Group 6
CHEN JIA
CHEN YI
GAO YUNYI
SUN YUROU
XIAO LIANG

Santander Hybrid Recommendation System

Github Link: https://github.com/xlsi/Santander-Hybrid-Recommendation-System

Abstract

Santander Bank is a wholly owned subsidiary of the Spanish Santander Group based in Boston and its principal market is the northeastern United States. With a huge customer base, the current recommendation system of Santander provides a disproportionate customer experience as only a few customers receive numerous recommendations while many others hardly receive any.

Thus in this project, we explored different machine learning based recommendation models using user personal information and their product purchase history. These models include: 1. baseline model (recommend the most popular products); 2. Tree based model (Decision Trees, Random Forest); 3. Autoencoder and its variants for collaborative filtering. We also explored combining these models into an ensemble model. After evaluating with the hit ratio, collaborative filtering is identified as the best-performing model.

1. Introduction

Santander Bank is a wholly owned subsidiary of the Spanish Santander Group based in Boston and its principal market is the northeastern United States. With a huge customer base, the current recommendation system of Santander provides a disproportionate customer experience as only a few customers receive numerous recommendations while many others hardly receive any.

This leads to two consequences: From the bank side, it will lose potential opportunities to sell products to its customers causing potential revenue loss. From the customer's side, those that hardly receive any recommendations will have bad user experience, and have little idea about the various products available in the bank.

By implementing a more efficient recommendation system, the team intends to cater to the needs of each customer, and enhance their overall user experience, which will in turn increase the revenue of the bank.

2. Dataset & Exploratory Data Analysis

The team is provided with 2 datasets - the training dataset and test dataset. In the training dataset, input variables include the basic information of each customer including their age, employment status, income level and the products they owned in each month between Jan 2015 and May 2016. In the test dataset, the same input variables are available except the product variables which are binary prediction variables that indicate additional products that a customer will get in the next month, in addition to what they already have in May 2016.

Table 1: Input Variable Description Table

Column Name	**Description**
fecha_dato	Month. 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

Table 2: Prediction Variable Description Table

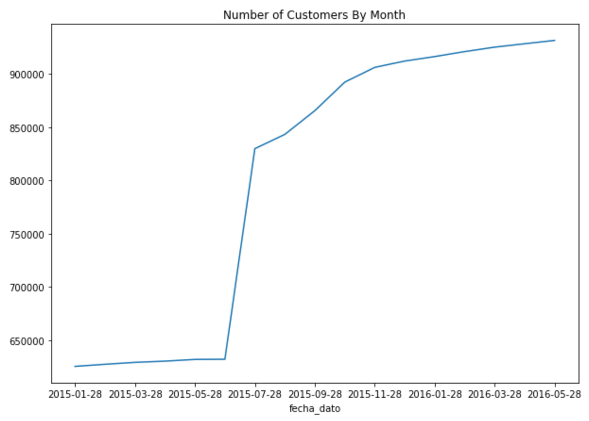
Column Name	**Description**
ind_ahor_fin_ult1	Saving Account
ind_aval_fin_ult1	Guarantees
ind_cco_fin_ult1	Current Accounts
ind_cder_fin_ult1	Derivada Account
ind_cno_fin_ult1	Payroll Account
ind_ctju_fin_ult1	Junior Account
ind_ctma_fin_ult1	Más particular Account
ind_ctop_fin_ult1	particular Account
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

2.1 Exploratory Data Analysis

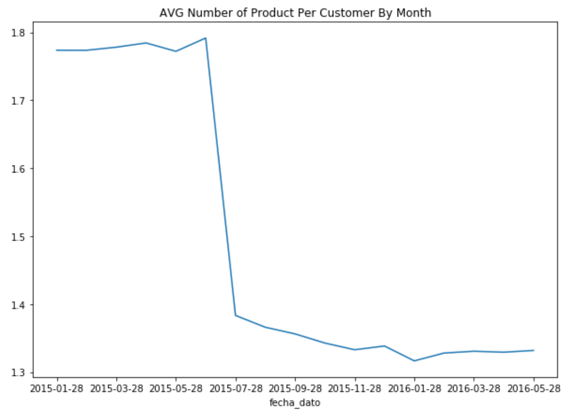
Since the training dataset contains Santander Bank's customers' monthly product usage records, time series plots can help better understand what's happening over time.

