Products Recommendation for customers

CMPE 256 Individual Project

Summer 2019

Rajesh Thummala

SJSU ID: 013829179

This project aims at analyzing the content of an E-commerce database that lists purchases made by ∼∼4000 customers over a period of one year (from 2010/12/01 to 2011/12/09). Based on this analysis, I develop a model that allows to anticipate the purchases that will be made by a new customer, during the following year and this, from its first purchase.

**1. Data preparation**

The datasets are collected from various websites like Amazon, ebay and flipkart. Once done, I also give some basic information’s on the content of the data frame: the type of the various variables, the number of null values and their percentage with respect to the total number of entries:

## 2. Exploring the content of variables

This data frame contains 8 variables that correspond to:

**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**: Invoice 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.

A close up of a device

Description automatically generated

## 3. Insight on product categories

In the data frame, products are uniquely identified through the **Stock Code** variable. A short description of the products is given in the **Description** variable. In this section, I intend to use the content of this latter variable in order to group the products into different categories.

### **3.1 Products Description**

As a first step, I extract from the **Description** variable the information that will prove useful. To do this, I use the following function:

This function takes as input the data frame and analyzes the content of the **Description** column by performing the following operations:

* extract the names (proper, common) appearing in the products description
* for each name, I extract the root of the word and aggregate the set of names associated with this particular root
* count the number of times each root appears in the data frame
* when several words are listed for the same root, I consider that the keyword associated with this root is the shortest name (this systematically selects the singular when there are singular/plural variants)

The first step of the analysis is to retrieve the list of products:

Once this list is created, I use the function I previously defined in order to analyze the description of the various products:

The execution of this function returns three variables:

* keywords: the list of extracted keywords
* keywords\_roots: a dictionary where the keys are the keywords roots and the values are the lists of words associated with those roots
* count\_keywords: dictionary listing the number of times every word is used

At this point, I convert the count\_keywords dictionary into a list, to sort the keywords according to their occurences:

### **3.2 Defining product categories**

The list that was obtained contains more than 1400 keywords and the most frequent ones appear in more than 200 products. However, while examination the content of the list, I note that some names are useless. Others are do not carry information, like colors. Therefore, I discard these words from the analysis that follows and also, I decide to consider only the words that appear more than 13 times.

#### **3.2.1 Data encoding**

Now I will use these keywords to create groups of products. Firstly, I define the XX matrix as:

where the ai,jai,j coefficient is 1 if the description of the product ii contains the word jj, and 0 otherwise.

The XX matrix indicates the words contained in the description of the products using the one-hot-encoding principle. In practice, I have found that introducing the price range results in more balanced groups in terms of element numbers. Hence, I add 6 extra columns to this matrix, where I indicate the price range of the products:

and to choose the appropriate ranges, I check the number of products in the different groups:

#### **3.2.2 Creating clusters of products**

In this section, I will group the products into different classes. In the case of matrices with binary encoding, the most suitable metric for the calculation of distances is the [Hamming's metric](https://en.wikipedia.org/wiki/Distance_de_Hamming). Note that the **kmeans** method of sklearn uses a Euclidean distance that can be used, but it is not to the best choice in the case of categorical variables. However, in order to use the Hamming's metric, we need to use the [kmodes](https://pypi.python.org/pypi/kmodes/) package which is not available on the current plateform. Hence, I use the **kmeans** method even if this is not the best choice.

In order to define (approximately) the number of clusters that best represents the data, I use the silhouette score:

In practice, the scores obtained above can be considered equivalent since, depending on the run, scores of 0.1±0.050.1±0.05 will be obtained for all clusters with n\_clusters >> 3 (we obtain slightly lower scores for the first cluster). On the other hand, I found that beyond 5 clusters, some clusters contained very few elements. I therefore choose to separate the dataset into 5 clusters. In order to ensure a good classification at every run of the notebook, I iterate untill we obtain the best possible silhouette score, which is, in the present case, around 0.15:

## 4. Customer categories

### **4.1 Formatting data**

In the previous section, the different products were grouped in five clusters. In order to prepare the rest of the analysis, a first step consists in introducing this information into the data frame. To do this, I create the categorical variable **categ\_product** where I indicate the cluster of each product:

#### **4.1.1 Grouping products**

In a second step, I decide to create the **categ\_N** variables (with N∈[0:4]N∈[0:4]) that contains the amount spent in each product category:

Up to now, the information related to a single order was split over several lines of the data frame (one line per product). I decide to collect the information related to a particular order and put in in a single entry. I therefore create a new data frame that contains, for each order, the amount of the basket, as well as the way it is distributed over the 5 categories of products:

**5. Testing predictions**

In the previous section, a few classifiers were trained in order to categorize customers. Until that point, the whole analysis was based on the data of the first 10 months. In this section, I test the model the last two months of the dataset, that has been stored in the set\_test dataframe: In a first step, I regroup reformates these data according to the same procedure as used on the training set. However, I am correcting the data to take into account the difference in time between the two datasets and weights the variables count and sum to obtain an equivalence with the training set:

Each line in this matrix contains a consumer's buying habits. At this stage, it is a question of using these habits in order to define the category to which the consumer belongs. These categories have been established in Section 4. \*\* At this stage, it is important to bear in mind that this step does not correspond to the classification stage itself\*. Here, we prepare the test data by defining the category to which the customers belong. However, this definition uses data obtained over a period of 2 months (via the variables \* count \*, \* min \*, \* max \*\* and \*\* sum \*\*). The classifier defined in Section 5 uses a more restricted set of variables that will be defined from the first purchase of a client.

Here it is a question of using the available data over a period of two months and using this data to define the category to which the customers belong. Then, the classifier can be tested by comparing its predictions with these categories. In order to define the category to which the clients belong, I recall the instance of the kmeans method used in section 4. The predict method of this instance calculates the distance of the consumers from the centroids of the 11 client classes and the smallest distance will define the belonging to the different categories:

**6.Conclusion:**

The work described is based on a database providing details on purchases made on an E-commerce platform over a period of one year. Each entry in the dataset describe the purchase of a product, by a particular customer and at a given date. In total approximately ~ 4000 clients appear in the datasets. Given the available information, I decided to develop a classifier that allows to anticipate the type of purchase that a customer will make, as well as the number of visits that he will make during a year, and this from its first to the E-commerce site.

The first stage of this work consisted in describing the different products sold by the site, which was the subject of a first classification. There, I grouped products into 5 main categories of goods. In a second step, I performed a classification of the customers by analyzing their consumption habits over a period of 10 months. I have classified clients into 11 major categories based on the type of products they usually buy, the number of visits they make and the amount they spent during the 10 months. Once these categories established, I finally trained several classifiers whose objective is to be able to classify consumers in one of these 11 categories and this from their first purchase.

Finally, the quality of the predictions of the different classifiers was tested over the last two months of the datasets. The performance of the classifier therefore seems correct given the potential shortcomings of the current model. In particular, a bias that has not been dealt with concerns the seasonality of purchases and the fact that purchasing habits will potentially depend on the time of year. In practice, the seasonal effect may cause the categories defined over a 10-month period to be quite different from those extrapolated from the last two months. In order to correct such bias, it would be beneficial to have data that would cover a longer period of time.