# BUSINESS CASES WITH DATA SCIENCE

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

# BUSINESS CASE:

# PRODUCTS RECOMMENDATIONS

Euclidean Consultancy Group

Ernesto Madrid, m20190559

Fernanda Zippinotti, m20190232

Filipa Cerqueira, r2016677

Laura Castro, m20190269

May 2020

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

# INDEX

# 1. INTRODUCTION

Groceries and food stores are one of the most frequent stores due to families or individuals' necessities. *Bigbasket.com* was established as online grocery and food store in 2011 in India to provide a convenient service in 12 Indian fast-paced cities, offering more than 18,000 products.

For the high recurrence purchases and the fast-paced lifestyle of the urban cities, it is easy for the customers to forget some items on the weekly or monthly purchases. *BigBasket* identified an opportunity to increase their sales by addressing their customers with a recommendation system to individually offer each customer the most frequent or probable items to facilitate the online purchases and help the customers remembering some items they might be forgetting.

This Business Case applies Machine Learning algorithms to estimate these most likely to buy products for the *Bigbasket.com* customers by using a provided dataset with more than 8,000 transactions.

# 2. BUSINESS UNDERSTANDING

## 2.1. BACKGROUND

E-commerce and online sales had rapidly expanded over many markets due to the logistics and time saving convenience. Online grocery market is not the exception, in India, this market has grown at 12% between 2010 and 2015. As mentioned, *BigBasket.com* counts with a few years of experience in the online grocery market in India, and the company has identified that customers prefer to buy their groceries online to avoid traffic congestions.

About 30% of the BigBasket.com customers order trough smartphones due to the bad quality of the Internet connections or because of the convenience of the mobile devices. Nevertheless, it is not a quick task to do the online purchases, because the user should select every desired product from the vast variety available on the website, making an order to take 20- or 30-minutes and being in some way painful to navigate on mobile devices.

Another common issue is that customers forget items on their main orders, and for instance, those situations represents loss for BigBasket in both cases: when the customers decides to do a second small order with forgotten items in BigBasket or when the customer does not buy the forgotten items in BigBasket. In the first case, the double delivery process affects the logistics chain, and for instance there is an indirect financial impact, but in the second case, loss is obvious, since it can be assumed that the customer made the purchase of this forgotten items in another store.

## 2.2. BUSINESS OBJECTIVES

Being exposed the core issues related to the online purchases, this project aims to create a recommendation model that ease the customers buying experience and avoid loss by the forgotten items issue. Therefore, the main objectives of the business case are:

- Ease the buying experience for recurrent members.
- Increase sales by recommending an accurate purchase list to each member, each time.
- Decrease costs associated to second small deliveries (forgotten items), by suggesting possible forgotten products.

## 2.3. BUSINESS SUCCESS CRITERIA

In order to develop a deepen success criteria for any project, it is crucial to have a comprehensive understanding of the business processes and financial statements to associate how an accurate recommendation system can impact the Business. Nevertheless, with the provided datasets, it is possible to focus on developing a model that should:

- Increase the average order value due to the convenience of the recommendations.
- Engage customers with differentiating the service from competition by providing a better customer experience.
- Have a better control on retailing and logistics, for instance, decrease costs associated to unnecessary deliveries by recommending possible forgotten items.

## 2.4. SITUATION ASSESSMENT

To address this business case, Python 3 will be used for data analysis and model development, deploying the results via Jupiter Notebook. It is advisable to rely on this version and platform to explore the work done by our consultancy group, which can be easily executed on a personal laptop.

Nonetheless, to deploy a recommendation system for all customers, it should necessary to connect information systems in real time to feed the BigBasket.com website every time a member access his/her account. This part of is outside of the project scope, but despite of this, once the recommendation system is ready to use, in terms of costs, most of the changes should be only on a software level and hiring experts to maintain the systems. In the end, the costs for deploying this system are low in contrast to the benefits in terms of logistics and revenues.

