Gift-a-Lot

Business Case 3:
Recommender Systems

Apex Pattern Deployers

Marjorie Kinney *m20210647*

Bruno Mendes *m20210627*

Lucas Neves *m20211020*

Farina Pontejos *m20210649*

Business Cases for Data Science

NOVA Information Management School

April 2022

Table of Contents

Introduction

Gift-a-Lot is a UK-based and registered retailer that shifted completely to the web as a non-store online retailer. Our task is to help the company build a recommender system able to facilitate customer choices on the website and increase sales.

Recommendation systems are very important in industries like e-commerce or retail, as customers tend to buy more when faced with easy choices and it helps them discover items they might not find otherwise. In general terms, these systems algorithmically filter information seeking to predict the preferences of a user regarding a given item.

In order to accomplish this task, we will also conduct a market basket analysis to understand the purchasing patterns and behaviors of customers, identify different types of products (e.g. complementary, substitute, etc) and ultimately understand the meaning of purchases that commonly happen together. This information can later be used to develop different marketing strategies for customers and greatly improve the recommender system development.

Lastly, we will discuss evaluation strategies and metrics, as well as appropriate quality measures for our proposal.

Business Understanding

For many years Gift-a-Lot sent physical catalogs to customers via direct-mailing campaigns, with orders being taken over the phone. Two years prior to now, the company shifted its orders online and launched its own website. It has both retail and wholesale customers from around the world, with most being from the United Kingdom.

In order to improve the user experience and increase sales, Gift-a-Lot must adopt what has become commonplace with online retailers- a product recommendation system. Consumers have become more technology-savvy in the past few decades, and online shopping has in turn become more ubiquitous. Recent quarantines due to the COVID-19 pandemic have increased the use of online shopping even further, with online retail sales in the UK up more than 33% on pre-COVID levels. To take advantage of this, Gift-a-Lot can implement a product recommender system based on market basket analyses of historical purchases. The data can be clustered according to customer type, and strategies for each type of customer can be developed. The market basket analysis can also be used to help address the “cold start” problem, that is, what to recommend to new website visitors.

Data Understanding

We were given a customer transaction dataset with 8 different variables, containing all the transactions between 01/12/2010 and 09/12/2011. The variables found are as shown in Table 1 (Appendix). Furthermore, we were informed that over that period there were 25 900 transactions associated with 4 070 unique items and 4 372 customers from different countries that can be of different types (e.g. wholesalers). Lastly, the dataset contains 541 909 instances, each for a particular item contained in a transaction.

After conducting an initial analysis of the variables, we have discovered some problems in the dataset like odd text and information in the place of the descriptions (E.g. “wrong code?”, “found”, etc), as well as similar items with different descriptions or no description at all. Moreover, we could indeed confirm the 25 900 unique invoices and identified 4 373 unique customer ids.

Regarding different findings, there were 135 080 transactions without customer id (24,93% of the data) from which 3 710 are unique (14,32% of all unique invoices). In addition, 3 836 unique invoices were canceled. Some of the transactions also had quantities and unit prices below or equal to 0 (10 624 quantities less than 0, 2 unit prices less than 0, 2515 unit prices equal to 0) corresponding to possible adjustments, cancellations or giveaways.

Further analysis showed that the distribution of customer purchases is mostly right-skewed with the quantity of items purchased varying in a wide range of frequencies occurring from 1 to 1 000 items (mean of 324,95 with a median of 252 items) . The same thing happens with invoice values (quantity * unit price) with likewise varying frequencies occurring from 1 to 5 000 both in absolute terms (mean of 986,41 with a median of 602,56) and on average as seen in Figure 1 (Appendix). Moreover, when looking at the sparsity matrix given by the heatmap of the invoices and the products purchased (Figure 2 and Figure 3 in the Appendix), we can once again confirm that there is a lot of sparsity in the data justified by very different types of transactions, customers and invoices.

Data Preparation

As mentioned before, we are using a new variable called quantity which is essentially unit price times the product quantity. In addition, we feature engineered a column called “isCancelled” which we used to filter out the invoices regarding adjustments and cancellations (the ones that begin with a letter “A” or “C”; A=3, 9 288=C) as we do not want these to influence the performance of our recommender system.

Regarding the outliers, we used the threshold method and removed 1 454 outliers just for clustering purposes (explained ahead), which significantly improved the invoice information distribution (as seen in Figure 4 and 5 in the Appendix). As for the rest of the algorithms only 3 641 additional rows were removed because of non-product descriptions. This being, there was an overall removal of 12 932 transactions (9 291 canceled invoices + 3 641 transactions with non-product descriptions), representing 2,39% of the data.

