Big Data Analytics: "Building Personalized Recommendation System for Banking Products"

Albert Bastian – albert.bastian13@student.xjtlu.edu.cn

Abstract

Recommendation system is an important medium used by company to promote their products to each of its important customer. Therefore, an accurate and personalized recommendation is needed because each individual could have different needs and characteristics. Similarly, this project (i.e. *originated from Kaggle's competition)* also focuses on building an efficient recommendation system from a past customer records of a famous financial company named *Santande*r [1][2]. In order to build a recommendation system, several work steps and methods implementations related to Big Data Analysis are required. This includes *data exploration* (e.g. Exploratory Data Analysis (EDA)) and *data modelling* (e.g. building a predictive model to generate prediction values).

1. Objective

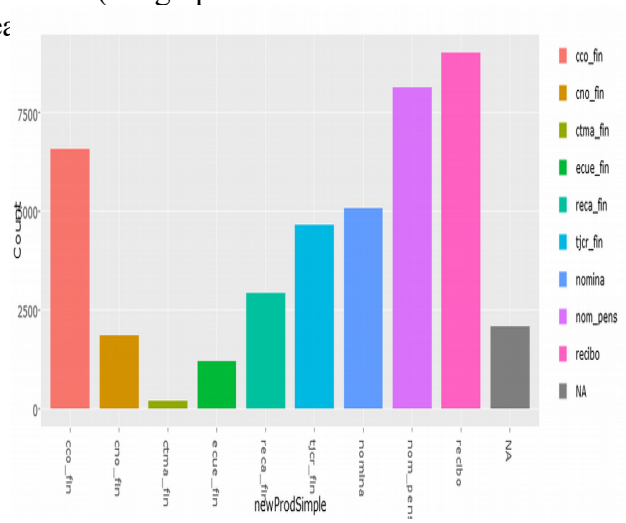
The goal of Big Data Analytics is to produce a useful information from processing a huge amount of data (e.g. Customer's transaction history). This project goal alone will specifically focus on building a predictive model for a product recommendation system to produce a prediction from a given data. The final accumulated result of this prediction will then be used to determine which product recommendation will suits the best for their (*Santander)* customer. This is done by processing the raw data using a data *model* such as "*xgBoost*" which syntax is predefined within its package.
In short, the main goal is to produce a prediction on which *new product* will be purchase on *June 2016 based on the previous customer record* (i.e. the given dataset) that consist 16 different months past record (from February 2015 to May 2016) [3].

2. About The Dataset

The data (i.e. customer's record) used in this project mainly comprises from approximately one million customers [3]. Specifically, it contains detail on both each individual personal data (e.g. age, gender, etc.) and various products (in total there are 24 products such as *credit card or debit card payment, pensions fund, mortgage, and tax*) ownership that each of the customer own in 16 different months (i.e. from February 2015 to May 2016) [3]. In addition, this provided data is split into two parts which are '***train data'*** for feature generation (e.g. finding product-to-product correlation/relationships or customer-to-customer ones) training the data with a *specified model (e.g. XgBoost)* and '***test data'*** for the final validation (i.e. used in the comparative process with the '*train data'* to produce the final prediction of each respective products) [3].

3. Data Exploration

An initial data analysis through data cleaning and visualization is an important step to have a proper understanding on the raw data. This process is also known as *Exploratory Data Analysis* (EDA) [4]. In this case, *R programming language* is use to implement the EDA task (i.e. graph visualization and data clea



Top 9 Products (name)
Current Account (cco_fin)
Payroll Account (cno_fin)
Mas Particular Account (ctma_fin
E-account (ecue_fin)
Taxes (reca_fin)
Credit Card (tjcr_fin)
Payroll (Nomina)
Pensions fund (nom_pens)
Direct Debit (recibo)

Figure 1. Top 9 most commonly products owned by customer in the 'train data' (based on number of count alone)

The above graph only shows an overall view of the most significant or commonly owned products from all customers. Therefore, a deeper analysis is still needed to show a more relevant information that is needed to make the final prediction (i.e. prediction on which new product will be purchased on June 2016). Additionally, the graph below shows further relationship between new product purchase with its respective occurrence time (i.e. in months).

