Santander Product Recommendation

Lakshendra Singh, Nishant Oli & Suparna Ghanvatkar

December 3, 2017

Abstract

A challenge in Kaggle to predict which products Santander Bank's existing customers will use in the next month. Santander Bank, is a bank which wholly owned subsidiary of the Spanish Santander Group. We have been given 1.5 years of customers behavior data from Santander bank to predict what new products customers will purchase.

1 Introduction

Santander Bank is one of the North Americas top retails banks by deposits and a wholly owned subsidiary of one of the most respected banks in the world: Banco Santander. Its parent company, Santander Group, serves more than 100 million customers in the United Kingdom, Latin America, and Europe.

Santander Bank offers a lending hand to their customers through personalized product recommendation to support needs for a range of financial decisions. Under their current system, a small number of Santanders customers receive many recommendations while many others rarely see any resulting in an uneven customer experience. The challenge here is to predict which products their existing customers will use in the next month based on their past behavior and that of similar customers. With a more effective recommendation system in place, Santander can better meet the individual needs of all customers and ensure their satisfaction no matter where they are in life.

2 Dataset

We are provided with 1.5 years of customers behavior data from Santander bank for approximately 1 million customers to predict what new products customers will purchase. There are approximately 13.5 million records entries which gives us the information about the customer and the products purchased he/she purchased.The data starts at 2015-01-28 and has monthly records of products a customer has, such as "credit card", "savings account", etc. We will predict what additional products a customer will get in the last month, 2016-06-28, in addition to what they already have at 2016-05-28. These products are the columns named: ind_(xyz)_ult1, which are the columns #25 - #48 in the training data. We have to predict what a customer will buy in addition to what they already had at 2016-05-28.

Out of all, 24 features gives us the information about the customer profile, along with this there are 24 flags which gives the information about the various products offered by the bank. If a customer owns a particular product, it will be marked as 1 otherwise it will be marked as 0.
The data description is:

Column Name	Description
fecha_dato	The table is partitioned for this column
ncodpers	Customer code
ind_empleado	Employee index: A active, B ex employed, F filial, N not employee, P pasive
pais_residencia	Customer's Country residence
sexo	Customer's sex
age	Age
fecha_alta	The date in which the customer became as the first holder of a contract in the bank
ind_nuevo	New customer Index. 1 if the customer registered in the last 6 months.
antiguedad	Customer seniority (in months)
indrel	1 (First/Primary), 99 (Primary customer during the month but not at the end of the month)
ult_fec_cli_1t	Last date as primary customer (if he isn't at the end of the month)
indrel_1mes	Customer type at the beginning of the month ,1 (First/Primary customer), 2 (co-owner),P (Potential),3 (former primary), 4(former co-owner)
tiprel_1mes	Customer relation type at the beginning of the month, A (active), I (inactive), P (former customer),R (Potential)
indresi	Residence index (S (Yes) or N (No) if the residence country is the same than the bank country)
indext	Foreigner index (S (Yes) or N (No) if the customer's birth country is different than the bank country)
conyuemp	Spouse index. 1 if the customer is spouse of an employee
canal_entrada	channel used by the customer to join
indfall	Deceased index. N/S
tipodom	Addres type. 1, primary address
cod_prov	Province code (customer's address)
nomprov	Province name
ind_actividad_cliente	Activity index (1, active customer; 0, inactive customer)
renta	Gross income of the household
segmento	segmentation: 01 - VIP, 02 - Individuals 03 - college graduated
ind_ahor_fin_ult1	Saving Account
ind_aval_fin_ult1	Guarantees
ind_cco_fin_ult1	Current Accounts

ind_cder_fin_ult1	Derivada Account
ind_cno_fin_ult1	Payroll Account
ind_ctju_fin_ult1	Junior Account
ind_ctma_fin_ult1	Ms particular Account
ind_ctop_fin_ult1	particular Account
ind_ctpp_fin_ult1	particular Plus Account
ind_deco_fin_ult1	Short-term deposits
ind_deme_fin_ult1	Medium-term deposits
ind_dela_fin_ult1	Long-term deposits
ind_ecue_fin_ult1	e-account
ind_fond_fin_ult1	Funds
ind_hip_fin_ult1	Mortgage
ind_plan_fin_ult1	Pensions
ind_pres_fin_ult1	Loans
ind_reca_fin_ult1	Taxes
ind_tjcr_fin_ult1	Credit Card
ind_valo_fin_ult1	Securities
ind_viv_fin_ult1	Home Account
ind_nomina_ult1	Payroll
ind_nom_pens_ult1	Pensions
ind_recibo_ult1	Direct Debit