From the monthly number of customers plot, it is observed that Santander Bank's customer base has been steadily increasing month on month over the period and achieved a great jump in June 2015. However, at the same time, the average number of products purchased per customer has been dropping significantly after June 2015, which further proves that a good recommendation system needs to be implemented for Santander Bank to provide better experience to their customers.

Plot 1: Number of Customers by Month

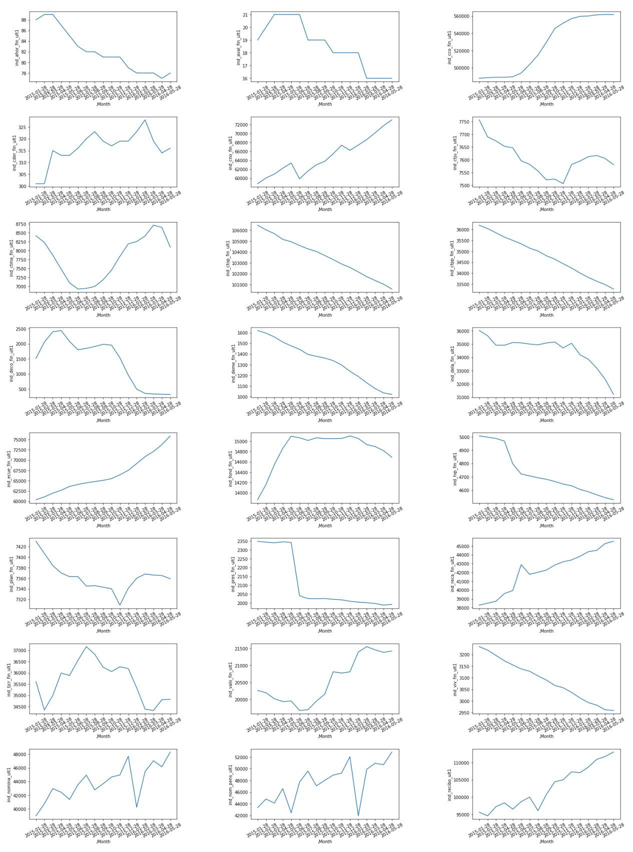


Plot 2: AVG Number of Products Per Customer by Month



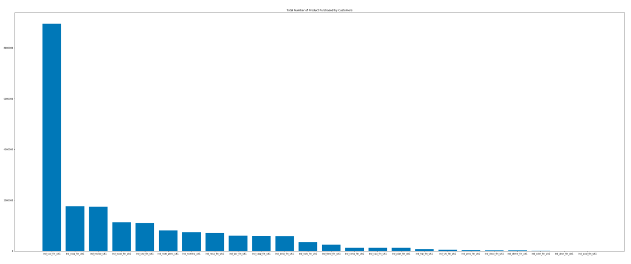
Again, when time series is plotted by each product, only some products showed a similar increasing trend as the customer base while the other products became less popular even when the number of customers increased. An example of the products that have been more popular with a larger customer base is ind_cno_fin_ult1, while is the Payroll Account. An example of the products that have been less popular even with a larger customer base is ind_ctop_fin_ult1, which is the Particular Account.

Plot 3: Number of Customers by Product by Month

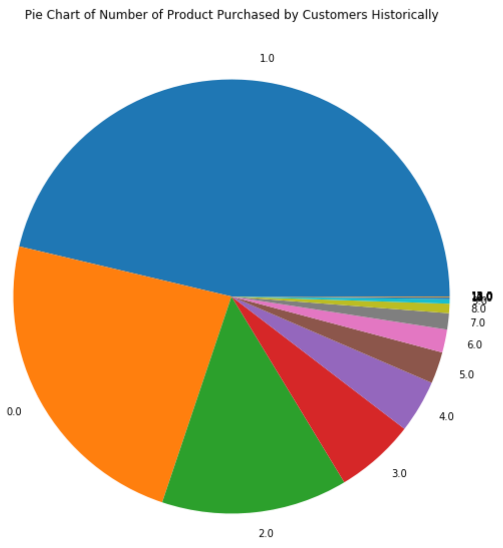


Another important finding is that almost all customers hold a current account and almost half of the customers only purchased one product throughout the period. This will make recommendation even harder as we have limited ground truth about the products that customers actually purchased.

