NOVA IMS Information Management School

BUSINESS CASES WITH DATA SCIENCE

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS – MAJOR IN BUSINESS ANALYTICS

Business Case 4 – Online retailer recommender system

Group W

Ana Luísa Mestre, number: 20200599

Beatriz Pereira, number: 20200674

Mariana Domingues, number: 20201040

Nadine Aldesouky, number: 20200568

May 2021

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

INDEX

1. INTRODUCTION

ManyGiftsUK is a UK-based non-store online retailer which focuses on selling unique gifts for all occasions. The company was established in 1981 and only recently switched to totally online. Using its newly collected data, the company would like to devise a recommender system to suggest relevant items to existing and new customers. Our team of data scientists has taken on this project to help ManyGiftsUK build its recommender system which facilitates user choices.

2. BUSINESS UNDERSTANDING

2.1. Determine Business Objectives

- **Background**: ManyGiftsUK is a UK-based online only retailer with around 80 employees which mainly sells unique all-occasion gifts. Most of the company's customers tend to be wholesalers. For many years, the company relied on direct mailing catalogs and orders were taken over the phone. Only two years ago ManyGiftsUK launched its own website and shifted completely to the web. Since then, it has maintained a steady number of customers from around the world which enabled it to accumulate a huge amount of data. Additionally, the company uses Amazon.co.uk to market and sell its products.
- **Business Objective/Problem**: To build a recommender system that can facilitate user choices by suggesting items that the users would like to buy.
- **Business Success Criteria**: A successful recommender system will result in an improved customer experience and an increase in sales as it accurately recommends what users would like thereby easing their choices by suggesting to them the most relevant products even to new users.

2.2. Assess the Situation

- **Inventory of Resources**: The company provided a database in a CSV file which will be analyzed by a team of four business analysts known as Group W. To manage the problem, Python, Microsoft Word and PowerPoint will be used.
- **Requirements, Assumptions, and Concerns**: This report is for managers' use which means that it is for private use at ManyGiftsUK only. Additionally, this is not a technical audience meaning that all aspects must be comprehensible in business terms without technical jargon. For this project, we will be working with the *retail* dataset composed of a sample of 541909 records and 8 variables.
- **Risks and Constraints**: The dataset contains all the transactions occurring between 01/12/2010 and 09/12/2011 only. Additionally, this report must be completed and presented to management within two weeks' time.

Risks	Contingencies
Useless features for the task	Ask for different variables or derive new features i.e., more types of events for collaborative filtering like views and clicks or more variables for content-based filtering like more description on products like category
Incoherent relevant information	Ask for more consistent and clear descriptions of products
Lack of funding or available time	Request an extended deadline
Losing the data	Create a copy of all aspects of the project

Table 2.1.– Risks and Contingencies

- **Terminology**:

Stock code: A code for each item stored in the inventory.
Wholesaler: A company or person which sells products in bulk (i.e., in large quantities) at low prices, mostly to retailers.

- **Costs and Benefits**:

As this is an academic project, we have no information about the costs of the data collection neither the ones regarding the development and the implementation of our solution for the business problem.
The benefits include increased sales, improved customer satisfaction, improved supplier and partner relationships, lower costs, and higher revenues.

2.3. Determine Data Mining Goals

- **Data Mining Goals**: To create a recommender system using collaborative and content-based filtering to predict which products users would like to increase sales and improve customer satisfaction in the future.
- **Data Mining Success Criteria**: A successful recommender system will result in high values for the metric AUC at k, in the case of the ALS model. Additionally, it will result in high lift, support and confidence values in the case of the Apriori model. Besides this, modern recommender systems usually combine both the collaborative and content-based filtering approaches into a hybrid system.

