both algorithm and analyze its results, however, due the client didn't have in mind a specific number of clusters to find, we used G-means in this project. The only condition was that the numbers of clusters was acceptable for its purposes. A low number of clusters will be useless for the client cause he couldn't discover remarkable differences between stores of each cluster, and a high number of cluster would be impossible for the client to apply a specific marketing strategy for each of them. For instance, if a cluster were composed for 4 shops had no rentability study and apply a strategy just for these stores. We decided to improve the data representation until G-means algorithm output 9 clusters.

In order to evaluate the clustering models we were obtaining, we were in periodically contact with the client. In classification problems, there are many techniques to analyse the performance of a model. However, in a clustering problem, the evaluation of it is more complicate, due there is no target to predict and compare results. Thus, to evaluate our clustering model, we were in constant contact with the client presenting the results we were obtaining. Moreover, at the end of the project, we presented to the client the association rules created for each cluster.

3.2 Software & Hardware used

As BigML is the one responsible of all the machine learning algorithms all this process was done on its servers, we didn't need hardware for it. However, for the feature engineering task, we needed it.

The software used in this project was the programming language Python. In addition, we used one of the most powerful libraries used nowadays for Data Science, Pandas. The environment to programme used was Jupyter Notebook [25].

To process all the data we used a Windows server. We upload there all the data needed and used via remote, the software previously mentioned to realize the feature engineering. Moreover, we had a personal laptop to analyse data, results and connect to BigML.

Server

- Windows Edition: Windows Server 2012 R2 Standard
- Processor: Intel(R) Xeon(R) CPU E5-2047 v2 @ 2.40GHz 2.40 GHz
- Installed memory (RAM): 24.00 GB
- System type: 64-bit Operating System, x64 based processor

Laptop

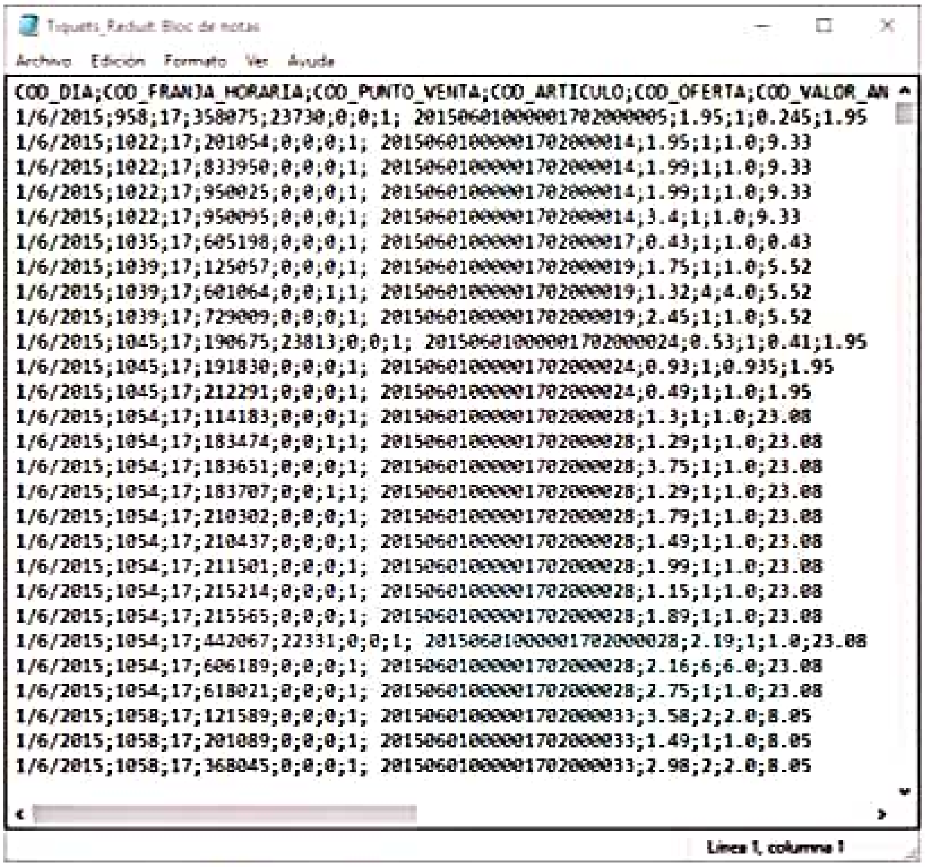
- Windows Edition: Windows 10 Enterprise
- Processor: Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz 2.40GHz
- Installed memory (RAM): 8.00GB
- System type: 64-bit Operating System, x64 based processor