3 Evaluation Metric

The evaluation metric used is Mean Average Precision@7 (MAP@7):

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

where, $|U|$ is the number of rows (users in two time points),
P(k) is the precision at cutoff k,
n is the number of predicted products,
m is the number of added products for the given user at that time point.
Note: If m = 0, the precision is defined to be 0.

Example:
Lets say, we recommended 7 products and 1st, 4th, 5th, 6th product was correct. so the result would look like 1, 0, 0, 1, 1, 1, 0.
In this case, The precision at 1 will be: $1/1 = 1$
The precision at 2 will be: 0
The precision at 3 will be: 0
The precision at 4 will be: $2/4 = 0.5$
The precision at 5 will be: $3/5 = 0.6$
The precision at 6 will be: $4/6 = 0.66$
The precision at 7 will be: 0

Average Precision will be: $1 + 0 + 0 + 0.5 + 0.6 + 0.66 + 0/4 = 0.69$
Here we always sum over the correct products, hence we are dividing by 4 and not 7. Mean average precision is an extension of average precision where we take average of all APs to get the MAP.
Formula for calculating AP@k is: sum k=1:x of precision at k * change in recall at k
Example: Its AP@2, So in fact only two first predictions matter.
Our prediction is product6 and product15.
First recommendation is wrong, so precision@1 is 0.
Second is correct, so precision@2 is 0.5.
Change in recall is 0 to 0.5, so AP@2 = 0 * 0 + 0.5 * 0.5 = 0.25

4 Data Preprocessing and Analysis

The data was very huge to be completely loaded and visualized on out computers. Using the free credits for virtual machine, we opened and analysed the data to obtain that a lot of columns had missing values. Around 27734 rows of mostly missing data were present which could be new customers.

The missing values were of columns which had categorical value (like gender, segmentation, province name, province code, customer active, new customer, customer index, etc) and numerical values (like age, seniority, household income, join data, number of products owned, married, etc).

For the categorical values, we added a new category of unknown data. For the numerical values, we did an analysis to visualize and understand the data before filling missing values.

We did various analysis, some significant of them were:

- Distribution of age - We realised that it was a bimodal distribution. There were lot of customers in young age group. But, these do not usually buy banking products so we focus on the other age group.

