Data: Online Retail

<https://www.kaggle.com/carrie1/ecommerce-data>

<https://archive.ics.uci.edu/ml/datasets/online+retail>

Code: <https://www.kaggle.com/fabiendaniel/customer-segmentation/notebook>

Dataset

E-commerce database: lists purchases made by ∼4000 customers over a period of one year (from 2010/12/01 to 2011/12/09)

Data shape: (541909, 8)

* InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
* StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
* Description: Product (item) name. Nominal.
* Quantity: The quantities of each product (item) per transaction. Numeric.
* InvoiceDate: Invice Date and time. Numeric, the day and time when each transaction was generated.
* UnitPrice: Unit price. Numeric, Product price per unit in sterling.
* CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
* Country: Country name. Nominal, the name of the country where each customer resides.

**1. Data Preparation**

Remove:

1. ~25% of the entries are not assigned to a particular customer (description and customer id)
2. duplicate entries

**2. Exploring the content of variables**

2.1 Countries

2.2 Customers and products

the existence of users who only came once and only purchased one product

the existence of frequent users that buy a large number of items at each order

2.2.1 Cancelling orders: 3654/22190 (16.47%)

when an order is canceled, we have another transaction in the dataframe, mostly identical except for the **Quantity** and **InvoiceDate** variables.

To check this, we locate the entries that indicate a negative quantity and check if there is *systematically* an order indicating the same quantity (but positive), with the same description (**CustomerID**, **Description** and **UnitPrice**)

checked the two cases:

* a cancel order exists without counterpart: 7521
* there's at least one counterpart with the exact same quantity: 1226

2.2.2 StockCode

2.2.3 Basket price

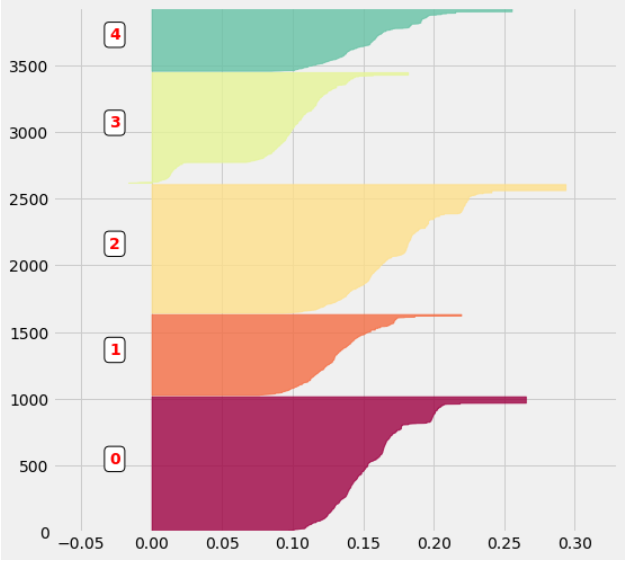
**3. Insight on product categories**

* 3.1 Product description

word occurence

* 3.2 Defining product categories
  + 3.2.1 Data encoding: use these keywords to create groups of product
  + 3.2.2 Clusters of products: kmeans (5 clusters)
  + 3.2.3 Characterizing the content of clusters

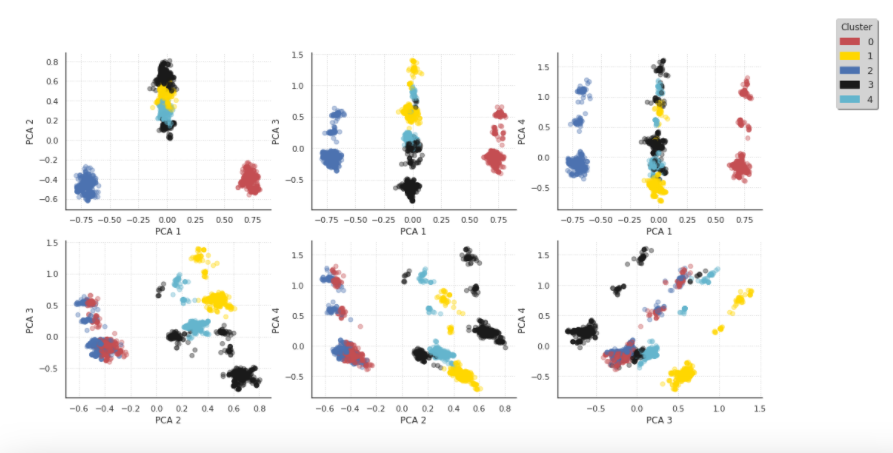
**a / *Silhouette intra-cluster score***