Plot 4: Total Number of Products Purchased by Customers



Plot 5: Distribution of Customer Products Number



3. Data Pre-processing

The data-preprocessing mainly performed in this project is null-value handling.

For baseline models and tree-based models, missing values are filled in with 0. Input data was grouped into feature variables (e.g. user age, user income level) and target variables (products that users purchased).

For the collaborative filtering model user-item matrix, rows where all values are 0, which means that the customer does not own any products are all, were removed from the dataset. Otherwise, null values are filled in with 0.

4. Modeling

4.1 Model Evaluation

The success metric to measure the effectiveness of the recommendation system in place is the recommendation hit ratio - the number of recommended products purchased by customers divided by the total number of additional products that they purchased in the following month. The recommendation system should serve to recommend products that are highly likely to be purchased by customers. The more recommended products bought by customers, the better the recommendation system.

4.2 Baseline Model Concept

The bank wants all the products that the customer is most likely to buy in the coming month based on the customers' meta data and historical purchases, so that they can recommend the top products to boost sales and provide better services.

To map this into a business problem, we have to predict if a customer would purchase a certain product in a month or not. This problem can be defined as a special multi-output classification problem where we calculate the probability that a customer would purchase a product as the output. According to the dataset specifications, at most 7 products are recommended to customers and the rest are ignored. In addition, we assume that customers cannot buy products he already owns, which means he can only add new products have not bought before in the next month.

To illustrate the basic concept, as we will be predicting probabilities of various products and customers can opt for more than one product at a time, we would need multi label log loss. The baseline model first collects products owned by all customers and deleted rows where no new purchase is made. Then the model randomly assigns products to customers denoted as "added products". If a product is owned by the customer already, then its probability will be set to 0 and append the next product. To make more sense, the next step is to recommend the most popular products that customers have not owned yet.

4.3 Tree Based Model

After introducing our basic concepts in mapping business problems, now it is time to implement them in models. In order to make better predictions with metrics to evaluate the final recommendations, we employed two tree-based models to predict products that may interest customers and recommended seven new products to them.

4.3.1 Decision Tree

We first used a simple decision tree classifier to fit our training data set split from the original train file. Given the size of the data, trees are chosen because they are fast to train and evaluate. Then we make predictions on the validation set to get the top ranked products with highest probability that customers may choose to purchase in next month. After removing the products that are already owned by customers, final "add_products" columns represent the model's recommendations for each customer for the next month's recommendation.

The overall performance of decision tree model is:

Table 3: Decision Tree Model Performance

Accuracy	Precision	F1-score	Hit-ratio
0.38	0.39	0.39	0.39

As we can see from the scores generated, the decision tree model did not perform well on recommending products even with the train file. These low scores could be due to noise and outliers in the data set that the decision tree model is sensitive to. In addition, based on our dataset, customer purchasing patterns are hard to predict by a single decision tree without a measure of feature importance.

To further improve the scores, decision trees can be combined with other trees to create more powerful models, such as Random Forests. Random Forest can reduce the variance by aggregating the results of multiple trees, moreover, Random Forest is more robust to outliers and noisy data than decision trees.

4.3.2 Random Forest

To make this prediction, we used a random forest classifier to determine the probability of a customer purchasing each product. We then filtered the top products with the highest probability, removed those already owned by the customers, and recommended the remaining products to the customers.

The sample result is shown in the following table. In this table, “added_products” denotes the top 7 products we recommend , and “actually_buy” represents the products that a customer actually buys. For instance, for the customer with ID 1229085, we recommend products such as Guarantees, Pensions, Derivada Account, Payroll, Home Account, Long-term deposits, and Loan. It is worth noting that this customer eventually purchased one product, namely Guarantee, which was among our recommended products. However, for customer 1229084, who bought the products Payroll Account, Payroll, and Pensions, our tree based recommendation model did not hit any of these products.