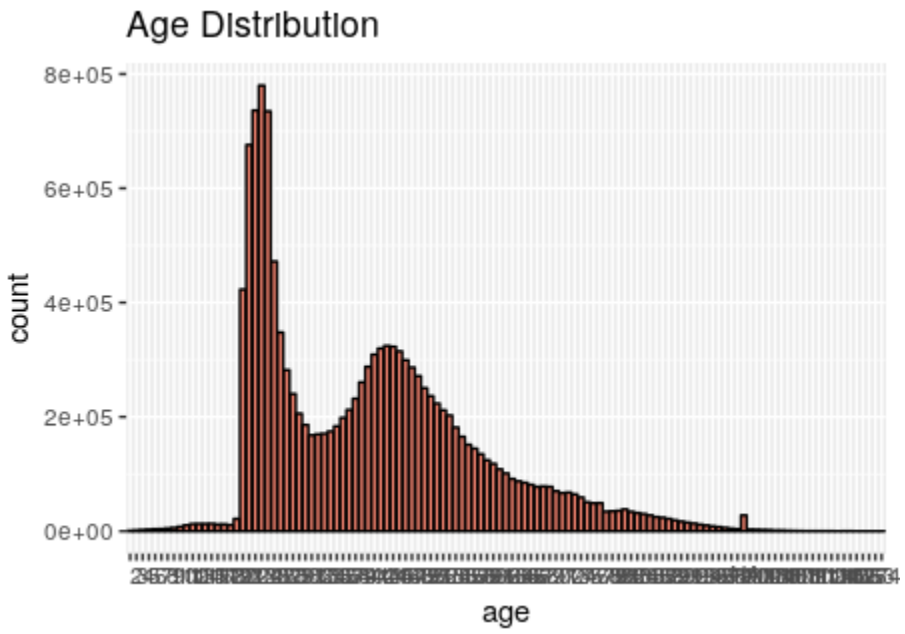

Figure 1: Age distribution

- Household income distribution - The household income seems an important feature which can determine which customers will buy products from bank. But, observing the income, we do not see very significant results.

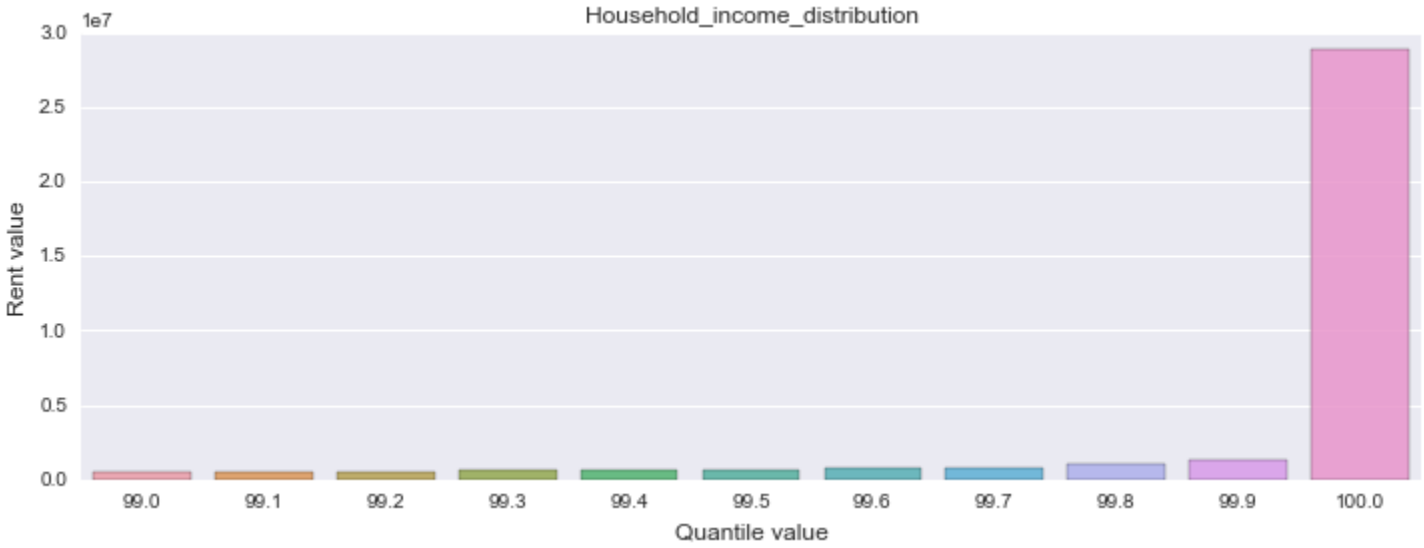

Figure 2: Income distribution

- Province v/s income - The province wise income is a more distinct chart which provides more information and helps in filling the missing values.