On a different note, there was a lot of cleaning and parsing of the product descriptions in order not to lose data and maximize the potential of our recommender system. Initially, there were a total of 4 224 unique product descriptions. We started by removing the non-products from the descriptions (215), such as “damage” or “error”. Then we removed all the non-words and symbols (191) using stop-lists, like “set” or “of” (E.g. “Set/4 rose botanical candles” to “ rose botanical candles”). Subsequently, we used a lemmatizer to remove the plural (11) of the items and lastly, the colors (476) and adjectives (173) were also removed.

In the end, there were a total of 3 158 unique item descriptions from the initial 4 224 (representing a reduction of 25%; E.g. “set of 4/ rose botanical candles” to “candle”).

It is also worth mentioning that even after the transformations in the product descriptions, the data continues to be very sparse as seen in Figure 6 (Appendix).

Market Basket Analysis

In order to solve the cold start problem, in which we don’t know what to recommend to new customers, and to analyze product sales patterns, we decided to do a market basket analysis. To prepare, we first separated the customers into two categories: retail and wholesalers. This is crucial since wholesalers are stated to be most of the customers and are expected to have different purchase behavior compared to retail customers.

In order to achieve this, we grouped the data by customer with the following features: total items purchased, total unique items purchased, total number of invoices, total value (quantity * price) and average value. We believe these reflect the key differences between retailers and wholesalers with the data provided.

Next, we removed a few outliers, scaled the data using the standard scaler and reduced dimensionality with the UMAP algorithm. Then, we obtained a K-Means clustering solution from it. We chose two clusters since we’re looking for two distinct categories. The results were as follows:

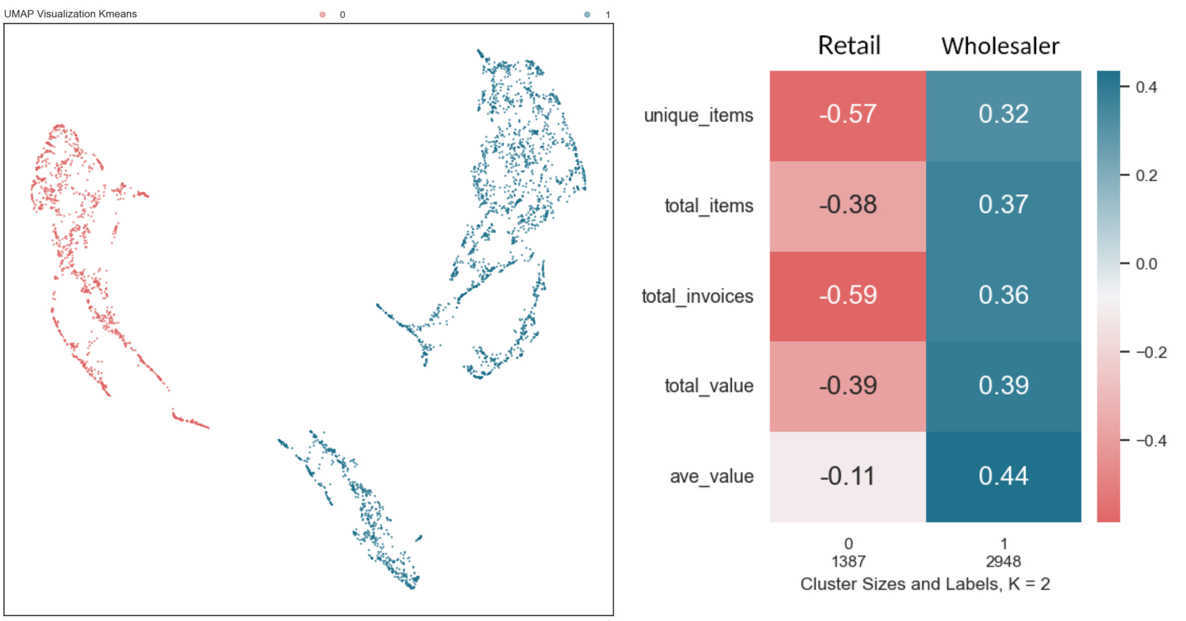

Figure C1 – UMAP and Clustering Solution of Customers

The wholesalers make up about 68% of the customers, which matches the statement made by the company that they are the majority. There is a clear and very significant distinction between the two: the wholesalers purchase more unique items, they purchase more items, they produce many more invoices, they have much more total and average value. The middle data “island” visible in the UMAP could be other types of companies, smaller wholesalers or very enthusiastic retail customers, but due to lack of information we do not consider them their own category in this case.

We proceeded to perform the market basket analysis using algorithms to extract frequent item-sets and reveal association rules, for retail and wholesalers. Applying the association rules algorithm to the frequent itemsets produced several measures. For each antecedents-consequences set, antecedent support, consequent support, support, confidence, lift, leverage, and conviction are provided.