## 2.5. DATA MINING GOALS

This project aims to deliver a Machine Learning model to feed a Content-Based Recommendation System with accurate products suggestion to each member. It is expected for the model to be able to predict with 40% of recall, this is 40% of the items in the customer's basket were recommended by the model. In this way, the reliability of the model is validated against actual orders placed by testing the model with orders placed after July 2014. From this partition, any random order for a particular customer is picked up for testing the products that were bought within a Monte Carlo Cross Validation evaluation.

Among all the evaluation metrics used to measure performance of recommendation models, recall was chosen to represent in the best way the most likely offer a money return to the business, since it represents the amount of the purchased items that are being present to the customer by the model.

## 2.6. PROJECT PLAN

To achieve the data mining goals and consequently the business goals, the following plan is considered as a guideline to guarantee a logical sequence on the project development:

1) An iterative process between Business Understanding and Data Understanding should be done to explore the data and determine the how to extract valuable information from the provided data.

2) The next stage should be the Data preparation and Modelling Techniques selection. This stage can have parallel tasks depending on the variety of models that are going to be executed. For this Business Case, more than 2 content-based recommendation models are planned to be executed, aiming to obtain the best results in terms of modeling performance.

3) Model Evaluation will be the step were comparison between models can be developed, nevertheless, this is also a review point to check the coherence of all models' results.

4) The key decision on the **Model Selection** is the next step after evaluating all results. This should not be time-consuming, but attention should be considered to guarantee a robust solution.

5) Once the models are developed, the final deliverables will be a Jupyter Notebook, with the descriptive analysis of all datasets and the evaluations of the models.

# 3. PREDICTIVE ANALYTICS PROCESS

## 3.1. DATA UNDERSTANDING

A descriptive analysis of the dataset provided was developed with the goal of better understanding the customer's shopping behavior. The dataset main insights are presented below. Further analysis was performed in the Jupyter Notebook, including insightful data visualization. The most important facts extracted from this dataset are the following:

- The provided records cover a time horizon from December 2011 until October 2014.
- There were 106 registered members and 8,386 orders.
- 1,727 different products are present on the transactions, represented by 215 categories.
- Distribution on purchases are similar along the days of each month. However, for the months in the middle of the of the year the distribution is lower.
- There are no first-time members in the database, the average number of orders by member is 79 orders.
- Members tend to buy 7 products per order, and there has been over 8,300 products sold.
- In average, at least 50% of the members' baskets are repeated.
- Purchasing behavior is similar throughout the week.
- 14% of the most sold products represent around 80% of the sales, in terms of quantity.

## 3.2. DATA PREPARATION

In the data preparation phase, a coherence filtering was done to exclude records with imputation errors and assure that the models are trained with valid records. Considering that some orders content only a few items, the following conditions were imposed:

- Orders done in the same day by the same member were aggregated, for these ones we could have a great confidence that products on the second purchase were forgotten products.
- Orders with less than 5 items were excluded from the dataset, considering that they could represent items that people forgot to buy given the small dimension of the baskets.
- Finally, a new feature called 'SKU_des' was created, which is the concatenation of SKU code and the category Description, just to have a better understanding a differentiation of results.

The train and test set were divided firstly by the date, as mentioned in the data mining goals. The test set holds the orders placed after July 2014. But the selection of each order to test were randomly done fallowing Monte Carlo cross validation schema.

## 3.3. Modeling

Based on the available data and the requirements of this business case, only content-based modelling methods were implemented, this means that all recommendations are based on self-content for each member, except for the recommendation to new customers that consider overall consumption from other customers. To ensure the recommendations are as accurate as possible tree methods were adopted:

- **Market Basket Analysis:** technique used to uncover associations between items. It looks at the most frequent product combinations to understand the purchase behavior.
- **Page Rank Algorithm:** technique that ranks the products bought by the customers measuring their importance, represented by the likelihood of the customer to buy each product. This is achieved by analyzing what each customer keeps buying in each basket as well as the products' co-occurrence.
- **Similarity Measures:** Measures that help identify the similarity among products, considering the different customer's baskets (Ricci et al., 2011). This way is possible to determine which products are more likely to be bought next given the current basket. Using one of the following measures: cosine similarity, Jaccard coefficient or dice coefficient.