2.4. Produce Project Plan

- **Steps**

1. General exploratory analysis, identifying clear problems in the data such as missing values, outliers, or incoherencies.
2. Deeper exploratory analysis.
3. Describe the insights obtained from the data exploration - explore; produce and interpret visualizations to understand the absolute frequencies of the variables, or the presence of outliers.
4. Prepare the data to the model input – select the required features, apply features transformations, create new features, detect outliers, and/or correcting incoherencies.
5. Review the data again, after cleaning it (e.g.: see if the insights are the same or if they changed slightly).
6. Reduce the sparsity of the data then split it into train and test sets.
7. Define and apply the ALS technique and recommendation algorithm to create a supervised predictive model.
8. Fine Tuning (making small adjustments to parameters to achieve the desired output of performance) of the model, and selection of the one that provides the best solution for ranking products for each user based on the higher scores (i.e., precision, ndcg, etc.).
9. Apply the *Apriori* Algorithm to identify the products that tend to be bought together.
10. Assess and interpret the results from the chosen models.
11. Provide a deployment plan.

Initial Assessment of Tools and Techniques

Technique	Pros	Cons
ALS[1][2]	-Easy implementation -Good performance	-Popularity bias -Scalability issue

***Apriori* Algorithm**[3]	-Simple and easy to implement. -Works well with large datasets (includes pruning steps).	- Computationally expensive based on the thresholds and diversity of items. - Very complex in terms of time and space. - Performance is reduced by the numerous times the algorithm scans the dataset.

Table 2.2.– Tools and Techniques

3. PREDICTIVE ANALYTICS PROCESS

3.1. Data Understanding

The dataset was provided in the form of a CSV file (*retail.csv*), which contains 8 variables and 541909 rows, each for a particular item present in a transaction. The data was taken from the period between 01/12/2010 and 09/12/2011. This corresponds to 53 weeks and/or 373 days. Inside the dataset, there are 25900 valid transactions, associated with 4070 unique items and 4372 customers from 38 different countries.

When exploring the dataset, we noticed that:

1. From the total of 25900 transactions, 3836 were canceled (14.81%).
2. There are 2515 instances with unit price equal to zero and 2 with negative unit price.
3. There are 10624 instances with negative quantities, from which 9288 correspond to canceled transactions and 1336 to not canceled transactions. From these last observations we noticed that all the prices are assigned as 0, there is no Customer ID and the products' descriptions are mostly relative to damaged information, leading us to think that it was actually a canceled transaction, but not by the customer rather by the company.
4. As mentioned above, we have transactions from 38 different countries, of which 24 are European, 8 are Asian, 3 are American, 1 is African and another one is Oceanic. There are also 13 transactions that have an unspecified country. However, 90.71% of the total number of transactions in the dataset came from the United Kingdom (23494 transactions).
5. There are 10147 duplicated instances, representing approximately 1.87% of the data.
6. There are 19265 duplicated instances considering all the variables except Quantity. These situations happen when we have the same product in a certain transaction and, instead of summing the quantity of that product, it is added in a different row.
7. We expected to have the variables 'StockCode', 'Description' and 'UnitPrice' with a relationship of one-to-one, even so, there are 1324 products with more than one description (varying from 2 to 9) and 3602 products with a lot of different prices (varying from 2 to 687).
8. From the information provided in the metadata, we know that the values of the variable 'StockCode' are in the form of a 5-digit code; however, there are products that do not follow this format, specifically 10 of them, that are shown below, have less than 5 digits.

Stock Code	Description	Frequency (Number of observations)	Variety of Prices
POST	POSTAGE	1256	114
DOT	DOTCOM POSTAGE	710	687
M	Manual	571	260
C2	CARRIAGE	144	6
D	Discount	77	75
S	SAMPLES	63	59

CRUK	CRUK Commission	16	16
PADS	PADS TO MATCH ALL CUSHIONS	4	2
m	Manual	1	1
B	Adjust bad debt	1	1

Table 3.1.– Doubtful Products' Characteristics

These products are the ones representing the highest fluctuation of prices, probably, not only caused by small variations during the presented year but also because of their properties varying from customer to customer or even by country. Removing the PADS product and looking at the products' descriptions, this group seems to be related with complementary taxes of a transaction rather than the description of a product. 0.52% of the total number of transactions include these products.