******

***b/ Word Cloud***

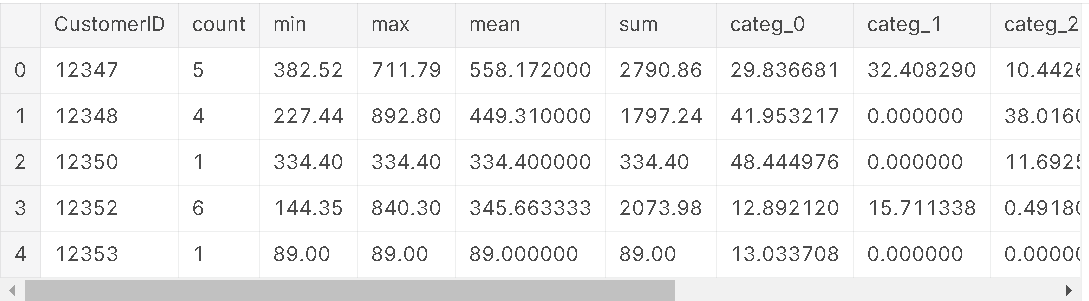
******

***c / Principal Component Analysis***

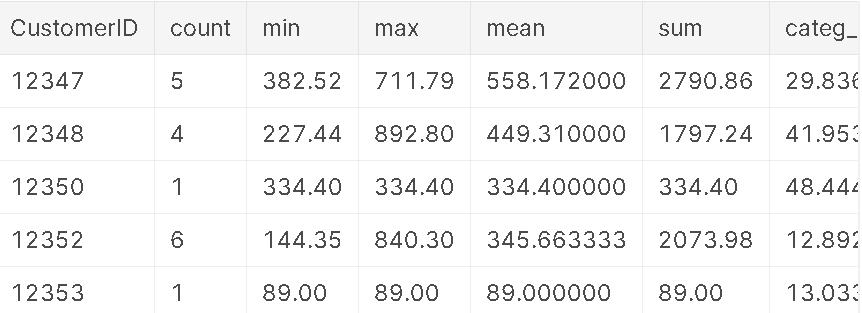
******

**4. Customer categories**

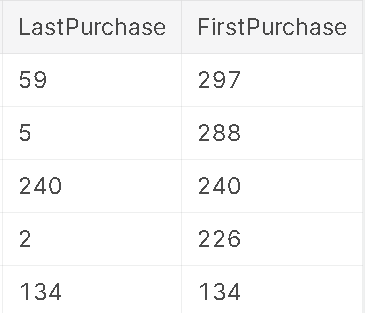
* 4.1 Formating data
  + 4.1.1 Grouping products



* + 4.1.2 Time spliting of the dataset: 10 months first for train and the rest for test
  + 4.1.3 Grouping orders:



Create 2 columns:



Customers make only one purchase: 1445/3608 (40.05%)

* 4.2 Creating customer categories
  + 4.2.1 Data enconding

list\_cols = ['count','min','max','mean','categ\_0','categ\_1','categ\_2','categ\_3','categ\_4']

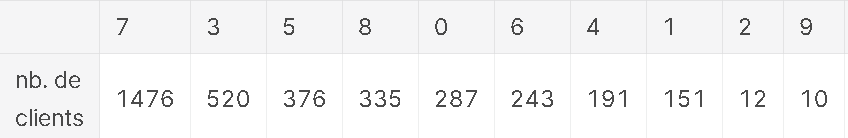
Create a matrix where these data are standardized:

[ 3.62305987 259.93189634 556.26687999 377.06036244 25.22916919 16.37327913 13.98907929 28.73795868 15.67936332]