Itemsets with high support occur often in the dataset. Antecedent support reflects the proportion of transactions that contain the antecedent, and consequent support computes the support for the itemset of the consequent. Support computes the overall support for the combined itemset. For each of these measures, higher support will indicate higher frequency, or higher importance, in the dataset. The confidence measure computes the probability of seeing the consequent in a transaction, given that the transaction contains the antecedent. This measure highlights the relationship between items. Itemsets with high confidence occur together often.

Lift is equal to confidence divided by support. This measure shows how much more often an itemset occurs together than if the individual items were statistically independent. It can help to predict what a customer will buy (the consequence), if they have the antecedent item in their cart. Related to lift is the concept of substitutes. Items with a lift less than 1 are considered substitutes. That is, a substitute is an item that the customer buys instead of the other item. Since there are no items with a lift less than 1 in this dataset, there are little to no substitutes present. However, there are a lot of complementary products such as playhouse kitchen and playhouse bedroom.

Leverage measures the difference between the observed frequency of the antecedent and consequent appearing together and the expected frequency if they were independent. Conviction measures how dependent the consequent is on the antecedent, if they’re independent the value is 1.

With a support of at least 10%, the algorithms detected 278 possible rules for the retail dataset, and 672 for the wholesale. This is expected because wholesalers buy more unique items and are the majority of the customers. A further exhaustive analysis can be conducted especially once types of products are established. See figures 7 & 8 for a visual representation of the various purchase connections between the items. It’s visible that the retail dataset seems to have many more “one-off” specific connections than the wholesalers.

Recommender System

To prepare for the recommender system, all invoices with only one line item were dropped, as they were not useful for finding additional products to recommend.

In order to obtain recommendations, first a Term Frequency Inverse Document Frequency vectorizer (TF-IDF) was applied. The TF-IDF measures the originality of a word, by comparing the term frequency in a document with the number of documents the word appears in. In this case the documents being compared are the invoices. This produces a numeric comparison of the similarity of two documents.

Next a cosine similarity matrix algorithm was applied to the data obtained from the TF–IDF. The algorithm can be applied in two ways. First, the algorithm matches the current basket with the most similar invoices from past purchases. Any items in the invoice that are not currently in the basket are then returned as recommendations. As another approach, the algorithm matches the customer with the current basket to customers with similar baskets. In this way, any items in the basket are used to match customers rather than invoices, and the matched customers' items are recommended.

When implementing this algorithm, we found that the invoices that were most similar to a given invoice were often past purchases from the same customer. In order to avoid just recommending items that the customer already purchased, recommendations from invoices with the same customer ID were excluded.

This recommendation system can be implemented in several ways. The first method is to implement it exactly as described above, in the same way for everyone. The other method is to divide the data into the two clusters found earlier, so that retail customers are only given recommendations from the data from other retail customers, and wholesale customers are only given recommendations from the data from other wholesale customers. In practice we found the results to be quite similar, so to reduce complexity we recommend implementing the same recommender system to all customer groups. In addition to using the TF-IDF cosine similarity matrix algorithm, other recommendations can be pulled based on similar items from the market basket analysis.

This recommender system works well for existing customers, and for a cold start customer who has added an item to their basket. Once an item is added, regardless of how many times the customer has or hasn't visited the site, the algorithm runs and recommendations are made. Gift-a-Lot has some options when it comes to a cold start without any items in the basket though. Based on the market basket analysis, Gift-a-Lot can recommend their most popular items to cold start customers. In this way they can be certain the item is likely of interest to anyone visiting the site. The unfortunate side effect of only recommending popular items is that eventually, only popular items are sold. One way to guard against this is to also recommend less-popular items based on whatever factor the company chooses. For instance, the system could recommend overstocked items, popular items from the current week, random items, and/or anchor items that are present in many itemsets. These options can be evaluated and improved using a multi-armed bandit algorithm once deployed.

Evaluation

It’s hard to implement evaluation metrics in this case, since the only available data is past data. By doing a time-split and training the model on entries from a previous time, we could in essence evaluate our model by measuring how good it is at forecasting what a customer will buy in the more recent months we have data on.

For example, if our recommender system decides to recommend a lighter to a certain customer and that customer did indeed buy that lighter, we could count it as a success and build a rating based on that. The problem here is that the customer bought that item anyway without the recommendation, so how valuable is that recommender system really? The forecasting could bring attention to certain relevant items sooner to the customer, but otherwise we hesitate to use common accuracy-based metrics to rate the recommender system.