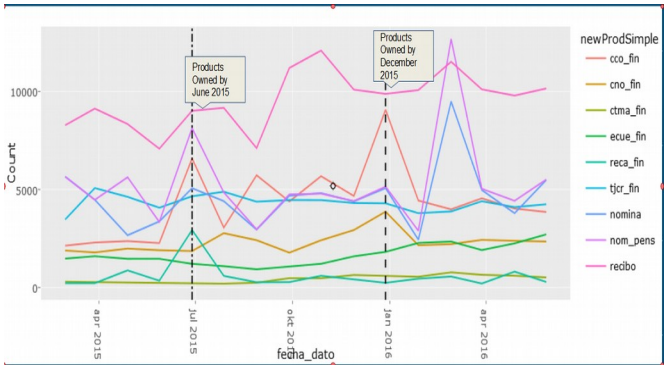


Figure 2. Newly purchase product counts (based on time) for the Top 9 product in 'train data'

As shown above, there are a number of interesting pattern that can deducted as a crucial insight that would later on be used for the base model features. However, the main focus here will be the two special months (i.e. June 2015 and December 2015) that shows a particular unique distribution for two products which are *current account (cco_fin)* and *taxes (reca_fin)*. This insight will be proven later on to be the best indicator for our prediction on future new product purchase on *June 2016 (i.e. the main goal of the project)*. Thus, some modeling restriction are possible in the data modeling process to improve the time efficiency of the data training process without compromising the result's accuracy *(this will be explained later on the base modeling part)*.

In addition, there also exist some product-to-product correlation obtained by using the *Pearson Similarity Measurement* [5]*:*

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_u)^2}}$$

where 'i' and 'j' are two similar items (or products) viewed as two different vectors to determine similarity between both as the angle between these vectors.

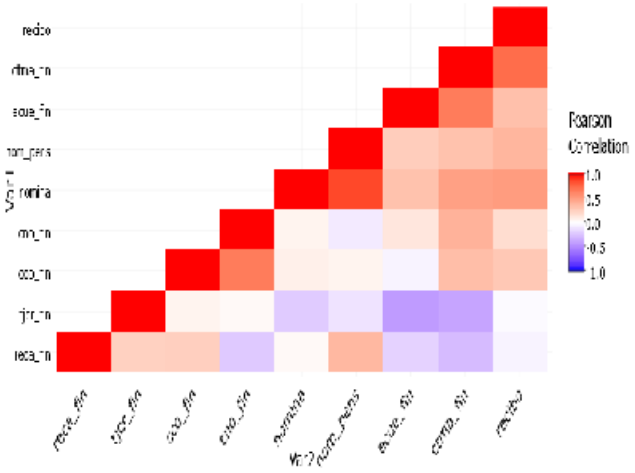


Figure 3. Top 9 products correlations/similarity mappings

The above graph representation shows a significant relationship or correlation between two products which are *pension fund (nom_pens)* with *payroll fund* (nomina). This correlation happens because these two product are often purchased together.

Moreover, all of these different approach on data exploration will be applied into the strategy or approach on building the predictive model that is consist of several different base model that is classified in sixteenth different months of data.

4. Data Model (XgBoost)

The model used in this work is *xgboost model* which is a popular model known for its versatility and fast processing speed [9]. It is considered to be versatile because the model implements two different models which are linear model and decision tree model. Therefore, it supports a variety of different functions such as *linear regression, classification, and ranking* [9]. In addition, it has a fast processing time because it has the capability of performing automatic parallel computation [9].