After treating some inconsistencies enumerated above and explained in greater detail in the next section, we can present the next visualizations:
First, we created some queries by grouping information per transaction, customer, and country, and visualized their general distribution in order to check for outliers or common patterns:

- nproducts_byTransaction: number of different products per transaction.
- Tquantity_byTransaction: total quantity of products per transaction.
- Maxquantity_byTransaction: maximum quantity of products per transaction.
- MoneySpent_byTransaction: total bill per transaction.
- ntransactions_byCustomer: number of transactions per customer.
- ntransactions_byCountry: number of transactions per country.

Variable	Count	Mean	Std	Min	P25	P50	P75	Max
nproducts_byTransaction	18416	20.98	23.85	1	7	15	27	541
Tquantity_byTransaction	18416	280.80	979.35	1	75	156	292	80995
Maxquantity_byTransaction	18416	69.52	828.01	1	12	24	48	80995
MoneySpent_byTransaction	18416	475.74	1678.31	0	156	302	465	168469.6
ntransactions_byCustomer	4336	4.25	7.64	1	1	2	5	206
ntransactions_byCountry	37	497.73	2721.19	1	4	17	47	16592

Table 3.2.– Descriptive statistics for some queries

To explore popularity patterns, we first plotted the Long Tail Plot and then focused only the top 15 products with the number of transactions per product. As expected, most products are in the "long tail", representing a small percentage of transactions (10%), and only a small percentage of products have a high volume of transactions. In the head, with a 90% threshold, there is a smaller variety of products than in the rest of the transactions. The product with the description 'White hanging heart t-light holder' is the one included in the highest number of transactions (almost 2000, representing 10.7% of the data).

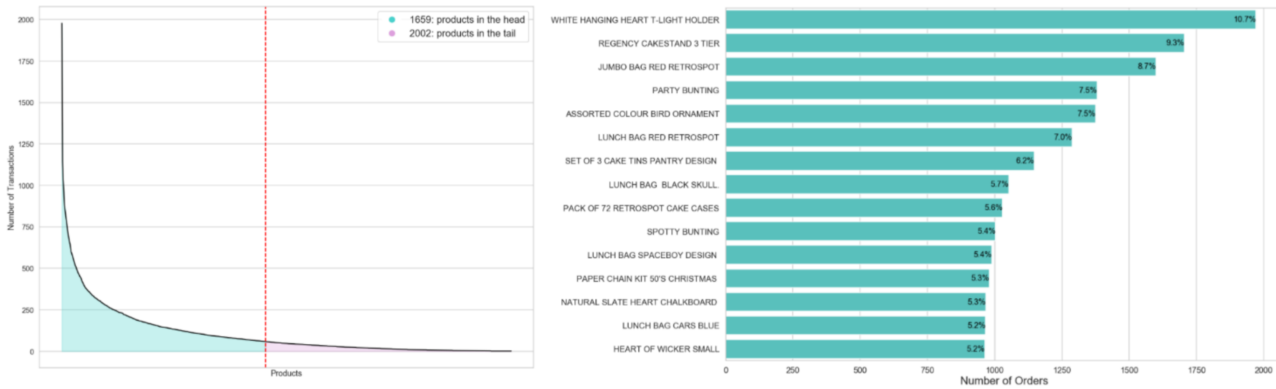

Figure 3.1. – Total Transactions of each product

Then, we plotted the number of transactions by month and by hour. Regarding the first graph, we can see that the last months of the year – September, October, November and December – are the ones when people buy most, November being the month with the highest number of transactions (around 60000). This is logical because these are the months that have Christmas sales, Black Friday and Cyber sales. Additionally, it is the gifting season. As for the second graph, the hours that correspond to the middle of the day are the ones that register the highest number of transactions.