To have a base line to compare the results from the methods adopted a model that create a random recommendation based on each customer consumption was created.

The following subsections in modelling stage are divided according the data availability to recommend products to customers. Firstly, the **Known Customers** subsection, the models will consider all transactions for a specific customer in the train dataset to make a recommendation. On the other hand, in the **New Customers** modelling subsection, the algorithms are not able to consider direct historical data from new customers and make recommendations, for instance, treatment is completely different. After present these two main differences, we present a subsection with the model for each problem - **Smart Basket** and **Did you forgot?.**

### 3.3.1. Known Customers:

For the known customers only the data of each customer was considered, as well as only the products purchased by each customer along historical transactions in the training dataset.

### 3.3.2. New Customers:

For new customers there is no historical data or any other behavior pattern to analyze. Hence, all customers' data was considered. Also, to be less computationally expensive and to include in our suggestions only the most bought products were included, cutting down the analysis from 1,727 products to only 200. For the did you forget problem, we include the top 200 products as well as the products that they have at the time in the basket.

### 3.3.3. Models for each problem:

#### 3.3.3.1. Smart Basket

Two algorithms were considered to solve the Smart Basket problem, the Page Rank Algorithm, and the Most Frequent products approach, nevertheless, these algorithms were developed to generate recommendations individually or by mixing results from both algorithms.

- **Page Rank Algorithm**

Considering all the products bought by the given member, a co-occurrence matrix is created. For each combination of 2 products a Relationship Coefficient is calculated (GFG, 2018). The rank of each product is given by the sum of the products of other product's ranks and their relationship confident. Given that ranks are unknown, this will represent a system of X equations with X unknown variables (with X being the number of products), using power iteration we get the values of each rank. At the end, the ranks are used for sorting the products and the top 20 are presented as the customer's smart basket.

- **Most Frequent Approach**

Considering all products bought for certain member, most frequent products are selected to create a recommendation list. After executing the algorithms, a maximum of 20 products are suggested. As both algorithms retrieve a list of recommended items, different combinations of the two list can be applied by taking different number of recommendations of each list and keep the repeated items.

#### 3.3.3.2. 'Did you forget?'

For each model, a function was created to return the most likely to be purchased products for each member, considering their basket and historical data.

- **Market Basket Analysis:** Firstly, a function was created that implements the Apriori algorithm for the data of each customer, with a lift threshold of 1.2. Which represents the observed support to that expected if X and Y were independent. The algorithm generates association rules, which returns products that are frequently purchased together. After the rules are calculated, the function prints the consequent products where the antecedent is in the basket. Meaning that it only retrieves products with higher probabilities of being purchased considering the customer basket. However, because there is a very large number of products, the association rules, with a lift higher than the threshold, didn't always contain the products in the basket. For this reason, if zero products from the basket were in the association rules antecedents, then description was used. This way, MBA is performed considering the products description, and the most sold product of each category, in the consequent association rules

- **Page Rank Algorithm:** The same algorithm used for the Smart Basket problem is here used as well. The only difference is that when the values of the ranks are found, the top products are returned excluding the ones that are already in the customer's basket.

- **Similarity Measures:** First a sample transactions' matrix is computed, showing whenever each product is present on each of the customer's transactions/baskets or not. After this it is possible to compute one of the provided similarity measures among each set of 2 products. Given this, the products that are not in the current customer's basket are ranked according with the highest similarity measure observed with the products that are currently in the basket. After this a top is selected and presents for this problem set. After running the algorithms, a maximum of five products are recommended on the check-out stage of the purchase.

## 3.4. Evaluation

In order to evaluate model performance and select the best model, the provided dataset was separated into two: train set (representing 90% % of the initial dataset, containing transactions until July 31st of 2014), and test set (representing 10 % of the initial dataset and containing data from 01/08/2019 until October 6th 2014). In this way it is possible to test each model performance, considering only historical data to train and provide results for the future.