3.3 Data

Through time, we have faced different projects on CleverData. Always that we start a new project, on the first meeting with the client, we perform the same question, which data posses. This question is maybe one of the most important ones. Depending the answer, we can know instantly if the project can be performed or not. The key in any data science project is data. Data is the principal component that makes a projects success or fail. Is the machine learning algorithms combustible. With no data, we cannot make miracles. However, if we possess the correct data, we can start to work.

One question we often listen from our clients is the quantity of data they need to start to using machine learning. They collect data like if they have diogenes syndrome. Influenced by the famous concept of Big Data. However, to use machine learning is not needed a huge amount of data. That's one advice we repeat constantly to our clients in CleverData. More than the quantity of data, the important is the quality of it. Without quality, the algorithm cannot learn any pattern from data, there is just a lot of nonsense data and noise. For instance, we face some projects where we have a lot of data. However, when we start to look the given data trying to understand what means we discover inconsistencies on it. Maybe some variables are not well calculated or others has no sense. All this "mistakes" influence negatively the model. In our case, we found some mistakes in the data during the project, however, nothing special. In addition, with the huge amount of data the client provided us, it was not a problem remove those cases from data.

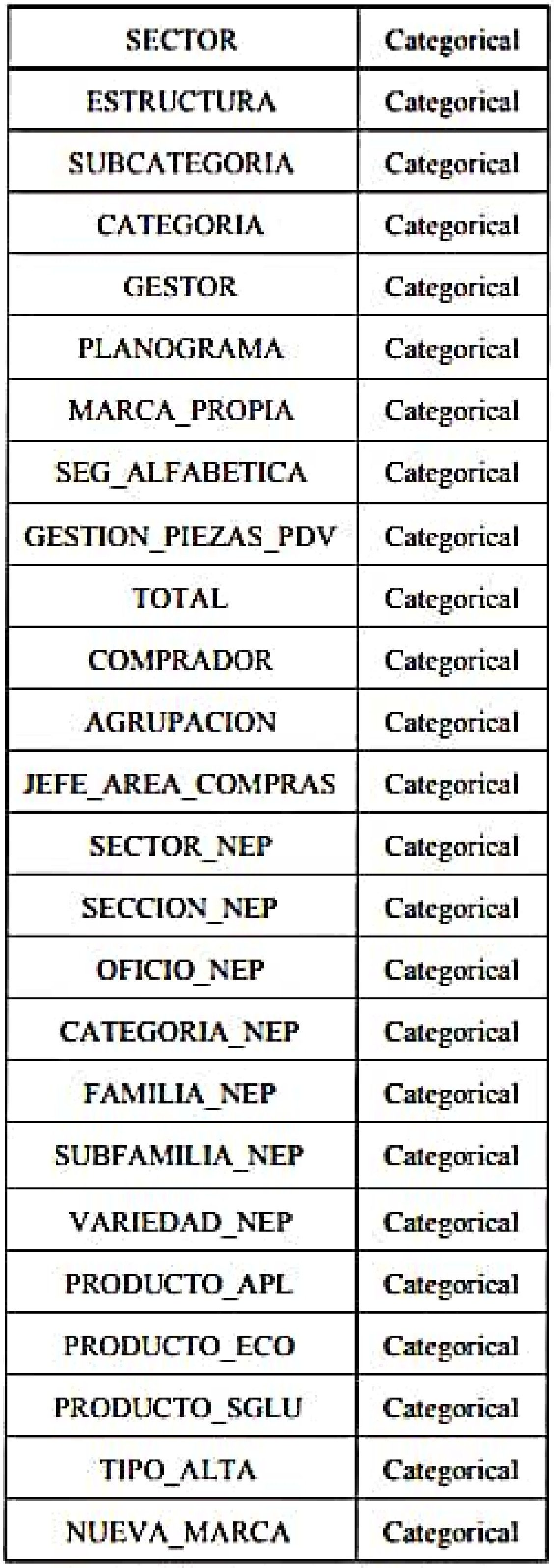
The client provided us three different datasets. Each of them was a *csv* file. The first one was the list of historical tickets. The second one was the stock of items. The last one was the list