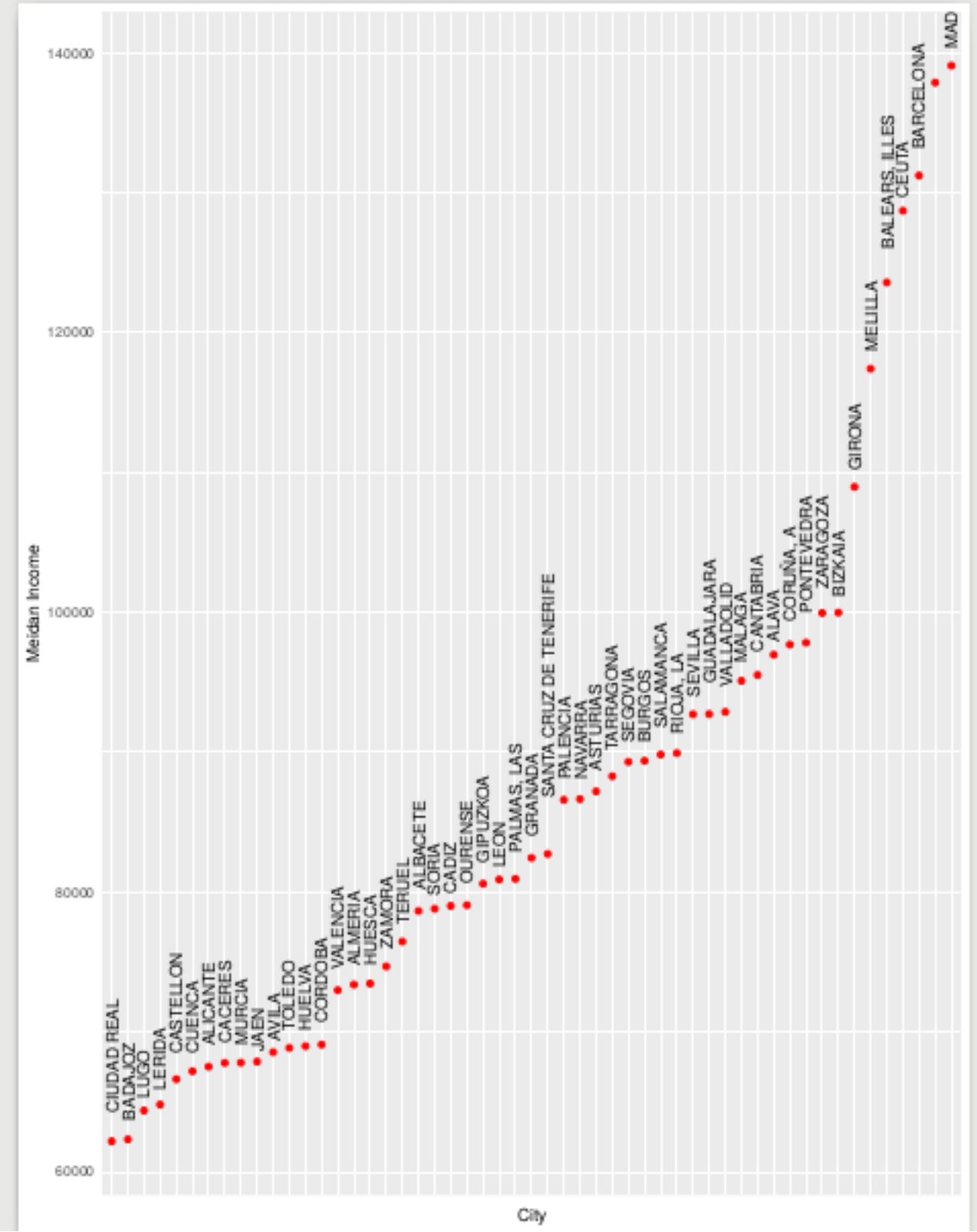

Figure 3: Province wise income

- Number of customers per month - On plotting the number of customers per month, we observe that the number of customers are steady till June 2015 and then they shoot up.

