NOVA
IMS
Information Management School

BUSINESS CASES WITH DATA SCIENCE

MASTER'S DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

Online retailer recommender system

Improve the customer experience and increase sales

Group Y

João César (20200669)

João Henriques (20200670)

Pedro Sancho Vivas de Castro (20200132)

Vilmar Bussolaro (20200268)

May 3rd, 2021

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

INDEX

1. INTRODUCTION

In this project we are asked to make a recommender system for ManyGiftsUK, a UK based company started in 1981 that recently shift to an online only retailer that focus on the sale on gifts that are unique and can fit all occasions. With this systems ManyGiftsUK wants to increase both sales and customer experience as a whole.

Recommender systems can be used by stores to increase volume of sales by showing to customers, products that they are more likely to buy, based on their previous purchases, like items that could otherwise never been found by the users. These systems help buyers make easier choices that are proven to help purchases altogether. Giant companies like Amazon have a large portion of their sales coming from the usage of recommender systems, which goes to show the importance that this system can have for a company.

The database we are studying contains all customer transactions that occurred between 01/12/2010 and 09/12/2011. It contains 25900 valid transactions, these transactions were associated to 4372 different customers, coming from 38 different countries, and 4070 unique items.

We are following the CRISP-DM methodology to achieve the final product, which is two Jupyter Notebooks, this report, and a presentation. We believe that this could entitle the enterprise on deploying two recommender systems, one for registered user and a second one for new unregistered users. Several recommendations and suggestions will be addressed in this report on how to maintain and improve the findings and systems generated by this assessment.

All information provided in this report, the presentation, and the respective notebooks with the recommendations for the company will be made available at: https://github.com/PedroSancho

2. BUSINESS UNDERSTANDING

2.1. BACKGROUND

ManyGiftsUK is a UK based company that after having physical stores since 1981, has recently decided to close them and shift its focus to online store only. The store sells mainly unique all-occasion gifts offering a high variety of products, over 4000 items.

Between 01/12/2010 and 09/12/2011, the company had 4372 different customers from 38 different countries. Since the enterprise decided on shifting all its operation to the online channel, there is a clear understanding that in order to recommend the best items for customers it needs to know the customer's profile and its purchase patterns. On that note, addressing both the issue of a recommender system for usual customers as well as the *cold-start problem* is paramount for ManyGiftsUK future growth prospect.

2.2. BUSINESS OBJECTIVES

This project has the objective of creating a recommender system that can increase sales and therefore increase profits. The business recognizes the importance of these systems to facilitate customers on their decision-making process and the importance on showing our customers new items from the wide selection available that otherwise the client could potentially not find. Therefore, it is important to both implement a general recommender system that can provide recommendation for all customers as well as to generate value through increased profits by its implementation.

2.3. BUSINESS SUCCESS CRITERIA

To be successful, the recommender system needs to be assessed, using an A/B testing approach, the variation (delta) in sales (units) and revenue between a subset of customers that will use the new system and those that will not, will be sufficient to evaluate its performance. This application will be considered a success if the result (yield) is positive and, at least, consistent over time, although we understand that it is possible to recommend some other things that could lead to performance improvements later on. Similarly, it will be a failure if it brings no results and no impact on revenue compared to the control group.

This assessment phase could be used as a Minimum Viable Product to generalize the possible impact on the whole operation of ManyGiftsUK, this could lead to adjustments prior to upscaling the suggested recommender system and therefore better yields.

2.4. SITUATION ASSESSMENT

For this project we have the retail.csv dataset, containing data collected by ManyGiftsUk, with their complete transactions in between 01/12/2010 and 09/12/2011.

The dataset contains 25900 valid transactions, all coming from 4372 different customers and 4070 unique items, having in total 541909 user/ item interactions. These are all customers who have purchased something from ManyGiftsUk in this period, one important fact is that we have a lot of returning customers. The customer transaction dataset contains 8 variables, from this feature set 6 are related to the invoice: invoice number, stock code, item description, quantity bought, invoice date and unit price. The dataset also contains 2 variables related to the customer, which are customer id and his country.

Our team is composed of 4 Data Scientists, which had 3 weeks to prepare a 5-minute presentation to the company's board, as well as a 10-Page Report (which you are reading) and the accompanying code.