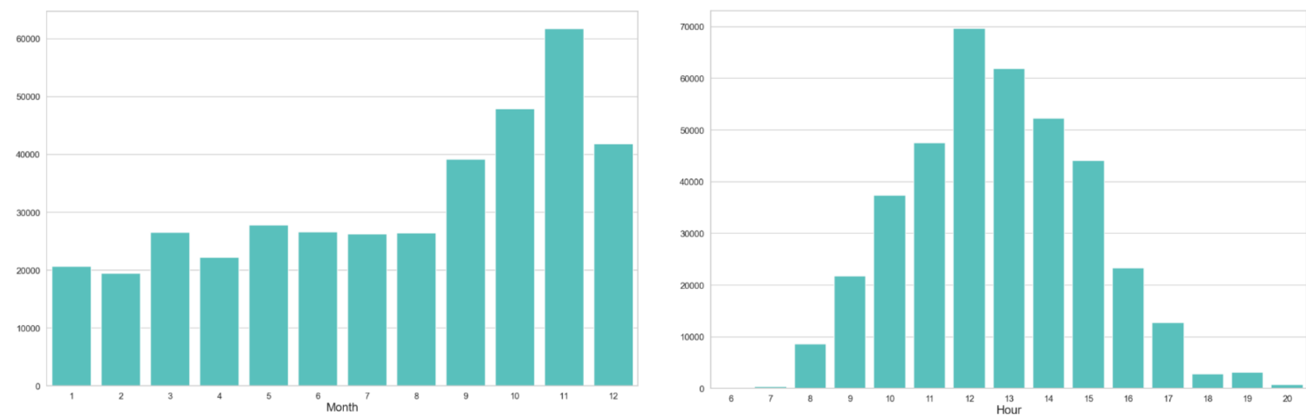

Figure 3.2. – Number of Transactions by time

3.2. Data Preparation

In this phase, we will apply some data cleaning, do several feature engineering and prepare the data to the input of the models being applied.

3.2.1 Missing data

There are 135080 missing values in the variable 'CustomerID' (24.93% of the total dataset), which represent the customers who do not have an account on the website of the company, and 1454 missing values in the variable 'Description'.

3.2.2 Data Selection and Treatment

Following the enumerated points in the data understanding, we decided to take care of some inconsistencies that could affect the model performance and the output interpretations.
We started by dropping the two observations with negative unit price, since we do not have any information about its meaning on the metadata and they are only two.

In order to have only one row for each specific product in a transaction, we treated the duplicates mentioned in point 6, aggregating them by summing the quantities by each row.
After this, given the updated information from the manager, we dropped all the cancellations since the data contains the original invoice with a distinct InvoiceNo, avoiding duplicate any information for the algorithm.
The customers without an ID were also dropped from the main dataset, since they are customers without any historical information and the ALS algorithm will not be capable of doing a good recommendation with only one transaction. For the latter, a general recommendation will be applied.
From the doubtful products mentioned in point 8, after the previous treatment we were only left with 5 of them: C2, DOT, M, PADS and POST. As said previously, only PADS seems to be a real product, and for that reason only the rows including C2, DOT, M and POST will be dropped, representing 0.39% of the data.
After this, the dataset had 386449 rows (71.3 % of the original data), however we only lost the rows containing the products mentioned above and the transactions with negative unit prices (0.50%), since the duplicated rows were aggregated keeping all the information, the cancellations have their original transaction in the data and the transactions with a missing customer ID were saved in a different dataset.

3.2.2.1 Outliers' Analysis

From the variables mentioned on table 3.2, we analyzed their boxplots and histograms in order to better understand their distribution and check for outliers.
Although these graphs showed very high values for the number of transactions and products by customer, we considered this logical since most of the customers of ManyGiftsUK are wholesalers. Thus, we did not want to be too critical about these observations, nevertheless looking with more detail to the total quantity and money spent by transaction we have a few extreme transactions. Consequently, we only dropped 2 transactions that had more than 70000 products and a bill higher than 168000, values extremely divergent from the rest of the observations. These 2 transactions correspond to only 2 rows (0.0005%) meaning that this extreme quantity and price were to buy a unique product.