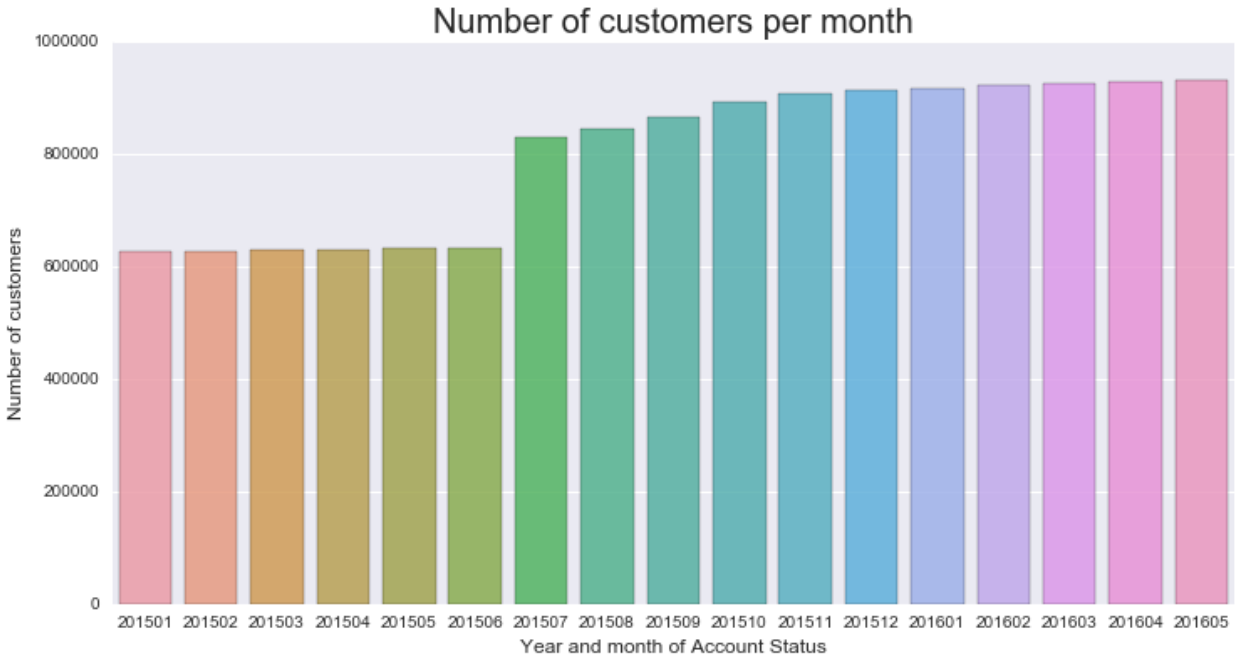

Figure 4: Number of customers per month

- Number of customers joining - On plotting the joining date wise customer pattern, we observe that there seems a cyclic pattern to the joining months. So, this can give us insight into the fact that for predicting data for month of June, we might just want to consider data for months January to June.