Tiquets_Reduit: Bloc de notas
Archivo Edición Formato Ver Ayuda
Línea 1, columna 1

```
COD_DIA;COD_FRANJA_HORARIA;COD_PUNTO_VENTA;COD_ARTICULO;COD_OFERTA;COD_VALOR_AN
1/6/2015;958;17;358075;23730;0;0;1; 20150601000001702000005;1.95;1;0.245;1.95
1/6/2015;1022;17;201054;0;0;0;1; 20150601000001702000014;1.95;1;1.0;9.33
1/6/2015;1022;17;833950;0;0;0;1; 20150601000001702000014;1.99;1;1.0;9.33
1/6/2015;1022;17;950025;0;0;0;1; 20150601000001702000014;1.99;1;1.0;9.33
1/6/2015;1022;17;950095;0;0;0;1; 20150601000001702000014;3.4;1;1.0;9.33
1/6/2015;1035;17;605198;0;0;0;1; 20150601000001702000017;0.43;1;1.0;0.43
1/6/2015;1039;17;125057;0;0;0;1; 20150601000001702000019;1.75;1;1.0;5.52
1/6/2015;1039;17;601064;0;0;1;1; 20150601000001702000019;1.32;4;4.0;5.52
1/6/2015;1039;17;729009;0;0;0;1; 20150601000001702000019;2.45;1;1.0;5.52
1/6/2015;1045;17;190675;23813;0;0;1; 20150601000001702000024;0.53;1;0.41;1.95
1/6/2015;1045;17;191830;0;0;0;1; 20150601000001702000024;0.93;1;0.935;1.95
1/6/2015;1045;17;212291;0;0;0;1; 20150601000001702000024;0.49;1;1.0;1.95
1/6/2015;1054;17;114183;0;0;0;1; 20150601000001702000028;1.3;1;1.0;23.08
1/6/2015;1054;17;183474;0;0;1;1; 20150601000001702000028;1.29;1;1.0;23.08
1/6/2015;1054;17;183651;0;0;0;1; 20150601000001702000028;3.75;1;1.0;23.08
1/6/2015;1054;17;183707;0;0;1;1; 20150601000001702000028;1.29;1;1.0;23.08
1/6/2015;1054;17;210302;0;0;0;1; 20150601000001702000028;1.79;1;1.0;23.08
1/6/2015;1054;17;210437;0;0;0;1; 20150601000001702000028;1.49;1;1.0;23.08
1/6/2015;1054;17;211501;0;0;0;1; 20150601000001702000028;1.99;1;1.0;23.08
1/6/2015;1054;17;215214;0;0;0;1; 20150601000001702000028;1.15;1;1.0;23.08
1/6/2015;1054;17;215565;0;0;0;1; 20150601000001702000028;1.89;1;1.0;23.08
1/6/2015;1054;17;442067;22331;0;0;1; 20150601000001702000028;2.19;1;1.0;23.08
1/6/2015;1054;17;606189;0;0;0;1; 20150601000001702000028;2.16;6;6.0;23.08
1/6/2015;1054;17;618021;0;0;0;1; 20150601000001702000028;2.75;1;1.0;23.08
1/6/2015;1058;17;121589;0;0;0;1; 20150601000001702000033;3.58;2;2.0;8.05
1/6/2015;1058;17;201089;0;0;0;1; 20150601000001702000033;1.49;1;1.0;8.05
1/6/2015;1058;17;368045;0;0;0;1; 20150601000001702000033;2.98;2;2.0;8.05
```

Figure 3.1: Tickets dataset screenshot.