3.2.3 User-Item Rating Matrix

Finally, to prepare the data for the ALS model we needed to create the Rating Matrix crossing the 'CustomerID' with 'StockCode' using the class coo_matrix from scipy.sparse [4]. Beforehand, in order to optimize the results of the recommender system we had to solve the cold start problem. Usually, recommender systems are restrained due to the lack of user similarity knowledge available. This tends to limit the creation of high-quality user communities which implicitly impacts the recommender algorithm [5]. To address this, we reduced the sparsity [6] by only including users with more than 5 transactions and products bought more than 10 times. The customers with less than 5 transactions were added to the dataset of the customers without an ID, redefined as the new customers. This is to have the same type of recommendation given the lack of personalized information.
In both data sets, the train-test split was the same, we selected 46 weeks for the training set and the remaining 7 for the test set (75/25).

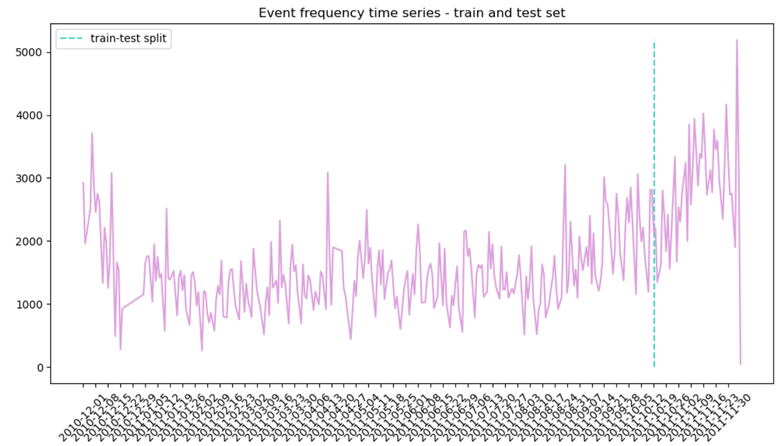

Figure 3.3. – Time Series

3.3. MODELING

3.3.1 Select Recommender Technique

To be able to respond to the proposed objectives, we resorted to 3 modelling techniques:

- Directed to the traditional/usual users: A personalized system using the ALS model which originates a recommender system built based on collaborative filtering, using a user's past item interactions (in this case, an implicit interaction: transaction) as well as similar decisions made by other users.
- Directed to the new users: A generalized system using the popular recommender model for the users who do not have an account on the website of the company or the ones who made less than 5 transactions during the time considered. It is important to underline here that we also treated these second group of users as new customers because we did not have sufficient data to make recommendations to them in a different way.
- Directed to all users: A complementary system using the *Apriori* algorithm with the objective of understanding and recommending products which are usually bought together.

3.3.2 Collaborative Filter – ALS

In this recommender system, it is possible to suggest the products which have the largest inner product with the user vector and find related items that have the largest inner product with the item vector.

To get the best values for the AUC metric used in the model, we applied a grid search for the regularization, number of factors, number of iterations, and alpha parameters. In this context, we ended up using the following values for these parameters:

- Regularization [0.05,0.01,0.005] = 0.01
- Factors [150,175,200,225,250] = 250
- Iterations [30,35,40] = 35
- Alpha [40,45,50] = 40 (default value)

Here, it is important to underline the alpha parameter which gives us a larger confidence the more times a user has made a transaction.

3.3.3 Top K Popular Products – Popular Recommender

Considering the Cold Start Problem, and to combat the fact of having few data about what the new customers usually buy, this recommender system will suggest the top 10 popular products that they buy when making a transaction on the website of the company for the first time. Although a system not personalized, it reached the highest score for these types of costumers where we do not have data to get to know the customer preferences.