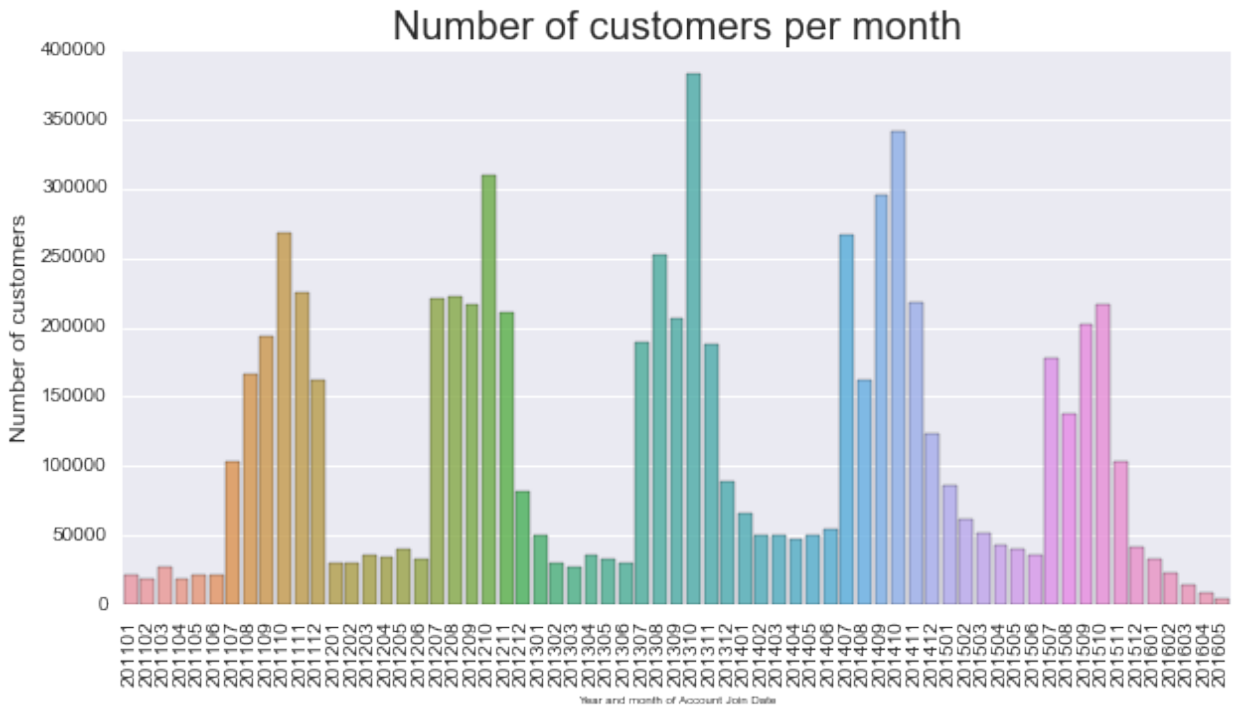

Figure 5: Number of customers joining per month

These were the few observations after the data analysis:

- The Customer data showed that buying is almost constant till Jun, and then there's sudden rise in the purchasing of the products from Jul-Dec. Maybe the bank offers some additional offers during this time.
- The buying pattern of a customer varies from month to month; there is some sort of seasonality in customer buying the products.
- As we have to predict customer; products for Jun 2016. We decided to take data only from the month of Jun 2015 and created lag features from Jan 2015 to May-2015.
- In Mar, April 2016, only 20 out of 24 products were purchased. That means, these 4 products have not been purchased at all. These products might have been stopped by Santander. So there is no point in recommending these 4 products to customers.

5 Feature Engineering

After the above analysis the dataset reduced to 27K rows in training data. But most of the features had missing values. It was found after analysis that test & train set had the same customers. So, missing values in train data can be imputed from test data; imputed almost 1K records from test datas actual values to train data. For the rest of the data, missing

values were imputed. Then, the features for the customers were used along with lag features. We focussed on only using the data of June 2015, and used the data of January 2015 to May 2015 as lag features.

Numerical Data

- Customer Age: The distribution was highly left skewed & bimodal. There are two segements which explains the bimodal distribution:
  1. University students
  2. The rest of the population.

  There were groups of customers having age <18. It was found that these were the customers having Junior accounts. Later, we decided to impute the missing values with, mean = 40.
- Seniority in months: Normalised customer seniority.
- Household income: The median income for each city was computed. Imputing missing values by the median of household income city wise and then normalised the household income.
- Join date: This seemed to be an important feature. Extracted the month of join date and report date.
- Married flag: Age > 27 then 1 otherwise 0.
- Total Number of products owned at each month.
- Calculated lag features for all the past 4 months up-to Jan 2015, to find which products were bought in the past months

Categorical Data

Missing values for all categories have been created as a separate category. The Categorical features are:

- Gender
- New Customer
- Customer employee status
- Segmentation
- Province Name
- Foreigner Index
- Residence Index
- Primary Customer at beginning of the month?
- Customer Relation type at the beginning of the month

- Customer Active
- Channel Entered
- Address block

Lag Features
The data of January 2015 to May 2015 were used as lag features. We used the vector of products owned by customer in each month (i.e. 22 products vector for each month). This 22 products vector for each of the five months was appended to the feature for the customers features of June 2015 as obtained above.

Similar approach was used for test data which used lag features of January 2016 to May 2016.
The output labels which we want are the 22 product vector for month of June 2015. For the month of June 2016, we output the probability of each product being bought by the customer and the top 7 products from this are predicted as output.