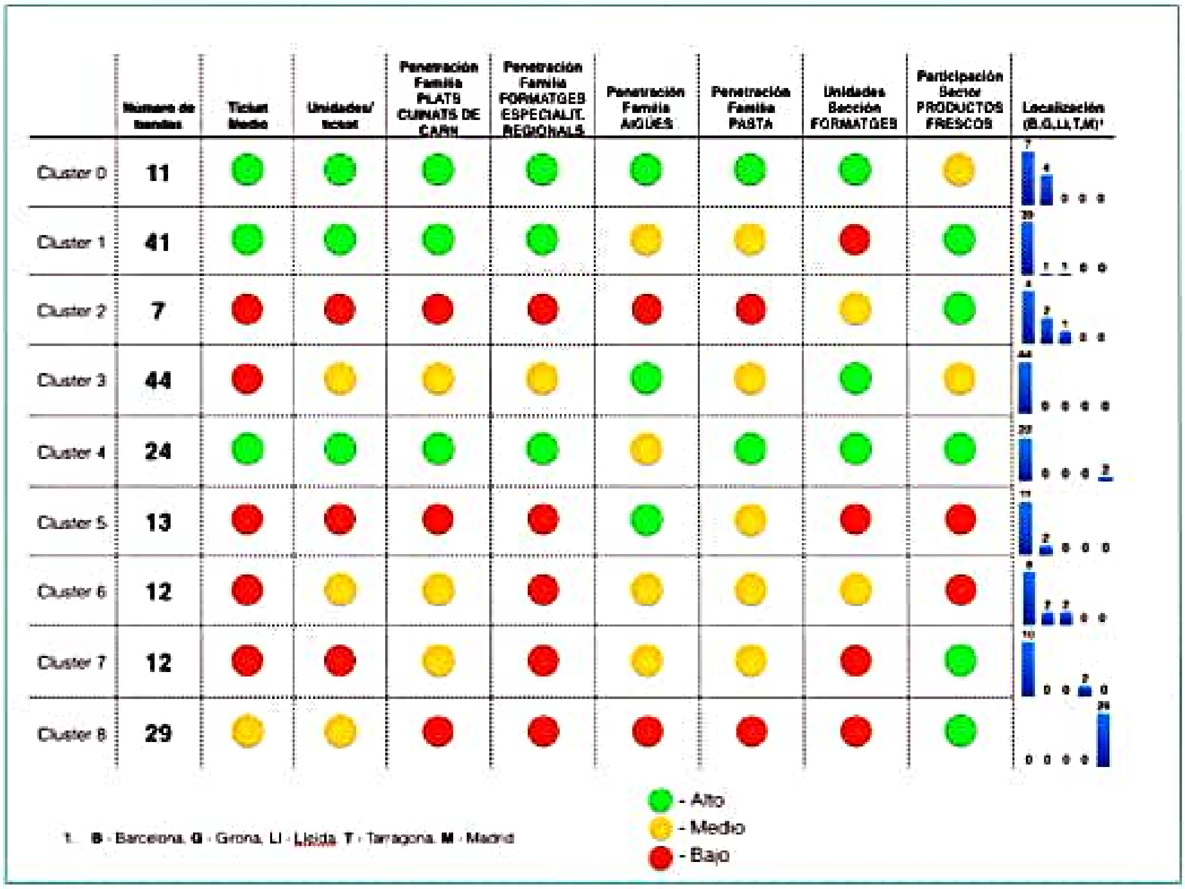
Feature	Type
COD_DIA	Date
COD_FRANJA_HORARIA	Categorical
COD_PUNTO_VENTA	Categorical
COD_ARTICULO	Categorical
COD_OFERTA	Categorical
COD_VALOR_ANIADIDO	Categorical
COD_MARCA_PROP	Categorical
COD_TIPO_LINEA	Categorical
COD_TICKET	Categorical
IMPORTE_PVP	Numerical



SECTOR	Categorical
ESTRUCTURA	Categorical
SUBCATEGORIA	Categorical
CATEGORIA	Categorical
GESTOR	Categorical
PLANOGRAMA	Categorical
MARCA_PROPIA	Categorical
SEG_ALFABETICA	Categorical
GESTION_PIEZAS_PDV	Categorical
TOTAL	Categorical
COMPRADOR	Categorical
AGRUPACION	Categorical
JEFE_AREA_COMPRAS	Categorical
SECTOR_NEP	Categorical
SECCION_NEP	Categorical
OFICIO_NEP	Categorical
CATEGORIA_NEP	Categorical
FAMILIA_NEP	Categorical
SUBFAMILIA_NEP	Categorical
VARIEDAD_NEP	Categorical
PRODUCTO_APL	Categorical
PRODUCTO_ECO	Categorical
PRODUCTO_SGLU	Categorical
TIPO_ALTA	Categorical
NUEVA_MARCA	Categorical

Figure 3.3: Articulos dataset features.

For people from machine learning world, is easy to explain that what defines a cluster is the sum of the different features, however, for person from marketing that's another story, they need something that tells them what has a cluster in particular. So, in order to have this little help that described what had each cluster in particular we created the brief tables summary (Figures 3.12 and 3.13).