The value would lie significantly in recommending items and/or a combination of items that the customer was not going to buy, but did because they were brought to the customers’ attention. With this in mind, accuracy-based metrics are insufficient since they downplay novelty, diversity and unexpectedness which, in our view, are especially important for the gift shop business model.

To really evaluate the system, a test version should be deployed which would provide us with certain utility metrics such as CTR (click-through rate) – that tells us how often the recommendations are clicked by the customer, and so their interest in them. We could also measure sales spikes that the implementation might produce afterward. Another factor that we think is important in this case are coverage-based metrics such as what type of products are being successfully recommended, since there is a large variety in the business. In terms of novelty, we could measure the increase in value derived from more fringe products.

One powerful method is A/B Testing, where we could, for example, set up an experiment where only a random part of customers interact with the recommender system. Then we could analyze if their value increases over-time at a higher rate than those who do not interact with the new system and if it is statistically significant.

Deployment

We developed a proof of concept application[1] to demonstrate how the company might make use of the insights gained from this project.

We propose two different views: the first is a customer view, which is what the customer would see as they browse the online shop. As they add items to their cart, the page would display a sidebar with other things that the algorithm would recommend.

[1] https://bc3.onrender.com/

The second is a manager view, which is where managers would see a graph-based representation of which items can be considered the anchor products, and which item combinations occur more rarely.

One challenge that we found in using available app hosting solutions is that large datasets become very slow in delivering recommendations. To be effective, the algorithm needs to be able to deliver recommendations within moments of customers performing an action on their website, whether by looking at a product page or adding an item to cart.

For this reason, we limited the number of invoices processed by the recommender system, and pre-calculated the market-basket graph beforehand instead of letting the server perform the calculations "live".

Another challenge comes from deciding on how many products to recommend, as research has shown that the right number of recommended items depends on both product category and customer type. Finally, in order to stay effective, new invoices will need to be fed back into the algorithm at regular intervals.

To improve the system in the future, we have several recommendations. First, including additional information about the customers would allow the implementation of a collaborative filtering algorithm, rather than just the content-based filtering applied here. To do this, Gift-a-Lot could track whether customers are retail or wholesale, and could use a service such as Google Analytics to track customer demographic data. Another recommendation is to keep the database a bit easier to work with by adding product categories and by keeping product descriptions free of extraneous information, possibly by adding an additional field for comments. Finally, the cold start problem can be improved with the addition of a multi-armed bandit algorithm. This algorithm takes into account the performance of the recommendations, and recommends well-performing items more often than lower-performing variations.

Appendix

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 pounds.
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.

Table 1 – Variables in the transaction dataset;

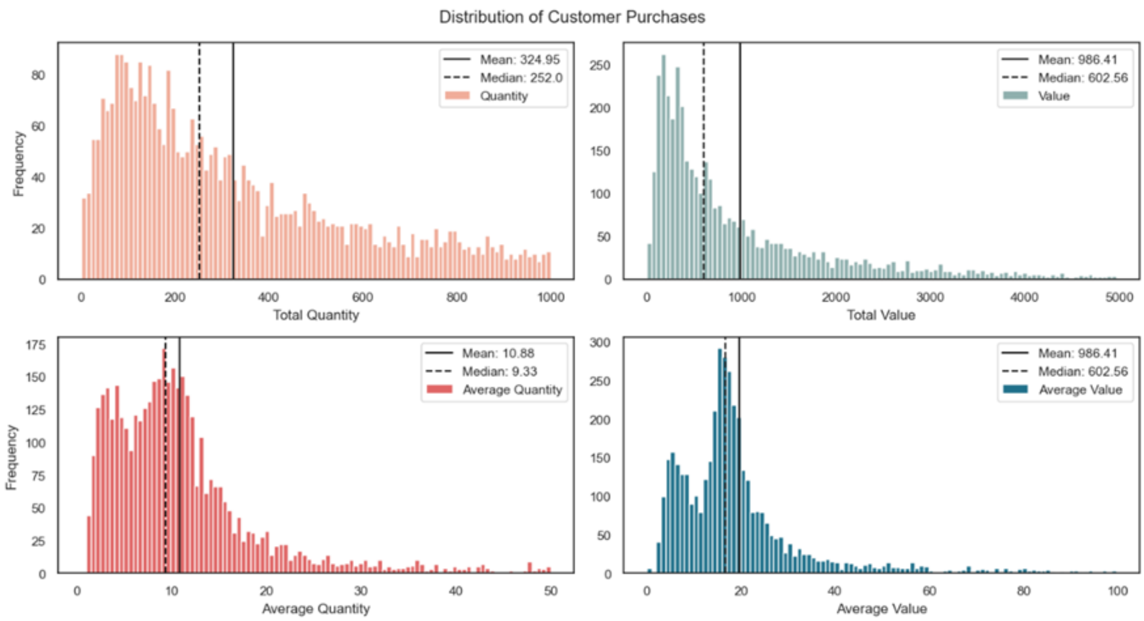

Figure 1 – Distribution of Customer Purchases;