6 Theory

Recommender systems are tools for filtering and sorting items and information. They use opinions of a community of users to help individuals in that community to more effectively identify content of interest from a potentially overwhelming set of choices. There is a huge diversity of algorithms and approaches that help creating personalized recommendations.

Two of them became very popular: collaborative filtering and content-based filtering. They are used as a base of most modern recommender systems.

6.1 Traditional Recommender Systems

6.1.1 Content Based Filtering

Content-based recommender systems work with profiles of users that are created at the beginning. A profile has information about a user and his taste. In the recommendation process, the engine compares the items that were already positively rated by the user with the items he didnt rate and looks for similarities. Those items that are mostly similar to the positively rated ones, will be recommended to the user.

This approach is not possible to meet the needs of our problem statement since it is difficult to define the products of banking (e.g. Credit account, Savings account, Payroll account etc.) with its content.

6.1.2 Collabarative Filtering

The idea of collaborative filtering is in finding users in a community that share appreciations. If two users have same or almost same rated items in common, then they have similar tastes. Such users build a group or a so called neighborhood. A user gets recommendations to those

items that he/she hasnt rated before, but that were already positively rated by users in his/her neighborhood.

These approaches are most followed by various organizations to recommend products for their existing customers while they do have a cold start problem with new customers. Decent number of kagglers tried collaborative filtering based approaches by clustering users based on their preferences and recommending the products most preferred in the respective cluster. However, this type of approach doesnt consider the situation or context in which the user preferred that particular product.

6.2 Other Approaches

6.2.1 Multiple Single Class Classification

For each product a separate classification model, taking same feature for every model; is used to predict the probability if the consumer is going to own the product.

6.2.2 Market Basket Analysis

It identifies the best possible combinatory of the products or services which are frequently bought by the customers. Association analysis mostly done based on an algorithm named Apriori Algorithm. The Outcome of this analysis is called association rules. Marketers use these rules to strategize their recommendations.

6.3 Our Methodologies & Implementation

6.3.1 Multi Class Classification

A single model is used to predict the probabilities for each class all at once. For customers that added multiple products, a single one was chosen at random as the target.

6.3.2 Grid Search

When there are three or fewer hyperparameters, the common practice is to perform grid search. For each hyperparameter, the user selects a small finite set of values to explore. The grid search algorithm then trains a model for every joint specification of hyperparameter values in the Cartesian product of the set of values for each individual hyperparameter. The experiment that yields the best validation set error is then chosen as having found the best hyperparameters.

Typically, a grid search involves picking values approximately on a logarithmic scale, e.g., a learning rate taken within the set .1, .01, 103 , 104 , 105, or a number of hidden units taken with the set 50, 100, 200, 500, 1000, 2000. Grid search usually performs best when it is performed repeatedly. For example, suppose that we ran a grid search over a hyperparameter using values of 1, 0, 1. If the best value found is 1, then we underestimated the range in which the best lies and we should shift the grid and run another search with in, for example, 1, 2, 3. If we find that the best value of is 0, then we may wish to refine our estimate by zooming in and running a grid search over .1, 0, .1.

6.3.3 Xtreme Gradient Boosting (XGBoost)

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

The xgboost was used with softprob as objective function so as to get the probabilities of output class rather than the product purchased. It provides a more regularized model formalization to control over-fitting, which gives it better performance. We used the parameters which were obtained by performance tuning from Grid Search.

7 Evaluation and Results

The results were stored in csv format which had top 7 products predicted which will be bought by customer.