3.3.4 Association Rules – Apriori Algorithm

For this recommender system we found it interesting to suggest products based on what the customers usually buy together. As a result, we made use of the the *Apriori* algorithm which assumes that all subsets of the frequent item sets must also be frequent while all supersets of the infrequent items will also be infrequent. In this context, we generated the following results:

antecedents	consequents	antecedent support	consequent support	support	confidence	lift
(22697)	(22699)	0.054585	0.057327	0.041301	0.756650	13.198791
(22699)	(22697)	0.057327	0.054585	0.041301	0.720450	13.198791
(22386)	(85099B)	0.065501	0.112503	0.044367	0.677340	6.020619
(21931)	(85099B)	0.063673	0.112503	0.038935	0.611486	5.435273
(22411)	(85099B)	0.063189	0.112503	0.036569	0.578723	5.144054
(22383)	(20725)	0.069105	0.084162	0.035225	0.509728	6.056476

Figure 3.3 – Association Rules

Indeed, we have a set of complementary products, for example, ALARM CLOCK BAKELIKE GREEN are bought together with ALARM CLOCK BAKELINE RED. Moreover, JUMBO BAG RED RETROSPOT is a complement of the products JUMBO BAG PINK POLKADOT, JUMBO STORAGE BAG SUKI and JUMBO SHOPPER VINTAGE RED PAISLEY.

3.4 EVALUATION

3.4.1 Evaluate ALS Model's Results

In order to evaluate the Popular Recommender (Naïve) and ALS models, we used the following metrics: Precision at K, Mean Average Precision at K (map), Normalized Discounted Cumulative Gain at K (ndcg) and AUC, AUC being the one we gave more importance to encounter the best parameters, with the grid search aiming to maximize this metric.

	pop_model	als_model
precision	0.095360	0.060253
map	0.061715	0.036538
ndcg	0.100412	0.060119
auc	0.506615	0.504074

Figure 3.4 – Metric Results from ALS model

From [7], *"A perfect recommender would yield a ROC curve that goes straight up towards 1.0 recall and 0.0 fallout until all relevant items are retrieved. Afterwards it would go straight right towards 1.0 fallout while the remaining irrelevant items follow. The obvious aim is consequently to maximize the area under the ROC curve. The area under the curve (AUC) can therefore be used as a single measure for the overall quality of a recommender system."* Thus, a greater AUC means that we are recommending items that are being purchased and which can be found in the first ranks of recommended items.

Even if AUC has a higher value on the popular recommender model, we chose the value on the ALS model, given its novelty and its higher personalization.

3.4.2 Evaluate Popular Recommender's Results

As for the results obtained in the Popular Recommender model, we got a performance of approximately 0.5 in the AUC metric and a precision of 1, meaning that 100% of the recommendations we made are relevant to the users.

	pop_model
auc	0.501992
map	1.000000
ndcg	1.000000
precision	1.000000

Figure 3.5 – Metric Results from POP model

3.4.3 Evaluate Apriori's Results

Regarding the results obtained in the *Apriori* algorithm, we can see in Figure 3.3 that both the confidence and the lift are high and that is why we can trust these rules. As for the support, the values may seem not very high; however, we consider them sufficient to proceed.

3.4.4 Review Process

During this project, we had some problems regarding the metadata, namely the cancellations, the negative quantities, the null prices or some strange descriptions of the products.
Thus, both the data and metadata were missing important information for us to understand the context and the meaning. Besides this, the time constraint was quite restricting. It would have been helpful if we have had additional time to research and familiarize ourselves with the more complex recommender models in order to not only present better recommendations, but also to solve the cold start problem.

3.4.5 Determine Next Steps

- Move to the deployment step.
- Improve the data preparation step, such as the outliers' analysis, for example.
- Train and test the predictive model with different data splits - using K-fold cross-validation, for example.
- Improve the predictive model - tuning the parameters or trying different recommender methods/models.
- Train and test different models – for example Light FM.

4. DEPLOYMENT

4.1. Deployment Plan - Recommendations and Challenges

After identifying the customers' preferences, the next step is to determine how to proactively reach those customers and improve the expression of meaningful and useful recommendations on the company's homepage by offering a set of products that the customers might be interested in.
To achieve this, we present three different recommender systems, according to the type of customers we are dealing with and the foundation on which the recommendations were built:

- **Additional products of interest**: Directed to the old/usual users of the website of the company. One way of improving it would be to incentivize the users to give ratings and reviews to the products, reinforcing the explicit interactions.
- **Your first time**: Directed to the new customers of the website of the company. A general recommendation with the top 10 products. Given the lack of information about these customers

this option is the one that gives the best results. One way to have higher personalization would be to create a preference menu, where we can recommend products by products' categories.
- **Fill your basket**: Directed to all the customers of the website of the company. In this case, a set of complementary products would be recommended.