Table 4: Recommendation results with Random Forest

ncodpers	added_products	actually_buy
1229085	Guarantees, Pensions, Derivada Account, Payroll, Home Account, Long-term deposits, Loan	Guarantees
1229084	Guarantees, Derivada Account, Home Account, Derivada Account, Loans, Taxes, Medium-term deposits	Payroll Account, Payroll, Pensions

The overall performance of the random forest model is as follows:

Table 5: Random Forest Model Performance

Accuracy	Precision	F1-score	Hit-ratio
0.44	0.63	0.52	0.44

Based on the result report, it can be observed that the tree-based model did not perform ideally in this recommendation project, with a lower accuracy and hit-ratio score. This is primarily due to the fact that customers' purchasing habits vary greatly from one another, making it difficult to identify similar patterns. Moreover, the dataset is quite large and contains many undefined and unstructured values, which create a lot of noise in our predictions.

4.4 Collaborative Filtering

Collaborative filtering is a recommendation technique that can be used to suggest relevant banking services to users based on their profile or banking activity. It works by analyzing the patterns and preferences of similar users, and then making recommendations based on the actions of those users. For example, if a group of users with similar banking activities and profiles have shown interest in a specific banking service, then it is likely that other users in this group would also be interested in the same service.

In this project, we used auto-encoder and its variants to implement collaborative filtering to capture user preferences and make recommendations.

An auto-encoder is a category of neural network that is appropriate for unsupervised learning assignments, such as creating models, reducing dimensionality, and efficient encoding. It has demonstrated its exceptional ability to learn the fundamental feature representations in various fields, such as computer vision, speech recognition, and language modeling. As a result, new recommendation systems have included auto-encoder, creating more chances to enhance user experiences and meet customer expectations.

The AE model architecture is implemented as below:

```
Auto-Encoder (AE)
|-- Encoder
|   |-- Linear (nb_products -> 512)
|   |-- Sigmoid
|   |-- Dropout (0.9)
|   |-- Linear (512 -> 80)
|   |-- Sigmoid
|-- Fully connected layers
|   |-- fc1: Linear (80 -> 32)
|   |-- fc2: Linear (80 -> 32)
|-- Reparameterization
|-- Decoder
    |-- Linear (32 -> 80)
    |-- Sigmoid
    |-- Linear (80 -> 512)
    |-- Sigmoid
    |-- Linear (512 -> nb_products)
```

In total we have 687830 records of the user-item matrix after deleting blank rows and filling nah values with 0. We first calculate how many products each user has owned to see the distribution and sparsity of source data. For data preprocessing ,the transaction data was read using pandas and grouped by the user identifier 'ncodpers'. The sum of all the products/services the user had was calculated. Below is the table showing the top 5 count of product rates.

Table 6: Top 5 Count of Product Rate

Count of product rates	Count of records
1	452337
2	117885
3	47187
4	27366
5	16824

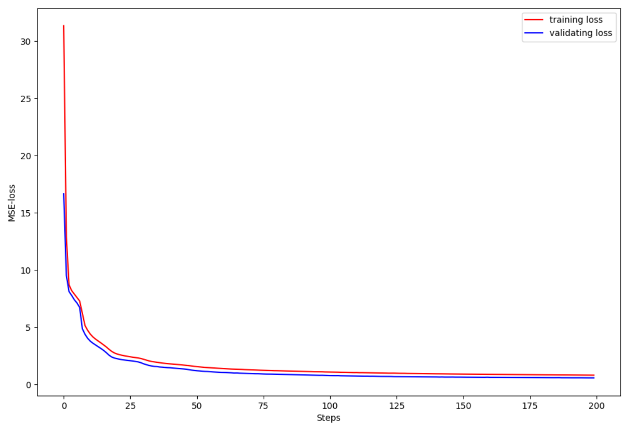
We can find from the table that most of the users own less than 3 items, which means that our data is very sparse.

The data was then split into training and testing sets. For model development, we trained an AutoEncoder to learn the low-dimensional representations of user preferences for banking products/services. The AE model consists of an encoder and a decoder. The encoder reduces the dimensionality of the input data, while the decoder reconstructs the input from the reduced dimensionality.

We implemented the AE model in PyTorch and trained on the user transaction data with a batch size of 256, using a mean squared error (MSE) loss function. The MSE loss function only considers the user-rated products while ignoring unrated products. A total of 200 epochs were used for training.

During the training process, the model's performance was evaluated on a validation set. The average training and validation loss for each epoch were printed and plotted in the graph below.