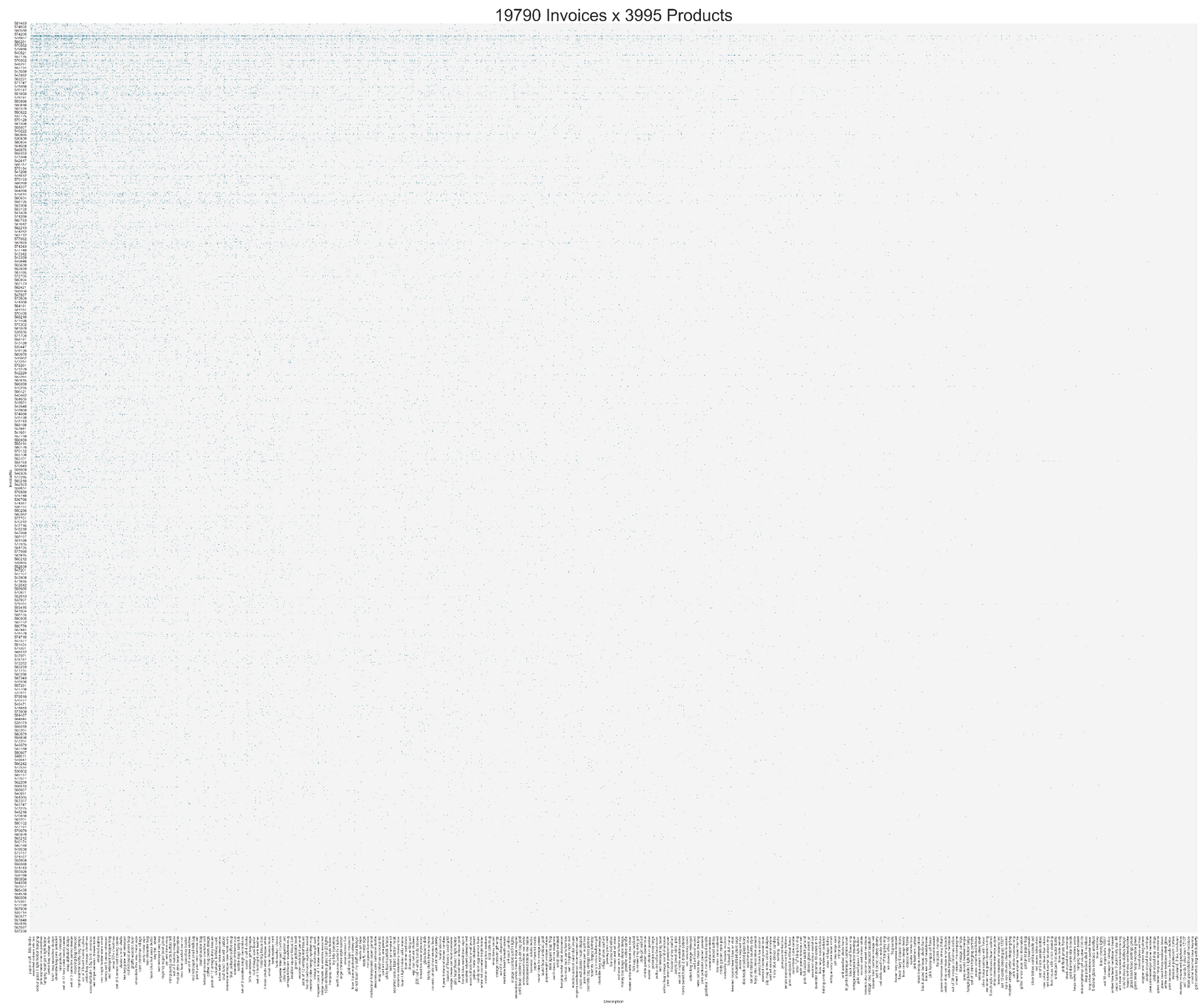

Figure 2 – Sparsity Matrix (heatmap of the invoices and the products purchased);

Figure 3 – Sparsity Matrix (heatmap of the invoices and the products purchased) Zoomed in with 100 invoices and 100 products purchased;