2.5. DETERMINE DATA MINING AND MACHINE LEARNING GOALS

A recommender system based on customer purchasing habits can use supervised learning to classify items into elements to be recommended/not recommended or use unsupervised learning to for example make sense of the user-item feature space. In this case we had only implicit data, so we need to work with Unsupervised Learning.

But before bringing information to the surface, we need to do the basics: preprocess and understand our data. On feature selection, the goal is to choose a set of variables that is both broad and relevant but that do not generate the curse of dimensionality.

A particular problem that will have to take our special attention in this project is the cold start problem, this problem as to do with the lack of information on new users. In the following chapters we going to better explore it and coming up with a solution to it.

Our benchmark for this project is to see if we can recommend to all users of our website a better recommendation of products than coin toss suggestions, which is getting an area under the curve (AUC) value of 0.5. Another possible benchmark is to compare different algorithms to a naïve popular recommender, which recommends most popular items regardless of user preferences and customer purchase patterns.

2.6. PROJECT PLAN

1. Data Understanding
2. Data Preprocessing
    a. Outlier/Missing Input Treatment
    b. Same Product Variation Treatment
    c. Sparsity Treatment
3. Train Test Split
4. Recommender System Implementation
    a. Alternating Least Squares Algorithm
    b. Logistic Matrix Factorization Algorithm
    c. Bayesian Personalized Ranking Algorithm
    d. Naïve Popular Recommender Heuristic
5. Remarks on Explicit Data Implementation
6. Deployment
7. Maintenance

3. RECOMMENDER SYSTEM IMPLEMENTATION

3.1. DATA UNDERSTANDING

After taking a first look to our data, we found out the given dataset has 8 different variables. From these variables we framed the variable in two different types of values: (i) invoice related, that included the variables 'InvoiceNo', ‘StockCode’, ‘Description’, ‘Quantity’, 'InvoiceDate', 'Recency', 'UnitPrice'; (ii) user related, composed by ‘CustomerID’, ‘Country’;

Our dataset is composed of 541909 rows, on for each user/item interaction, the total sum of items sold is 5176450. There were 25900 purchases made from 4372 different customers, 92% of invoices are from returning customers, e.g., they have CustomerID, the following histogram shows the count of customers within a given invoice number interval. In it we can see that there is in fact a large portion of our customers that in the space of the one year studied made more than 100 purchases.

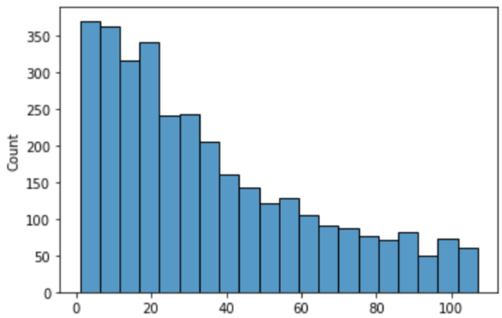

Figure 1: Histogram of Invoices per Customer

The team considered that it would be important to check for any patterns on customer purchasing habits behavior, since we only have data from 1 year it is not enough to prove seasonality but the spike on purchases on the end of the year is probably explained due to Christmas shopping for unique gifts for family and friends.