The cross-validation methodology used was the Monte Carlo cross validation as exposed by (Rong et al., 2014) and (Chicco & Jurman, 2020), but with a few restrictions, since it is relevant for this problem considerer the time frame of the record. As the Monte Carlos cross validation suggest we set a variable for the number of splits to be made, in the notebook this variable is known as "times to run", in the report we will reference to is as *m*. The orders that are included in the test set are randomly selected and may repeat in different *m* values. As the validation for every customer would be computational expensive, we also have a variable referred here as *n* that is the number of customers to be randomly select in each *m*. Finally, as mentioned, the test and train set are not randomly split as we take in consideration a specific time frame for both, which is not always the case in Monte Carlos cross validation.

Therefore, specific functions for validation purposes were design to comply the Monte Carlo cross validation conditions: First, *n* random members are selected from the test set. For each member, a random order code (transaction) is then selected to recommend products to this specific member, and finally, the recommended items are then compared with the real purchased items. This is made *m* times. The values of n and *m* are present in the metric results table.

- **Smart Basket:** Using the train set, the smart basket is suggested for each customer, no matter the order selected. After this the products in the smart basket and in the actual customer's basket in the selected order are compared using different evaluation metrics described below, but mainly the recall.

- **Did you forget:** For that certain order, a random number of items from 10% to 50% of the basket are took out to simulate items that that customer would forget. After this, this smaller customers' basket (without the items that the customer supposedly forgot) and train set are used to provide results, which are later compared with the items hold back using different evaluation metrics described below, but mainly the recall.

**Metrics used in the evaluation:**

- Precision: Rate between True Positives and Predicted Positives (True Positives + False Positives). This is, percentage of items that the model got right comparing with all items it has suggested.

- Recall: Rate between True Positives and Real Positives (True Positives + False Negatives). This is, percentage of items of items that the model got right comparing with the customer basket's items. Being this the most important metric in terms of business, because it reflects if the ratio of well recommended products.
- F1: A composition of precision and recall.
- Regular shop: The number of products in the customer purchased that are in the regular products list, which is the 50 products that represent 50% of all products sells.
- Regular Recommendation: The number of products in the customer recommendation that are in the regular products list. This is used to be compared with the regular shop metrics to evaluate if the models are recommending only high overall frequent product instead of personalized recommendation.

Table 1 shows all the metrics for the Smart basket problem considering old customers and recommending 20 products. These results were obtained evaluation 50 different order from different customer 20 times.

| Model | Precision | Recall | F1 | Reg. shop | Reg. recom |
|---|---|---|---|---|---|
| Random | 0.068 ± 0.073 | 0.132 ± 0.129 | 0.084 ± 0.082 | 0.454 ±0.22 | 0.283 ± 0.114 |
| Page Rank | 0.223 ± 0.151 | 0.435 ± 0.220 | 0.280 ± 0.151 | 0.454 ±0.22 | 0.619 ± 0.143 |
| Most Frequent | 0.225 ± 0.147 | 0.444 ± 0.219 | 0.284 ± 0.148 | 0.454 ± 0.226 | 0.638 ± 0.148 |
| Mix (page rank + most frequent) | 0.223 ±0.151 | 0.436 ± 0.219 | 0.280 ± 0.15 | 0.454 ± 0.226 | 0.622 ± 0.14 |

**Table 1.** - Metrics results for tested models with *n* = 50 customer and *m* = 20 times.

The chosen model was Mix model, the reason being that the results were not much different from page rank and most frequent and the mix model, but the mix model has a component of randomness to it that is interesting to keep the diversity of the recommendation.