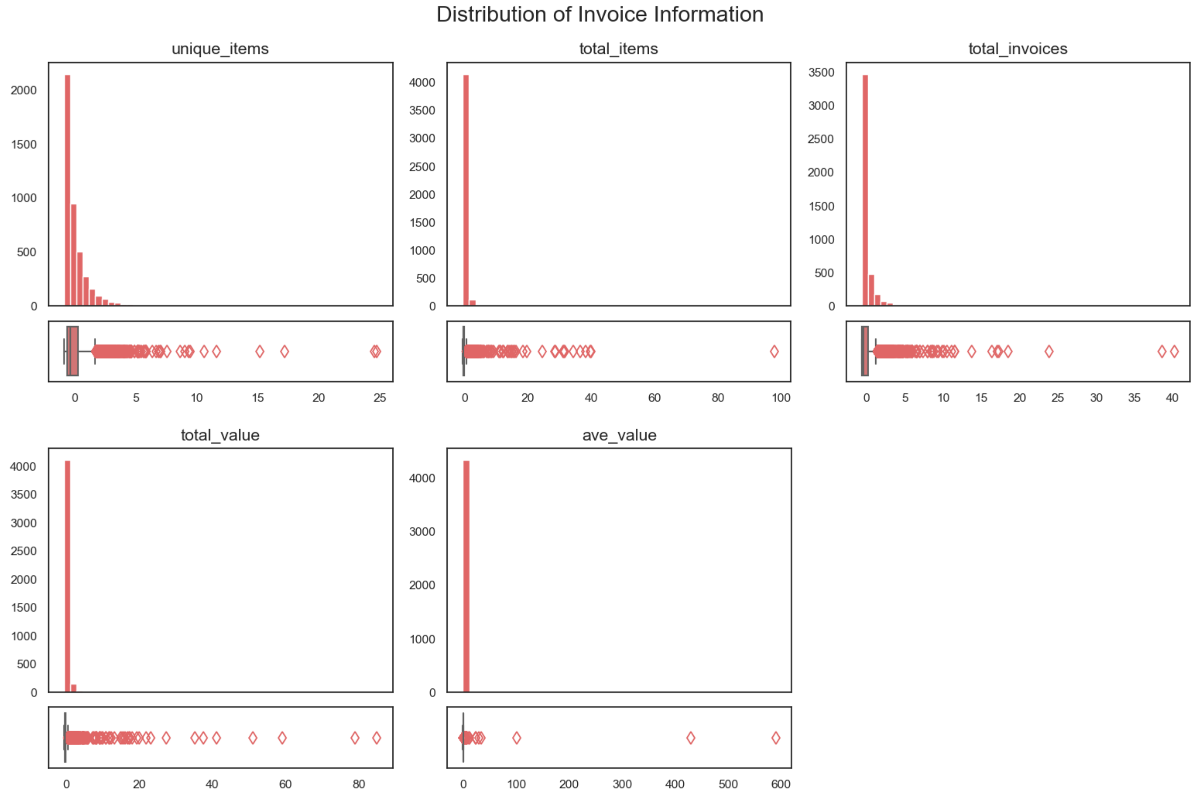

Figure 4 - Distribution of Invoice Information;