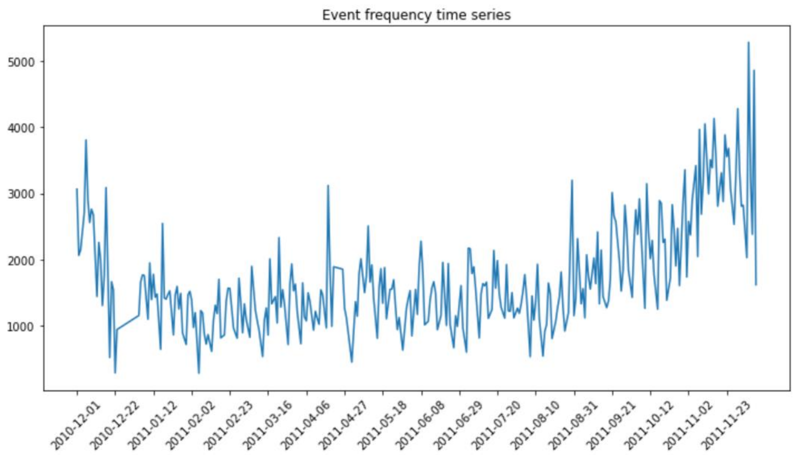

Figure 2: Invoice time series

3.2. DATA PREPARATION

After analyzing the above £200 items and seeing the names to try and identify potential errors and decided to remove some items, like ‘DOTCOM POSTAGE’, ‘Manual’, 'AMAZON FEE', 'Bank Charges', ‘POSTAGE’, ‘Discount’, ‘SAMPLES’, ‘Adjust bad debt’, ‘website fixed’, ‘wrong code?’, ‘mystery! Only ever imported 1800' and ‘found box'.

As the Null values on the Description feature represent less than 0.5% of the records, we decided to drop the correspondent records as well as the rows that presented negative quantities at that Invoice.

In consideration with Null values on the CustomerID there was not much that could be done about them, these Null values represent our new customers that still do not have an ID that could trace their customer preferences. We dropped these records and will consider them on a later stage when accessing the cold start problem separately.

On a last assessment, we transformed items that vary only in color into a single item, like a a blue mug and a brown mug with the same design is the same product after this preprocessing step. This led to an improvement of 20% in the sparsity criteria, e.g., 20% less sparse co-occurrence matrix, without removing any item rarely bought or user that had little number of purchases.

A last measure on reducing sparsity, was to define 2 thresholds for removing items and users from the matrix: items must be bought at least 10 times and users must buy at least 10 times as well, this led to a improvement of 30%.

3.3. ALGORITHM SELECTION

Our main objective is to build a Recommender System, to do so we based our work on Collaborative Filtering (CF) which is a method of making automatic predictions about the interests of a user by learning its preferences (or taste) based on information of his engagements with a set of available items, along with other users’ engagements with the same set of items. In other words, CF assumes that, if a person A has the same opinion as person B on some set of issues X={x1,x2,…}, then A is more likely to have B‘s opinion on a new issue y than to have the opinion of any other person that doesn’t agree with A on X.

To do this filtering, we must first start by doing a matrix factorization and to do so the model we'll use is an Alternating Least Squares (ALS), this model will allow us fit our data and find similarities within it. Additionally, we will proceed to test a few other algorithms in order to decide the one that can best recommend new items for regular customers of the gift company.

3.4. ALGORITHM IMPLEMENTATION

At the implementation phase, we divided our data into a train and a test though the split method, we decided to split the time series in the following way.

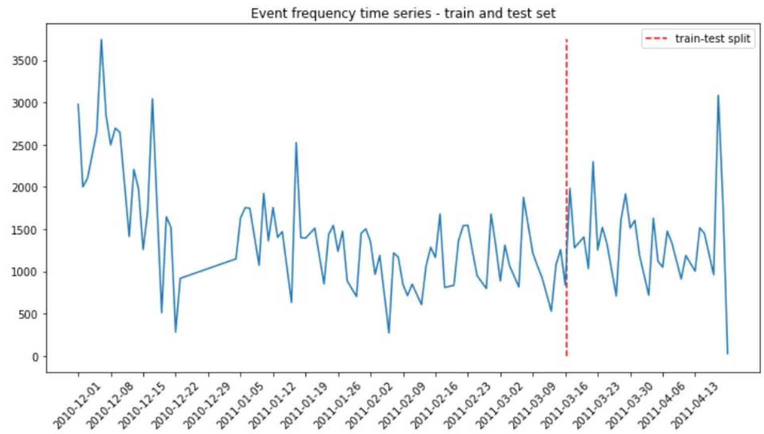

Figure 3: Invoice time series and train-test split visualization

After that two sparse matrixes of all the item/user/counts triples for the train set and test set were created, one for the train and another for the test data, these will be the matrixes used by our models.