Furthermore, ManyGiftsUK can start customizing its landing page or website layout so that the preference of the user/customer can always be at the top of the page. Additionally, the company can start to make coupons or promotions with bundles of products that were identified as preferences of the user.

When deploying a recommender system, it is also important to consider some key aspects such as: privacy concerns, trust issues, scalability (it should be scalable to billions of users with different habits and preferences), adaptability (it should adapt quickly to the fast-changing world of content), lack of data, unpredictable items, diversity, and user preferences. As time goes by and the number of new and different products increases, the coverage will decrease if the company continues to apply our model. Our suggestion in this situation would be to review the model whenever X new products enter the company. Besides this, the Cold Start Problem, which is the capacity of the recommender system to suggest relevant products to new customers, is another important aspect that should be considered. Finally, there is also the biased characteristic of the popular model, meaning only recommending the popular products, which are the most bought, and leaving behind the ones which are not bought as often.

4.2. Plan Monitoring and Maintenance

- Review the model when its evaluation metrics decrease drastically.
- If the company moves to a more digital business model, it can benefit of an app/program that alerts when a product becomes more and more popular as it is identified as a preference for more users.
- If the customers of the company diversify, meaning they stop being mostly wholesalers, this project would have to be reviewed as this would give rise to new customer profiles, behaviours, preferences and order quantities.
- If current conditions do not change, the analysis must be reviewed quarterly to ensure its viability and relevance since data associated with gifting products is quite seasonal.

4.3. Review Project

To sum up, we believe that additional data would have been very beneficial for this project. Supplementary information about the products i.e., its category or other attributes would have been beneficial for content-based filtering. Additionally, more information about the customers' interaction with the website (such as views or likes) would provide for much better recommendations and a richer analysis as this could help with collaborative filtering. Additional implicit and explicit knowledge would have also aided the model. Explicit information such as a score, rating or review of a product made by the customer would help with the identification of preferences. Lastly, structured and normalized data, presented in separate files containing the information of the products, the transactions or the customers would certainly have helped in the initial data management. Thus, in the future, and to prevent these situations from happening again, we think that it is very important and urgent for the company to hire people who can work and be responsible for the data management, since it was not achieved in this project.

5. REFERENCES

[1] "Prototyping a Recommender System Step by Step Part 2: Alternating Least Square (ALS) Matrix Factorization in Collaborative Filtering | by Kevin Liao | Towards Data Science." https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1

[2] "Matrix factorization using the alternating least squares algorithm for collaborative filtering - Statistics for Machine Learning [Book]." https://www.oreilly.com/library/view/statistics-for-machine/9781788295758/9f97a5da-0997-4941-aff4-3ab3d9dc7c6d.xhtml

[3] G. L. Team, "What is Apriori Algorithm in Data Mining Implementation and Examples?," 2020. https://www.mygreatlearning.com/blog/apriori-algorithm-explained/

[4] "Sparse matrices (scipy.sparse) — SciPy v1.6.3 Reference Guide." https://docs.scipy.org/doc/scipy/reference/sparse.html

[5] D. C. Wilson, B. Smyth, and D. O'Sullivan, "Sparsity reduction in collaborative recommendation: A case-based approach," in *International Journal of Pattern Recognition and Artificial Intelligence*, Aug. 2003, vol. 17, no. 5, pp. 863–884, doi: 10.1142/S0218001403002678

[6] "Intro to Implicit Matrix Factorization: Classic ALS with Sketchfab Models | Ethan Rosenthal." https://www.ethanrosenthal.com/2016/10/19/implicit-mf-part-1/

[7] G. Schröder, M. Thiele, and W. Lehner, "Setting Goals and Choosing Metrics for Recommender System Evaluations." Available: http://www.grouplens.org/node/73