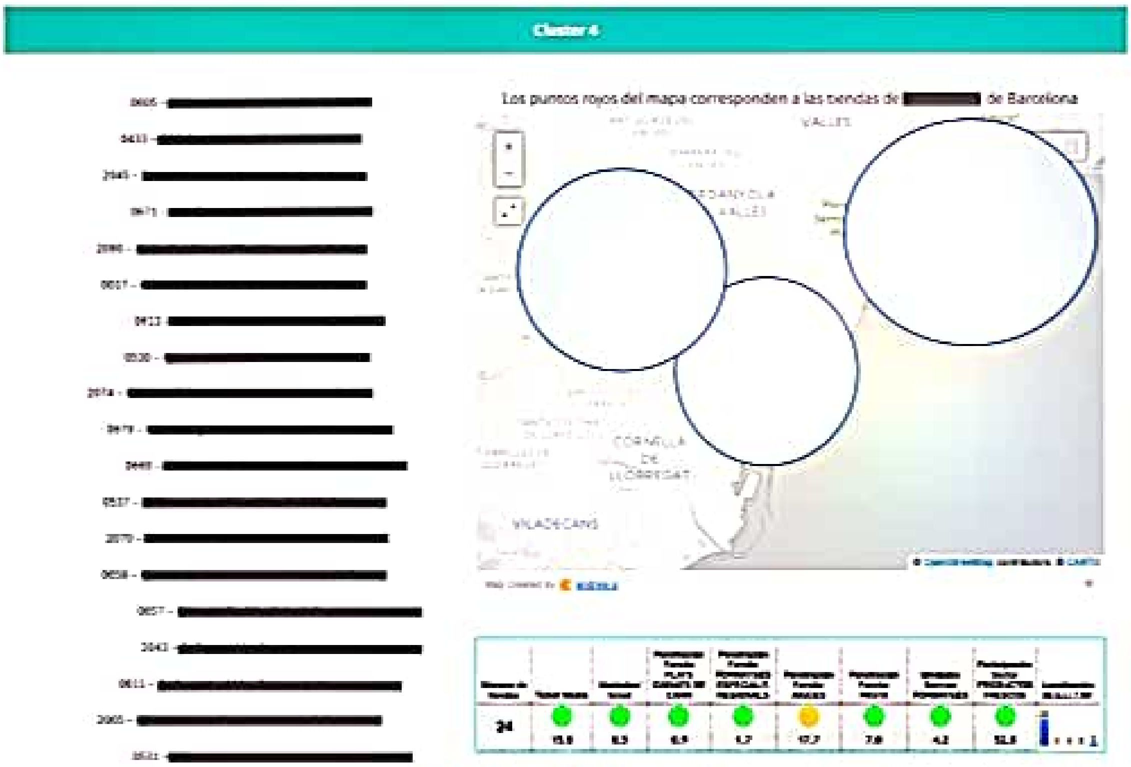


	Número de tiendas	Ticket Medio	Unidades/ ticket	Penetración Familia PLATS CUINATS DE CARN	Penetración Familia FORMATGES ESPECIALIT. REGIONALS	Penetración Familia AIGÜES	Penetración Familia PASTA	Unidades Sección FORMATGES	Participación Sector PRODUCTOS FRESCOS	Localización (B,G,Ll,T,M)¹
Cluster 0	11	🟢	🟢	🟢	🟢	🟢	🟢	🟢	🟡	7 4 0 0 0
Cluster 1	41	🟢	🟢	🟢	🟢	🟡	🟡	🔴	🟢	39 1 1 0 0
Cluster 2	7	🔴	🔴	🔴	🔴	🔴	🔴	🟡	🟢	4 2 1 0 0
Cluster 3	44	🔴	🟡	🟡	🟡	🟢	🟡	🟢	🟡	44 0 0 0 0
Cluster 4	24	🟢	🟢	🟢	🟢	🟡	🟢	🟢	🟢	22 0 0 0 2
Cluster 5	13	🔴	🔴	🔴	🔴	🟢	🟡	🔴	🔴	11 2 0 0 0
Cluster 6	12	🔴	🟡	🟡	🔴	🟡	🟡	🟡	🔴	8 2 2 0 0
Cluster 7	12	🔴	🔴	🟡	🔴	🟡	🟡	🔴	🟢	10 0 0 2 0
Cluster 8	29	🟡	🟡	🔴	🔴	🔴	🔴	🔴	🟢	0 0 0 0 29