3.5. DECISION CRITERIA FOR ALGORITHM SELECTION

The goal for the recommender system for registered users is to have a customer-centric recommendation for customers and therefore we wanted algorithms that could provide increase in sales, e.g., AUC greater than 0.50 and being an algorithm itself, not a heuristic method like the Popular Naïve Recommender. Therefore, we choose the Logistic Matrix Factorization algorithm (log_model) since it met the criteria above stated and had the best performance overall, when considering the customer-centric algorithms implemented.

	pop_model	als_model	bayes_model	log_model
precision	0.093329	0.062358	0.018645	0.096955
map	0.046369	0.026182	0.004523	0.043988
ndcg	0.106599	0.062092	0.015923	0.102256
auc	0.520663	0.510542	0.501581	0.520000

Figure 4: Comparison table for different recommender systems for Registered Customers

By deploying the same logic for the *cold-star problem* using the records that did not have CustomerID, and using InvoiceNo as a proxy to the purchasing patterns of first time users we decided to test whether different algorithms could be the Popular Naïve Recommender heuristic. As stated below, the heuristic that recommends the most prevalent items to all customers beats the other algorithms and it is our recommendation for future first time customers, since it has better performance and simpler implementation.

	pop_model	als_model	bayes_model	log_model
precision	0.281910	0.024793	0.174472	0.101010
map	0.137527	0.005813	0.076364	0.032473
ndcg	0.235188	0.018139	0.153950	0.080198
auc	0.511305	0.498188	0.505730	0.500787

Figure 5: Comparison table for different recommender systems on the Cold-Start Problem

4. RESULTS EVALUATION

4.1. DIFFERENT IMPLICIT ALGORITHMS FOR RECOMMENDATION SYSTEMS

Because the dataset we had is implicit, the scope of algorithms to be used is substantially narrowed. The *implicit* library on Python had different algorithms to tackle that problem. We implemented all suitable options from that library and compared results.

By using Collaborative Filtering (CF), the first algorithm we tested was ALS (Alternating Least Square). That is a matrix factorization algorithm and uses Alternating Least Squares. It is a two-step iterative optimization process. In every iteration it first fixes Products and solves for Users, and then it fixes Users and solves for Products. The Bayesian Personalized Ranking also is a matrix factorization algorithm similar to the ALS model and tries to arrive at a ranking by using the preference and confidence values in for the co-occurrence matrix. Like the previous two, we have the Logistic Matrix Factorization, that is a matrix factorization algorithm. This recommendation model learns probabilistic distribution, independently of the user liking the products or not.

Lastly, the Popular Naive Recommender is an algorithm that recommends specific products based on the products that are being bought the most at that time. It basically follows the trends in order to make naïve recommendations to all users irrespectively of their purchase patterns and consumption preferences.

4.2. USING LOGISTIC MATRIX FACTORIZATION FOR REGULAR CUSTOMERS

LMF is a probabilistic model for matrix factorization that uses collaborative filtering with implicit feedback. This makes us able to analyze the past behavior of each customer for a better prediction of how they will act in the future. Implicit feedback can be inputted from clicks, page views for every online platform and for example number of streaming of a music album or tv series. On our case, it gets the number of units ordered, the number of orders and the overall relation between the purchase patterns and the record of customer purchases.

We chose this model due to its simplicity of implementation, the capacity of parallelizing at a higher rate and it provides us the ability to get the probability that a customer will prefer a single item. Lastly, we understand that it holds a good possibility that this model may be able to improve its performance with time and more data being ingesting on the recommender system pipeline.

4.3. USING MOST POPULAR ITEMS FOR COLD-START PROBLEM

On the other hand, we have data from purchases made by users that are not even registered. These users can not be identified so we cannot get implicit feedback from them using CustomerID. We tested all the four models mentioned before using InvoiceNo feature as a substitute for CustomerID (user entity) and, as expected, the model with the best metrics for us to give recommendations to these users was the Popular Naïve Recommended Algorithm.

This algorithm is independent to the different data we get from each customer, and only retrieves the items that are more popular at the time. We would like to have a better solution for the customers that are not registered but for our conditions this is the best algorithm for recommendations related to the non-registered customers.

5. DEPLOYMENT AND MAINTENANCE PLANS

5.1. PLAN DEPLOYMENT

As we have two different types of customers, we will create a Deployment Plan for each of the two types.

Unregistered Customers

To address the cold-start problem as we explained before, we have no individual data regarding these customers. This makes it harder to implement a personalized recommendation model, which is our objective, in order to provide a better service to our customers and consequently a better relationship that will lead to higher loyalty and revenue.