added_products	ncodpers
ind_recibo_ult1 ind_reca_fin_ult1 ind_ecue_fin_ult1 ind_ctop_fin_ult1 ind_nomina_ult1 ind_nom_pens_ult1 ind_fond_fin_ult1	15889
ind_recibo_ult1 ind_reca_fin_ult1 ind_nomina_ult1 ind_nom_pens_ult1 ind_cno_fin_ult1 ind_tjcr_fin_ult1 ind_ecue_fin_ult1	1170544
ind_recibo_ult1 ind_nom_pens_ult1 ind_nomina_ult1 ind_cno_fin_ult1 ind_tjcr_fin_ult1 ind_reca_fin_ult1 ind_ecue_fin_ult1	1170545
ind_recibo_ult1 ind_nomina_ult1 ind_nom_pens_ult1 ind_reca_fin_ult1 ind_cno_fin_ult1 ind_tjcr_fin_ult1 ind_ecue_fin_ult1	1170547
ind_recibo_ult1 ind_reca_fin_ult1 ind_nomina_ult1 ind_nom_pens_ult1 ind_cno_fin_ult1 ind_tjcr_fin_ult1 ind_ecue_fin_ult1	1170548
ind_recibo_ult1 ind_nomina_ult1 ind_reca_fin_ult1 ind_nom_pens_ult1 ind_cno_fin_ult1 ind_tjcr_fin_ult1 ind_ecue_fin_ult1	1170550
ind_nom_pens_ult1 ind_reca_fin_ult1 ind_nomina_ult1 ind_cno_fin_ult1 ind_tjcr_fin_ult1 ind_ecue_fin_ult1 ind_fond_fin_ult1	1170552
ind_recibo_ult1 ind_reca_fin_ult1 ind_nomina_ult1 ind_cno_fin_ult1 ind_nom_pens_ult1 ind_tjcr_fin_ult1 ind_ecue_fin_ult1	1170553
ind_recibo_ult1 ind_nomina_ult1 ind_nom_pens_ult1 ind_reca_fin_ult1 ind_cno_fin_ult1 ind_tjcr_fin_ult1 ind_ecue_fin_ult1	1170555
ind_recibo_ult1 ind_nom_pens_ult1 ind_nomina_ult1 ind_cno_fin_ult1 ind_tjcr_fin_ult1 ind_reca_fin_ult1 ind_ecue_fin_ult1	1170557
ind_cco_fin_ult1 ind_recibo_ult1 ind_cno_fin_ult1 ind_nom_pens_ult1 ind_nomina_ult1 ind_reca_fin_ult1 ind_tjcr_fin_ult1	1170559
ind_recibo_ult1 ind_nomina_ult1 ind_nom_pens_ult1 ind_reca_fin_ult1 ind_cno_fin_ult1 ind_tjcr_fin_ult1 ind_ecue_fin_ult1	1170563

Figure 6: Results

The results obtained were submitted on Kaggle and a MAP@7 score of 0.0301859 was obtained which is top 5% on Kaggle.

8 Future Work and Conclusions

The data was huge and we tried to make it work on our system by considering only January to June data, after observing the trends which showed that it was a feasible option. The recommendations can be made by making use of entire 1.5 years of data which might improve the results.

The filling of missing values was also done in a naive way. A sophisticated model based approach for imputing missing values could be used.

Experimenting with various other alternative ML algorithms can also be done. Also frameworks of recommender systems like GraphLab and LightFM can also be tried for various recommendation approaches.

The project helped us realize the importance of data analysis which gave us the idea of reducing the data and make it work on normal computers. It also helped us understand various ways of dealing with recommendation and how the lag features and feature engineering helps achieve the required task. The task was completed with fair enough results, though there is always scope for imporvement.

References

[1] *Santander Product Recommendation*, available at `https://www.kaggle.com/c/santander-product-r`

[2] *Intuition behind Average Precision and MAP*, available at `https://makarandtapaswi.wordpress.com/2012/07/02/intuition-behind-average-precision-`

[3] Ian Goodfellow and Yoshua Bengio and Aaron Courville, *Deep Learning*,(MIT Press, 2016).

[4] Tianqi Chen and Carlos Guestrin, *XGBoost: A Scalable Tree Boosting System*,(CoRR, 2016).

[5] A. C. Melissinos and J. Napolitano, *Experiments in Modern Physics*, (Academic Press, New York, 2003).

[6] N. Cyr, M. Têtu, and M. Breton, IEEE Trans. Instrum. Meas. **42**, 640 (1993).

[7] *Expected value*, available at `http://en.wikipedia.org/wiki/Expected_value`.