🟢 - Alto
🟡 - Medio
🔴 - Bajo

1. **B** - Barcelona, **G** - Girona, Ll - Lleida, **T** - Tarragona, **M** - Madrid

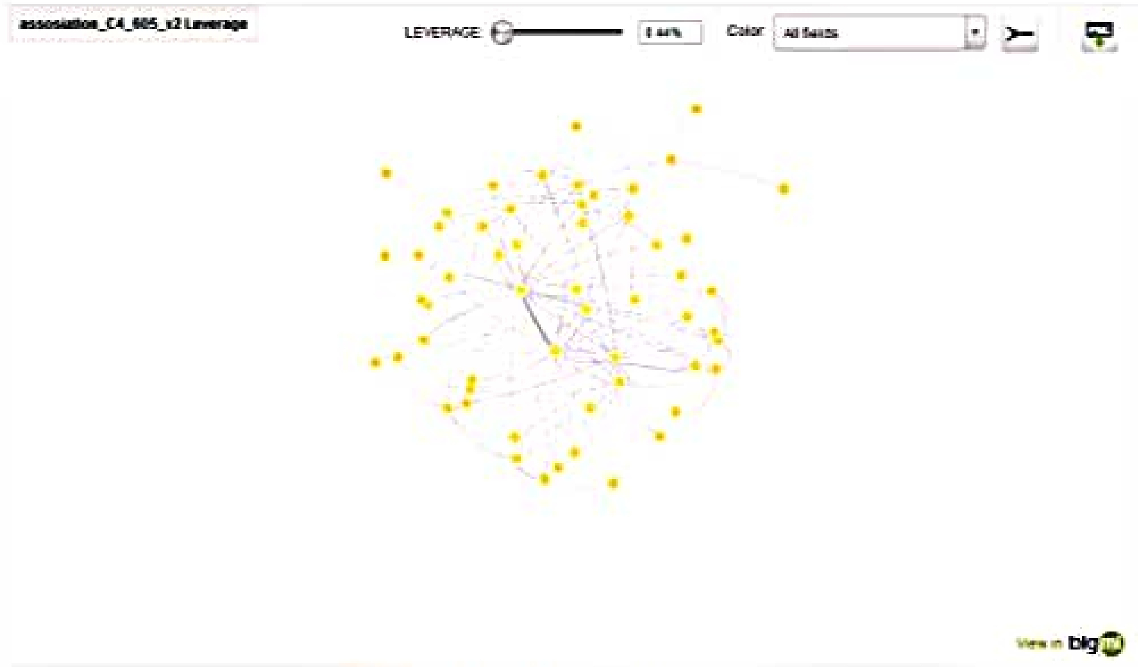
Figure 3.12: Table summary.



Cluster 4
Los puntos rojos del mapa corresponden a las tiendas de [redacted] de Barcelona
[redacted]
CORNELLA DE LLOBREGAT
VILADECANS
24
15.8
8.5
6.9
4.7
17.7
7.8
4.2
52.8

Figure 3.15: Cluster 4 Web.

On the second page of the web, the client could saw the results according the associations rules. Both associations rules based on lift and leverage strategies were plotted. Using the BigML API, we were able to create a widget that communicated with our BigML account and plotted the corresponding associations, with that, the client was able to analyze the results in a dynamic way. Both associations are the ones of the figures in the previous section of this memory. In addition, we added two relationship diagrams for the leverage and lift rules (Figures 3.16 and 3.17).