For these customers, our first suggestion is to offer some sort of discount on the first order or a gift card for the second order when they first buy from us. This will make more people register and for the second option it will incentivize them to make a second purchase, that may lead for more purchases if we invest on a good platform and good conditions for our customers to shop online. Besides that, by registering even in the first order we can access data on where the purchase is being made and therefore recommend using the Popular Naïve Recommender segmenting for the customer's country.

Another suggestion will be to keep more data from those users, even if they do not register. Even if we do not keep their behavior throughout time, those orders can count to our implicit feedback in an independent level. Some data we could get is the country they live in and their age. Only these two variables would help tremendously on recommending for different areas and different ages.

A final suggestion, that may be hard to implement, is to try to get the process of storing the address the more objective way possible. This is: the street and number, floor and apartment. This could make us able to create a Dummy customer ID not for the customer himself but for the household/ZIP Code, so we can more approximately know the behavior of customers based on where they live throughout time. Additionally, ZIP Code data can be traced in order to obtain predictions on net income, number of children per household, customary bureau of statistics open data to source a segmentation policy through clustering.

Registered Customers

Regarding our registered customers our already implemented Logistic Matrix Factorization Algorithm can be improved, specially by increasing the number of inputs.

We suggest that information about the clicks, web visits, products visited but not bought and information like the address and city could be used as inputs in order to refine the model in the best way possible. Specially since to only the few information we currently have greatly undermines these unsupervised algorithms we proposed.

On that note, we cannot know for a fact if little number of purchases of certain products by customers means that they dislike the product or that they do in fact like it. It naturally could be the case that only when customers buy a great number of times the same product, we may infer that it shows a customer preference that we can extrapolate into a recommendation (main reason why we select 10 as threshold for reducing sparsity and trying to get strong signals).

General Recommendation

We also suggest an implementation of the reviews for the product listed, so we can get the perception the customers have from each product in a more explicit way. By having explicit data, a new paradigm of recommender systems could emerge and recommendations that combine both types of customer-item relationship data may prove to yield great results in sales and profits.

Naturally, this would improve our relationship with the customers since we would recommend the products also by their perceived quality and their preferences. To incentivize the success of obtaining reviews and the necessary data to implement this other type of recommender system, we believe there could be a compensation for the customers that review the products when they receive them. Iteration is paramount for adjusting costs of implementation of such policy without too much hampering of company's profits. A possible way to deploy this compensation could be given in some sort of loyalty points that would convert in discounts for future orders and therefore increase customer loyalty and frequency of purchases.

5.2. MAINTENANCE

In terms of Maintenance, we suggest a creation of a pipeline for this recommendation system, periodically sends new records for the pre-processing phase described, adds to the old data used on this assessment and makes a fine tuning of the hyperparameters of the already implemented Logistic Matrix Factorization Model, in order to maximize the area under curve (AUC) once again and possibly improving the performance of the recommendation system.

Also, the implementation of the suggestion above for the non-registered users regarding the discounts would lower the number of invoices without CustomerID and that will naturally source more records for the pipeline describe above.

Lastly, we believe that by implementing the targeted Popular Naïve Recommender using IP information to narrow the most popular items suggested by our heuristic method to only those most popular items purchased on the same country as the new customer seems something that could be assessed in the maintenance phase maybe through an A/B testing fashion. This could be a measure deployed with the goal of avoiding a quicker degradation of this heuristic method for suggesting products to new customers.

6. CONCLUSION

ManyGiftsUK provided us with a dataset with transactions from approximately 1 year of operations and asked our team to produce a recommender system that would add value to the company by facilitating customers purchases and therefore increase sales.

Data understanding and preprocessing was a crucial part of this project since it gave us a lot of valuable information about our customers and their preferences. To do the recommender system, we used Collaborative Filtering to recommend the best items possible and deployed a few algorithms made for Implicit Data.

Throughout the project we had to make the decision of splitting our strategy in two, one for registered customers and another for new customers. For the customers we had prior information about items purchased we used Logistic Matrix Factorization Algorithm and for new customers (cold-start problem) we decided that best practice would be to recommend the most popular items.

Lastly, the team provided the Recommender system that addresses both new and returning customers, this system can, nonetheless, be improved. Creating a review process, for example, that would allow buyers to evaluate purchased items will give us explicit data that would be helpful for a future project to try and improve the current recommender system.