Plot 6: AVG Training and Validation Loss at Each Epoch



The training record shows a consistent decrease in both training and validation loss as the number of epochs increases, indicating that the model is learning to reconstruct the input data more accurately over time. And it can be observed that the model's performance improves substantially in the initial epochs and continues to improve, albeit at a slower rate, in later epochs.

After training, the AE model was used to predict product preferences for users in the test set. The recommendations were generated by selecting the top 2 products for each user that they had not yet interacted with, based on the predicted preferences.

We experimented with different numbers of recommended items to evaluate the models' performance, and the results are as follows:

Table 7: Model Performance with Different Number of Recommended Items

#Recommended Items	Model	Hit Ratio
5	AE	0.09
	VAE	0.03
10	AE	0.53
	VAE	0.42

The Hit Ratio is not particularly satisfactory, which we believe may be due to the sparsity of our data. Once the model is deployed, we can get more feedback to retrain the model.

5. Conclusions

5.1 Model Results Comparison

The summary of each model's performance is in the table below:

Table 8: Model Performance Summary Table

Method	Model	Hit ratio
Baseline	Decision tree	0.39
User Based	Random forest	0.44
Collaborative Filtering	Auto-Encoder	0.53
	VAE	0.42

5.2 Insights

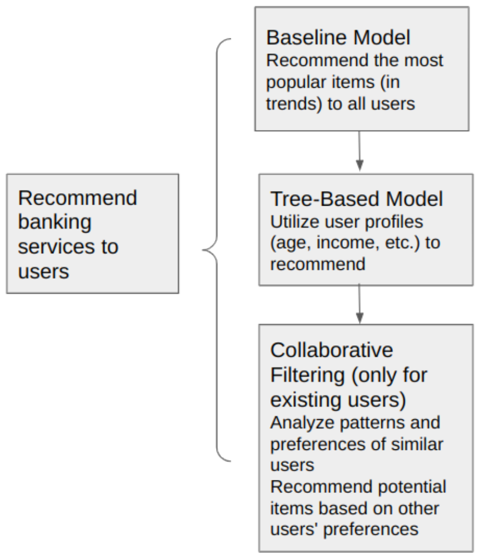
We find that the most recommended items are Guarantee, Mortgage, Saving Account, Medium Term Deposit and Loans.

We assume that our baseline model will always recommend the most popular item which is the Saving Account as almost everyone will get a saving account when opening a new bank account. Then, as the model needs to recommend products that were not purchased by the customer before, products like Guarantee, Mortgage, Medium Term Deposit and Loans which were less popular among existing customers before are now surprisingly highly recommended by the recommendation system to customers.

6. Recommendation Strategy

We come up with hybrid methods to satisfy different needs of users. For all users, we will first use a baseline model to recommend the most popular items that are in trends. Then, we will utilize tree-based models using their profiles, like age and income to recommend items that most fit their background. Finally, we will use collaborative filtering to recommend the potential items that they may like if they are existing users. Strategy graph is as below:

Plot 7: Final Recommendation System Strategy


Recommend banking services to users
Baseline Model
Recommend the most popular items (in trends) to all users
Tree-Based Model
Utilize user profiles (age, income, etc.) to recommend
Collaborative Filtering (only for existing users)
Analyze patterns and preferences of similar users
Recommend potential items based on other users' preferences

7. Limitations & Future Work

One of the most significant limitations with this project is the lack of enough factors to do causality analysis. The financial products that customers want to purchase are highly dependent on other factors as well. For example, external factors like economic situations or user profile like occupation. When a fresh graduate first starts working, he/ she would want to set up a payroll account first instead of a mortgage. But input data doesn't include such data.

Another important reflection is that some factors may be highly dependent. However high correlation does not mean causality. For example, in order to make a mortgage payment,one must deposit first then mortgage, but it doesn't mean we should recommend a mortgage if someone gets a deposit account.

Also for experiment set up, there is data sparsity issue, hence it is challenging to make accurate recommendations, as the model may not have enough information to learn from.

Finally, a cold-start problem exists for the recommendation system: new users without any historical data might not receive accurate recommendations, as the collaborative filtering method relies on the past behavior of similar users. But we have collated the different methods and proposed in section 6 a recommendation strategy to mitigate this issue.

References

Authors, A. Suppressed for anonymity, 2010.

Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*. John Wiley and Sons, 2nd edition, 2000.