4.1 Base Model and Feature Generation

In order to produce an actual prediction, a base model needs to be constructed based on the past records of 16 months (i.e. February 2015 to May 2016) data for all existing 24 products (e.g. credit/debit card) for feature generation. To be exact, an *xgboost model* for each 24 products from 16 different months will be used as the base model. For instance, data in February 2015 is used to train a specific product such as "*reca_fin (i.e. taxes)* and the remaining 15 month lags is also used for the data training (March 2015 to May 2016). However, this *greedy approach* are proved to be ineffective since it takes to much process time (*approx. 1 week to process with a high spec machine*) in producing the base model predictions. Therefore, a better or more efficient approach are needed for the *feature selection*. This is done by restricting some base model on a specific month lag only. For instance, the model for product named *current account (cco_fin)* is predicted only by the December 2015 record. This restriction is done on the basis of relative base model weight of each product on 16 different months. The evaluation on relative base model weight of each month lags that corresponds to all products (February 2015 to May 2016) will determine the significance of the corresponding months (e.g. the record/data on June 2015 for *taxes* holds a large relative weight value which indicates that June 2015 holds a significant influence on the product behaviour).

Product/Lag	3	4	5 Jun15	6	7	8	9	10	11	12	13	14	15	16 May16
Cco_fin	0.3	0.3	13	0.3	0.3	0.3	0.3	0.3	9	0.3	0.3	0.3	0.3	0.3
Ctma_fin	0	0	0	0	0	0	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5
Ecue_fin	1.2	0	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5
Reca_fin	1.2	1.3	52	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5
Recibo	0	1.3	13	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5
Other	1.2	1.3	13	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5

Table 1. *Relative lag/month weights for top purchased products (1st and 2nd lag month set to 0 or excluded since both does not have such significance) produced with simple weighted average of all base model predictions*

4.2 Final strategy on Base Model Combination and Feature Generation

As previously mentioned, to improve the effectiveness of the base model classifier in purpose for the *feature generation* only a few month lags data are chosen from the whole lag months (i.e. 16 *months).* In addition, some *products* out out all 24 products will also be dropped in the data training (base *modelling and feature generation*). Firstly, the base model (created with the *train* data) are constructed based on:

- 20 different products data except "savings account (*ahor_fin)",

"guarantees (aval_fin)", "short-term deposits (deco_fin)", and "medium-term deposits (deme_fin)" which is set into a fixed value of 1^-10 since it does not hold much significance over the overall prediction result.

- Product of *current account ("cco_fin")* is trained (or predicted) only with a single month lag of *December 2015 (2015-12-28)* because it shows a unique product distribution on this month.
- Another product named *taxes ("reca_fin"*) is also trained with a single month lag data of *June 2015 (2015-06-28*) data, since J*une marks a special month for tax paying*.
- Moreover, the remaining probability (or predictions) of other products (approx. 18 products) is predicted with a lag months data ranging from December 2015 (2015-12-28) to April 2016 (2016-04-28) which represents data that contains a list of new purchased products

While the feature generation or classification strategy are based on:

- *The base features were made from user's data except from these attributes table partition for a column*("fecha_dato"), customer's code("ncodpers"), *date of first customer registration*("fecha_alta"), customer resignation date("ult_fec_cli_1t"), *address type* ("tipodom"), and province code ("cod_prov").
- Grouping or coupling of product ('ind_actividad_cliente') and its last month value to generate one feature.
- Group all last month lag (May 2016) values of all included 20 different products data *except* the previous excluded product ("savings account (*ahor_fin)", "guarantees (aval_fin)", "short-term deposits (deco_fin)", and "medium-term deposits (deme_fin)"*) which has a positive flank or new products indication.
- Set all previously made features from 20 different products as a character variable(1 feature)
- The total number of purchased products in the last month of May 2016 (1 feature)
- Make binary classification from index change pattern of the 20 remaining product purchase(e.g. 0 to 0, 0 to 1, 1 to 0 and 1 to 1) from the first month lag (February 2015) to the last one (May 2016).
- Counts the length of continuous '0' index from the *purchase behaviour* from all month lags (February 2015 to May 2016) of 20 products.