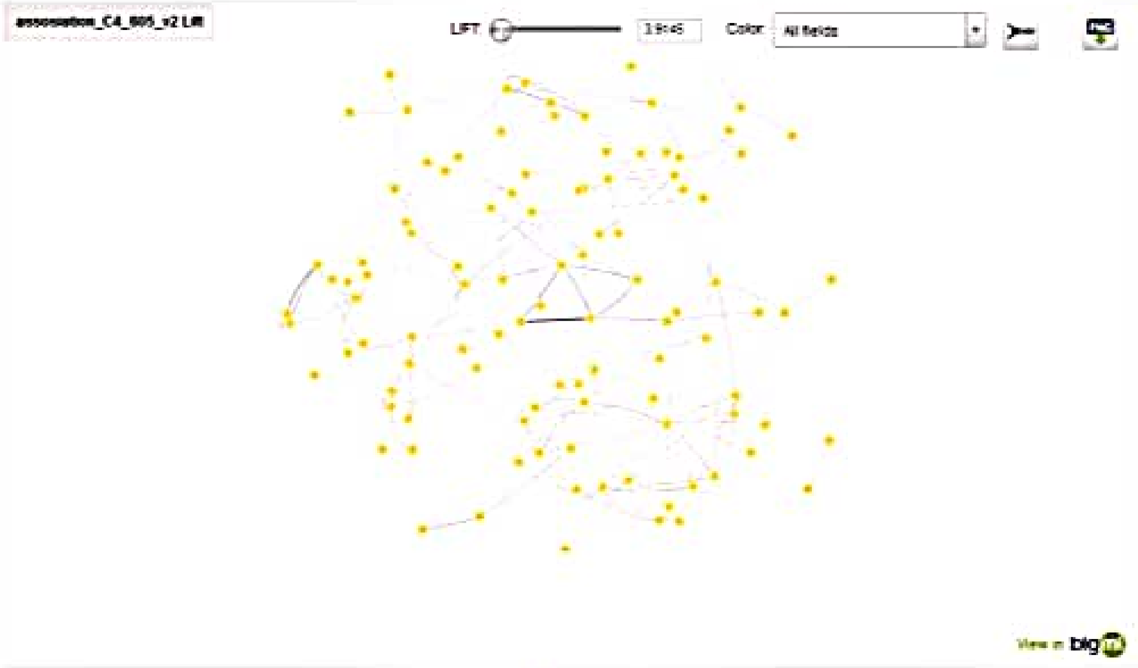


assosiation_C4_605_v2 Leverage

LEVERAGE: 0.44%

Color: All fields

Figure 3.16: Leverage Diagram.



assosiation_C4_605_v2 Lift

LIFT: 1.9146

Color: All fields

Figure 3.17: Lift Diagram.

To decide which associations were interesting we let the client choose between them. One interesting point with the projects we realize, is that once the results are obtained and analyzed, the corresponding action has to be taken using the knowledge of the client. At the end, the one who knows better the company is the client itself, and he has to be the one who

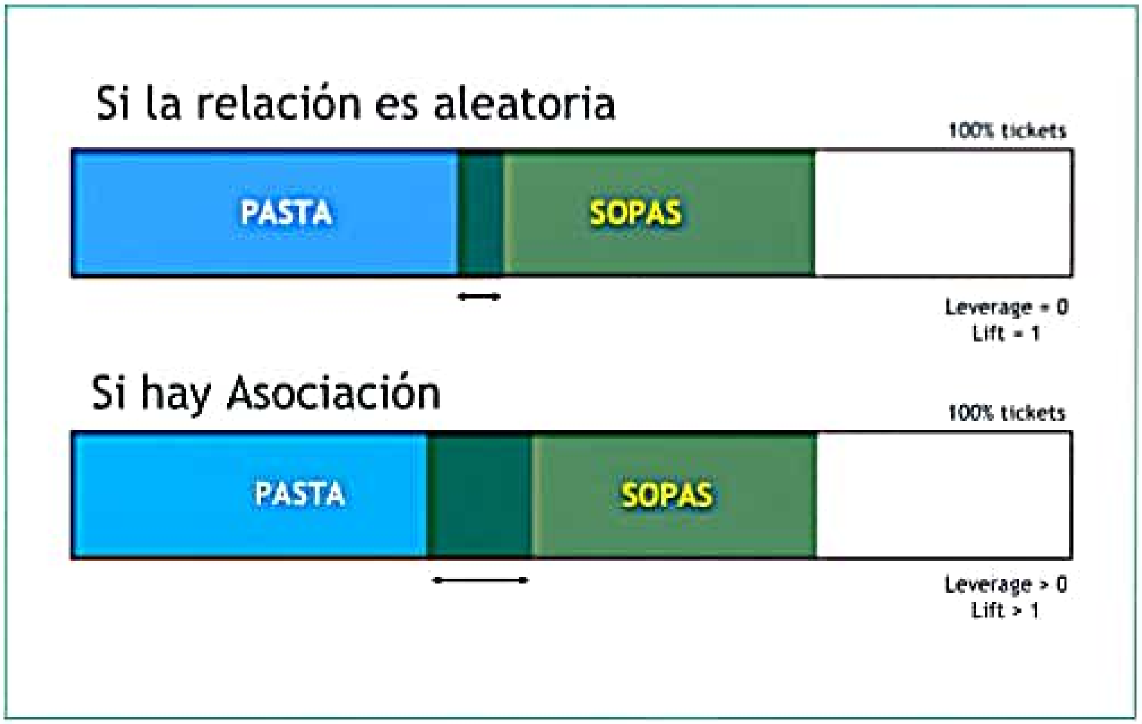


Figure 3.19: Description of rules.