Table 2 shows all the metrics for the Did you forget problem considering old customers and recommending 5 products. These results were obtained evaluation 10 different order from different customer 5 times. Also, there is no regular shop and regular recommendation in this evaluation since those forgotten items were randomly select from the whole purchased.

| Model | Precision | Recall | F1 |
|---|---|---|---|
| Random | 0.048 ± 0.109 | 0.0618 ± 0.136 | 0.0484 ± 0.098 |
| Page Rank | 0.212 ± 0.22 | 0.220 ± 0.217 | 0.195 ± 0.174 |
| Jaccard | 0.144 ± 0.192 | 0.170 ± 0.245 | 0.144 ± 0.180 |
| Cosine | 0.136 ± 0.185 | 0.160 ± 0.237 | 0.133 ± 0.170 |
| Dice | 0.144 ± 0.192 | 0.170 ± 0.245 | 0.141 ± 0.180 |
| MBA | 0.136 ± 0.146 | 0.202 ± 0.234 | 0.152 ± 0.166 |

**Table 2.** - Metrics results for tested models with *n* = 10 customer and *m* = 5 times.

The chosen model was Page rank because is the one with the higher recall.

Table 3 shows all the metrics for the Smart basket problem considering new customers and recommending 20 products. These results were obtained evaluation 50 different orders from different customer, 20 times. The results in the random model are the ones from the old customers model because for the new customer the results for randomness will be much worse. Based on results, the page rank model performed better than the random model, so it was chosen to address this problem.

| Model | Precision | Recall | F1 | Reg. shop | Reg. recom |
|---|---|---|---|---|---|
| Random | 0.0680 ± 0.073 | 0.1322 ± 0.129 | 0.0849 ± 0.082 | 0.454 ± 0.22 | 0.283 ± 0.114 |
| Page Rank | 0.125 ± 0.089 | 0.366 ± 0.245 | 0.177 ± 0.114 | 0.634 ± 0.256 | 1 |

**Table 3.** - Metrics results for tested models with *n* = 50 customer and *m* = 20 times.

Table 4 shows all the metrics for the Did you forget problem considering new customers and recommending 5 products. These results were obtained evaluation 30 different order from different customer 10 times.

| Model | Precision | Recall | F1 |
|---|---|---|---|
| Random | 0.048 ± 0.109 | 0.0618 ± 0.136 | 0.0484 ± 0.098 |
| Page Rank | 0.142 ± 0.177 | 0.118 ± 0.159 | 0.122 ± 0.155 |
| Jaccard | 0.238 ± 0.351 | 0.184 ± 0.233 | 0.192 ± 0.244 |
| Cosine | 0.244 ± 0.351 | 0.190 ± 0.234 | 0.198 ± 0.244 |
| Dice | 0.238 ± 0.351 | 0.184 ± 0.233 | 0.192 ± 0.244 |
| MBA | 0.189 ± 0.419 | 0.143 ± 0.258 | 0.147 ± 0.260 |

**Table 4.** - Metrics results for tested models with *n* = 30 customer and *m* = 10 times.

The chosen model was Cosine because is the one with the higher recall.

## 4. RESULTS EVALUATION

The final goal of this Business Case was to provide a recommendation model able to suggest 40% of customers' needs to mitigate customers loss, logistical costs, increase customers experience and average order value.

Regarding the Data Mining goals, the selected model achieves 40% of recall for known customer and gets very close to achieve similar results for new customers. And for the '*Did you forget problem*?', the models were able to surpass random suggestions, as in (Longo, 2018) comparison for evaluation.

## 5. DEPLOYMENT AND MAINTENANCE PLANS

With the results from the analysis it is possible to recommend products to the members, however it will only add value to the company after the insights gathered are regularly available to end-users, for this, it is imperative to deploy the model simply and efficiently.

It is necessary to provide an easy way to deploy it, so it can be used frequently on new data. With the functions created, the problem *'Did you forget?'* only requires the products in the customers basket. As for the *Smart Basket* it only requires the member's number.

Every time a known customer places an order the recommendations will likely improve because there will be more data to look at, for this reason the model should be deployed for every order. The same applies for *Smart Basket*.

As for new customers, the *'Did you forget?'* problem will also need to be deployed for every order since their results depend on the products in the basket. However, for the *Smart Basket* since they don't have their own historical data, the results will be the same. And so, the model only needs to be updated every week months to account for seasonality products or changes in the overall consumer behavior.