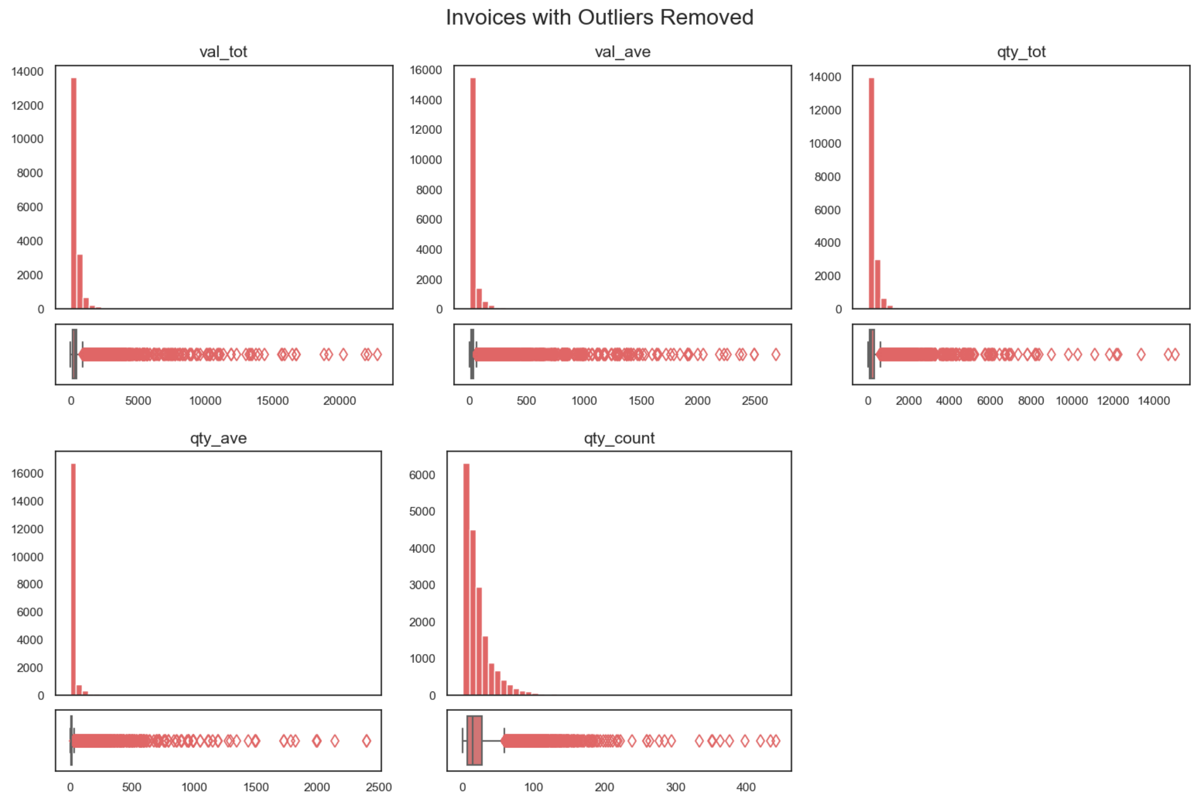

Figure 5 - Invoices with Outliers removed;