The last page consisted in a dynamic scatterplot. As we did with the associations, this widget communicated with our BigML account, in concrete, the scatterplot tool that has BigML. With this, the client was able to visualize different variables and how they were correlated. In the scatterplot each point is a shop and each color a cluster. Some examples of scatterplots that the client could visualize are the following: ticket mean price (Figure 3.20), the region (Figure 3.21) and mean units per ticket (Figure 3.22)

With the files delivered and the web constructed, the project was considered concluded.

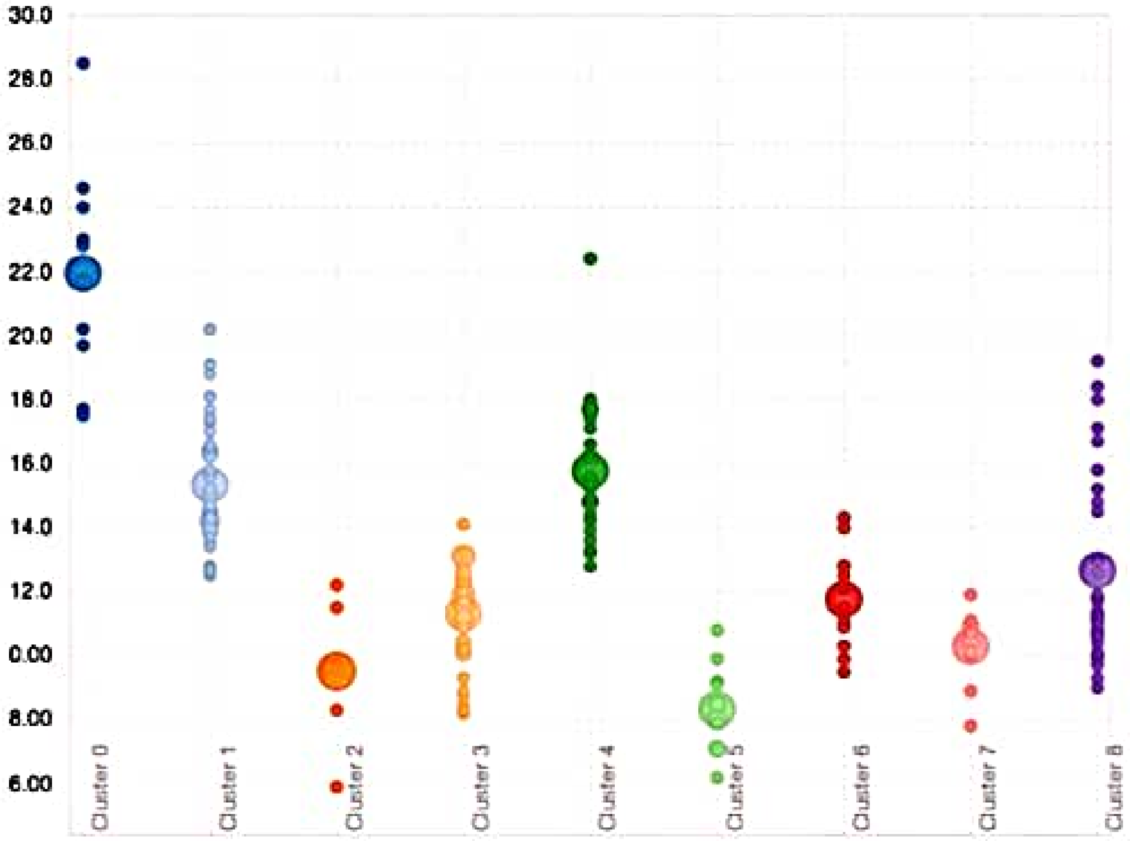


Figure 3.20: Mean price ticket scatterplot.