On overall, these models are close related with the Information Systems of the company to feed the models, the IT department should be highly involved on the deployment of theses models. As changes should be done on the website and the mobile app to display the suggested items, web and apps developers should work on behalf with the Marketing Department to show the suggested items in the most practical and attractive way for the customers. As it is expected to increase sales with the deployment of this recommendation system, it is advisable to track sales and asses the changes over time, because of that, the Financial Department should be involved since the beginning.

Considering the maintenance plan, the database should only have products that are still being sold, so it's important to update it every time a product is no longer sold, or a new product is added. Additionally, customers patterns change, so it is best to only consider the five most recent years of each customer. This way old products that are no longer purchased by the client will not affect results.

# 6. CONCLUSIONS

With the results from this project, the company will be able to provide to the customer a more seamlessly experience, which should result in the increase of sales and customer loyalty.

From all the "Did you forget?" problem, it was found that the one with the smallest prediction error, was cosine similarity-based recommendations for the new customers, and Page Rank for the known customers. And for the "Smart basket" problem, it was found that the one with the smallest prediction error, was page rank for the new customers, a mix of page rank and most frequent item for the known customers.

## 6.1. CONSIDERATIONS FOR MODEL IMPROVEMENT

The dataset only provides information regarding the orders, ideally data about the client should also be provided. With this additional information the *Smart Basket* and *Did You Forget?* Problem for new customers would be a lot more accurate. This because with techniques such as clustering, it would be possible to group the new customers with other clients more similar to them, and with probably more similar shopping patterns too.

Besides demographic data, the predictions would also improve if the monetary value from transaction was included. Customers preferences vary considering the products price. For example, in a week where product C is in sales, customers are more likely to buy it instead of a substitute product with no discount, even if C is not the one, they usually buy.

Additionally, the stock of the products should also be added in the dataset. By recommending a product that is not available the customer might be unsatisfied. Meaning only products available stock should be recommended.

## 7. REFERENCES

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 1–13. https://doi.org/10.1186/s12864-019-6413-7

GFG. (2018). *Page Rank Algorithm and Implementation - GeeksforGeeks*. https://www.geeksforgeeks.org/page-rank-algorithm-implementation/

Longo, C. (2018). *Evaluation Metrics for Recommender Systems - Towards Data Science*. https://towardsdatascience.com/evaluation-metrics-for-recommender-systems-df56c6611093

Ricci, F., Rokach, L., Shapira, B., Kantor, P. B., & Ricci, F. (2011). Recommender Systems Handbook. In *Recommender Systems Handbook*. https://doi.org/10.1007/978-0-387-85820-3

Rong, Y., Wen, X., & Cheng, H. (2014). A Monte Carlo algorithm for cold start recommendation. *WWW 2014 - Proceedings of the 23rd International Conference on World Wide Web*, 327–336. https://doi.org/10.1145/2566486.2567978

## 8. APPENDIX: RFM ANALYSIS FOR DELIVERY CUSTOMERS

As an extra analysis was decided to include an RFM analysis for members in the dataset. With RFM analysis is possible to quantify customer behavior and decide the best way to target them with specific marketing campaigns, increasing response rates. This is done by evaluating the customer's recency, frequency and monetary. As Monetary value was not available, we used information we decided to translate this variable into Quantity, measuring the total amount spent by customer.

After calculating such variables for each customer, in order to further perform K-means, given this algorithm's sensibility, outliers were also extracted. At the end, was possible to conclude that the store's customers could be divided in 3 segments:

**Frequent High Orders Members (Keepers):** Come frequently and purchasing the biggest amounts of products. In average they have not return for more than a week, this can be explained by their shopping behavior. There were 18 members, 17% of the members.

**Frequent Members**: In average they have not done purchases for 3 weeks, coming frequently and purchasing smaller quantities of products. There were 45 members, 42% of the members.

**Lower Quantities Members:** In average they are costumer that have not purchases in 3 weeks, the ones that came the biggest amount of times but purchasing smaller quantities of products. There were 42 members, 40% of the members.