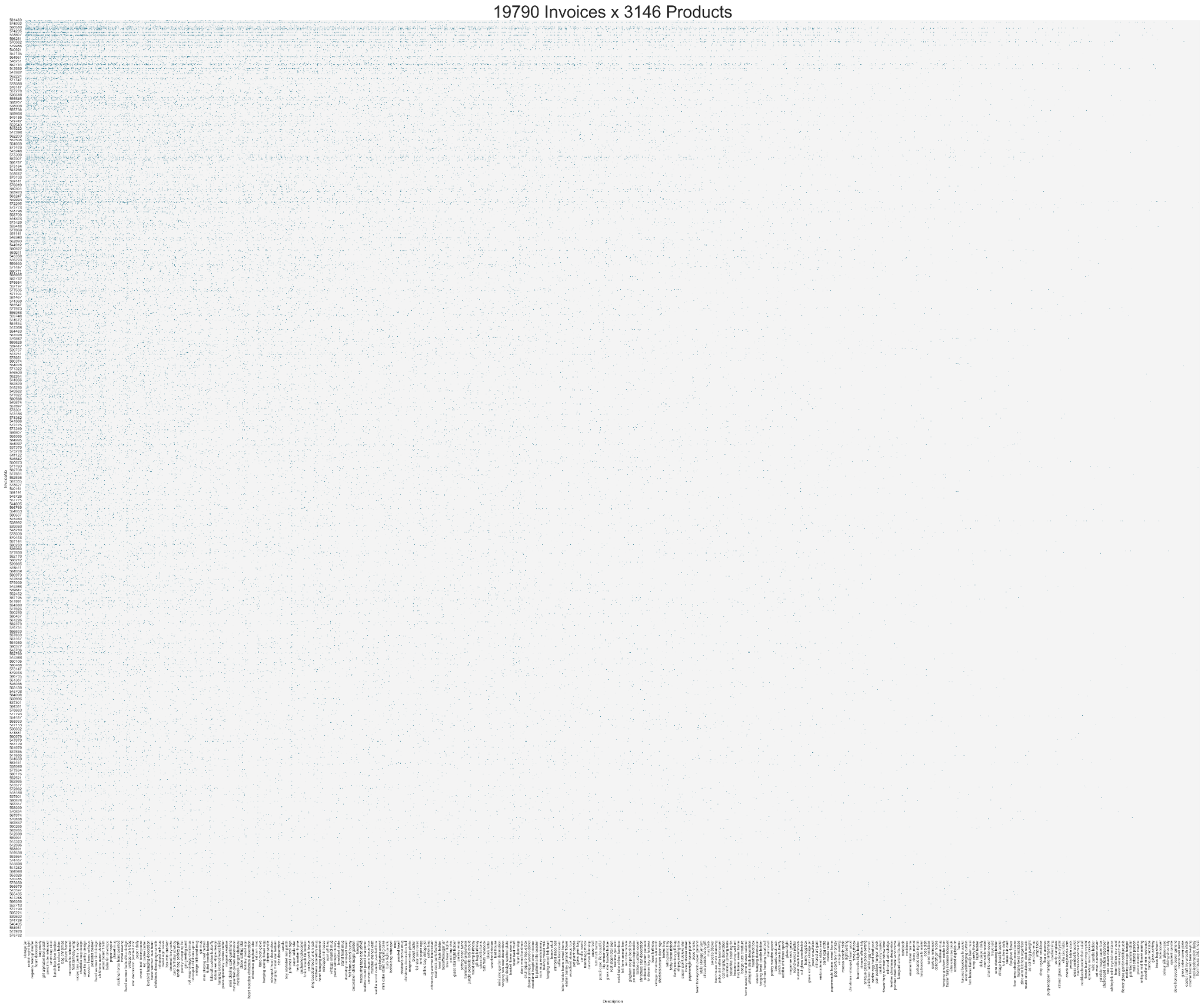

Figure 6 – Sparsity Matrix (heatmap of the invoices and the products purchased) after text processing;

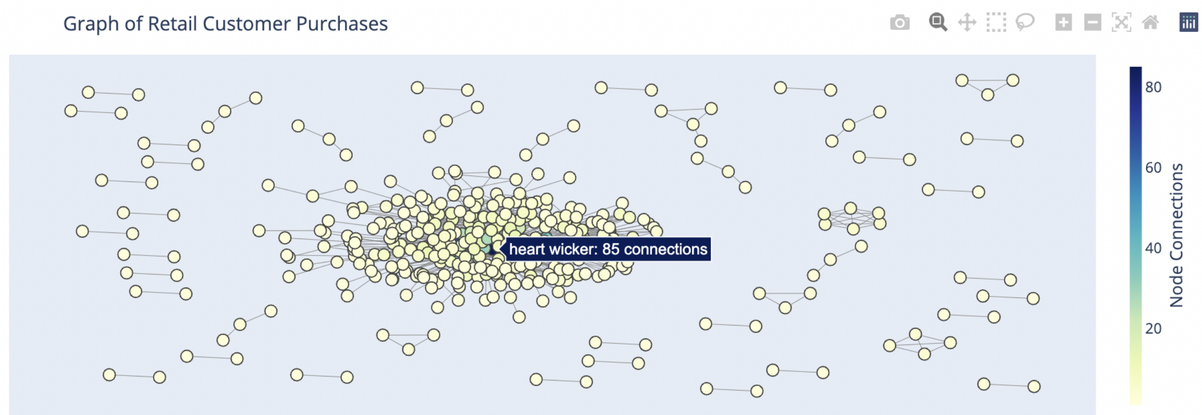

Figure 7 – Node graph of retail customer purchases;

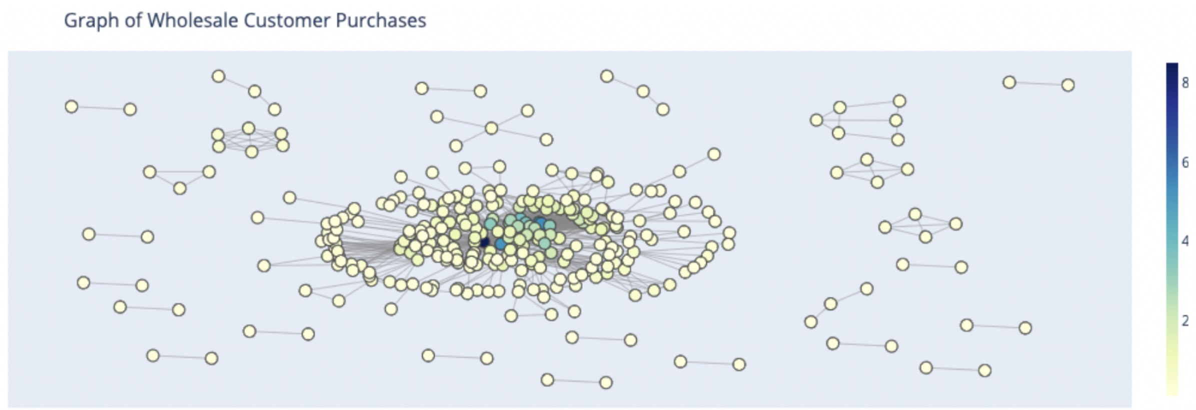

Figure 8 – Node graph of wholesaler customer purchases;

References

Raschka, S. (n.d.). *Association Rules.* Association rules - mlxtend. Retrieved April 18, 2022, from http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/

Retail sales, Great Britain. (2022, March 25). Home - Office for National Statistics. https://www.ons.gov.uk/businessindustryandtrade/retailindustry/bulletins/retailsales/february2022

Silveira, T., Zhang, M., Lin, X. *et al.* How good your recommender system is? A survey on evaluations in recommendation. *Int. J. Mach. Learn. & Cyber.* 10, 813–831 (2019). https://doi.org/10.1007/s13042-017-0762-9