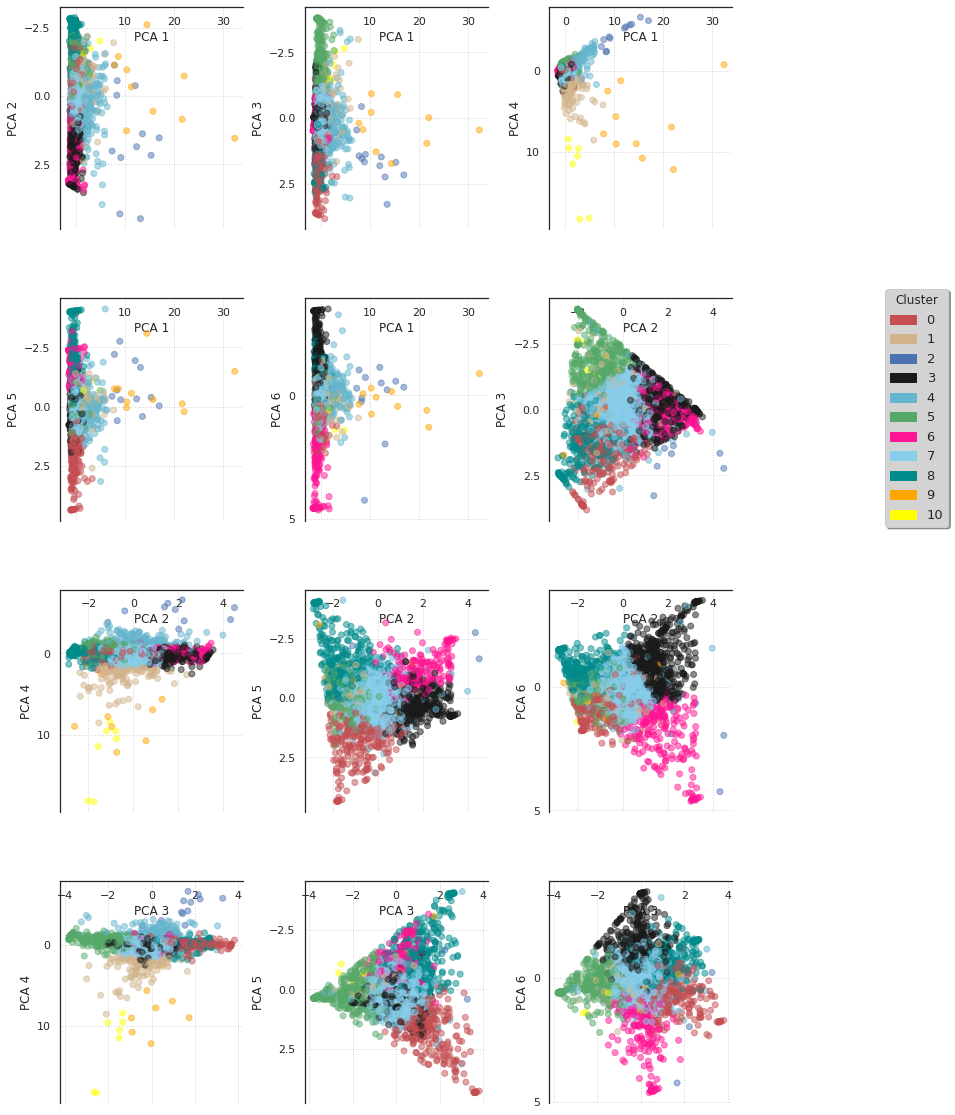
Create clusters of customers

* + 4.2.2 Creation of customer categories: kmeans 11 clusters

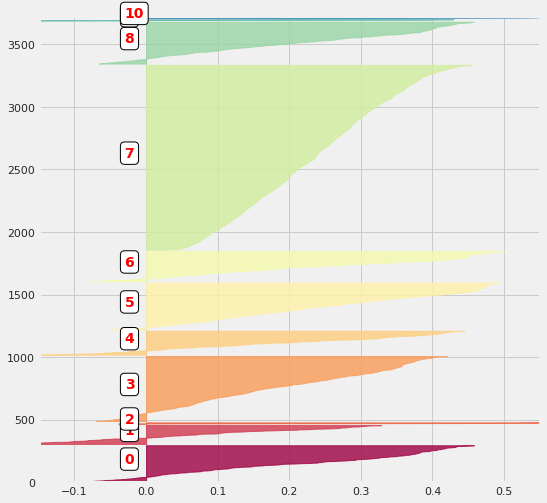
The number of customers in each cluster:



**a / *Report via the PCA***

******

b/ silouhette scores



**c/ *Customers morphotype***

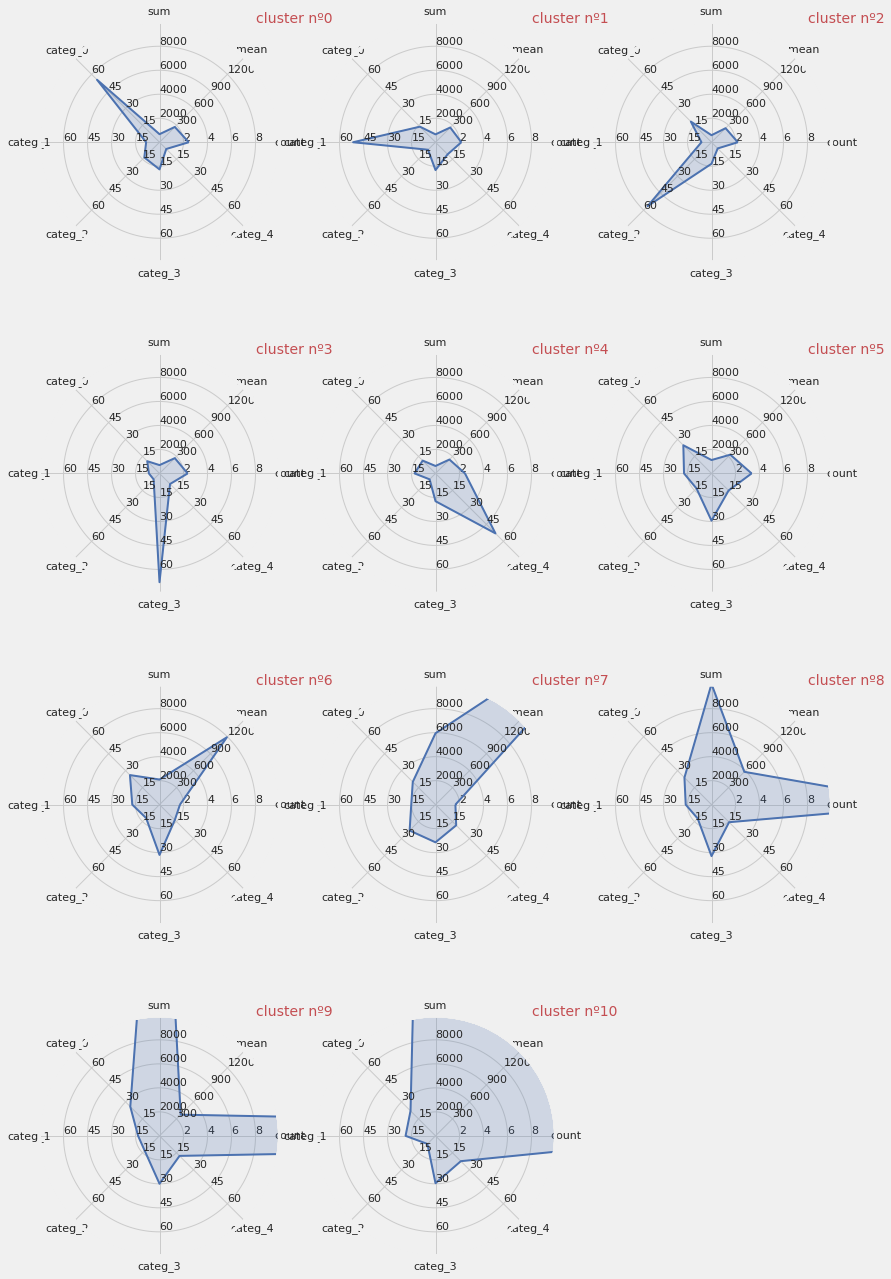
the different clusters are indeed disjoint

adding to the selected\_customers dataframe a variable that defines the cluster to which each client belongs

average the contents of this dataframe by first selecting the different groups of clients

re-organize the content of the dataframe by ordering the different clusters

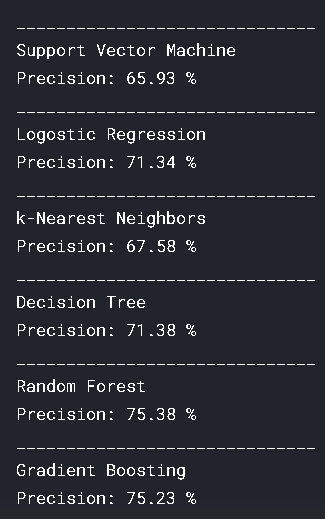
**d / *Customers morphology***

******

**5. Classifying customers**

* 5.1 Support Vector Machine Classifier (SVC)
  + 5.1.1 Confusion matrix
  + 5.1.2 Leraning curves
* 5.2 Logistic regression
* 5.3 k-Nearest Neighbors
* 5.4 Decision Tree
* 5.5 Random Forest
* 5.6 AdaBoost
* 5.7 Gradient Boosting Classifier
* 5.8 VotingClassifier

**6. Testing the predictions**

****

*Random Forest*, *Gradient Boosting* and *k-Nearest Neighbors: 75.46%*

**7. Conclusion**