From this approach, 20 different base models is produced to process 20 different products. In addition, the validation dataset (or test data) is also split into 20 different separated parts. Each of it corresponds to each respective base models. The evaluation on each base model is done with the *Area Under Curve (AUC)* metric to determine the base model classifier performance (i.e. accuracy) on validation data (or test data) as shown below [7]:

$$P\Big(\text{score}(x^+) > \text{score}(x^-)\Big)$$

where a classifier will only rank based on a randomly chosen positive example rather than the negative ones [7].

In addition, *logarithmic loss (logloss)* metric is also used to evaluate the base model created from base model combination that has already omitted all negative flanks (i.e. records which has now new products purchase) [7][8].

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}\log(p_{i,j})$$

The above formula is a multi-class version of *logloss* metric, where 'N' is the number of observations, 'M' is the number of class labels,

log is the natural algorithm, 'Yi,j' has value '1' iff observation *'i'* is in class '*j', while* the value will be '0' if it's the opposite. In addition, '(Pi,j)' is the predicted probability that observation *'i'* is in class *'j'* respectively [7][8].

5. Post-processing and Final Submission

- The post-processing is done with some normalization on each prediction (or probability) value produced by each of the base model (i.e. 20 different models based on the chosen month lag). This is done by transforming each of the prediction value by increasing each value into an exponential value. This is important for the final submission because public leaderboard scores does not translate directly into positive leaderboard counts
- The final submission is done by combining all generated prediction values from each base model. All of these accumulated values will be then evaluated with the 7 criterion *Mean Average Precision (or MAP@7) metric* for the final submission as shown below [7]:

$$MAP@7 = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{1}{min(m,7)}\sum_{k=1}^{min(n,7)}P(k)$$

where |U| is the number of rows *usersintwotimepoints*, 'Pk' is the precision at cutoff 'k', 'n' is the number of predicted products, and m is the number of added products for the given user at that time point. If 'm = 0', the precision is defined to be '0' [7].

6. Conclusion

To sum up, the process of building a personalized recommendation system can be divided into these 3 main tasks:

- Exploratory Data Analysis (EDA), learn to understand the data by visualizing raw data.
- Data classification, learn to group individual data variable into smaller groups for better categorization that would be used for the next stage of data training for *base model generation*.
- Base Model Generation, choose the appropriate base model (e.g. Xgboost) and produce the predictions based on the chosen classified data (i.e. process/train the chosen data).

In addition, there are two main learning points from all of the above work process which are:

- Understanding customer purchase behaviour based on product ownership history
- and Data Training based on accurate data selection/classification

References

[1] Kaggle, Inc. "Your home for data science," 2016. [Online]. Available: https://www.kaggle.com/. Accessed: Sep. 19, 2016.

[2] Wikipedia (2016), "Kaggle". URL: https://en.wikipedia.org/wiki/Kaggle. (Accessed 19 September 2016).

[3]Kaggle,Inc."SantanderProductRecommendation, "2016.[Online]. Available: https://www.kaggle.com/c/santander-product-recommendation. Accessed: Sep. 19, 2016.

[4] Santander,“AboutTheGroup,”2016. [Online].Available: http://www.santander.com/csgs/Satellite/CFWCSancomQP01/en_GB/Corporate/About-The-Group.html. Accessed: Sept. 19, 2016.

[5] Marko, B. and Yoav, S. (1997). Fab: Content-based, collaborative recommendation. Communications of the Association for Computing Machinery, 40(3):66–72.

[6] Daniel, B. and Michael, J. P. (1998) Learning collaborative information filters. In Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98), Madison, WI. Morgan Kaufmann. pp. 46–54.

[7] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. (1999). An algorithmic framework for performing collaborative filtering. In Proceedings of 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 230–237, Berkeley, CA. ACM Press.

[8] T. Hastie, R. Tibshirani, and J. Friedman. (2001). The Elements of Statistical Learning. Springer Series in Statistics. Springer-Verlag, New York.

[9] V. Kumar, M. V. Joshi, E.-H. Han, P. N. Tan, and M. Steinbach. High Performance Data Mining. In *High Performance Computing for Computational Science* (VECPAR 2002), pages 111–125. Springer, 2002.