Datathon Cajamar

Microsoft Predictive Modelling

• • •

DATASHOTS

#UniversityHack March 2017

About DataShots

Ana Valdivia

Mathematician (UPC)

Data Scientist in UGR

UNIVERSIDAD DE GRANADA

Hugo Casero

Computer Science (UPV)

PhD student in UGR

Master in Data Science 2015-2016 (UGR)

Contents

1. The Challenge

Could you recommend your ideal financial product basket to Cajamar customers?

3.350.601
registros para el training

1.147.687
registros para el testing

8
variables por registro

cajamar
CAJA RURAL

*The challenge consists in **analyzing** the **historic of contracted products** by customer in order to develop an engine to **recommend future products**.*

2. Visualization of Data

*The first step after the problem is understood is to **analyze the data***

Statistically

We check that train and test have the same distributions of variables

R functions

`str()` `summary()`

Visually

Customers **Products**

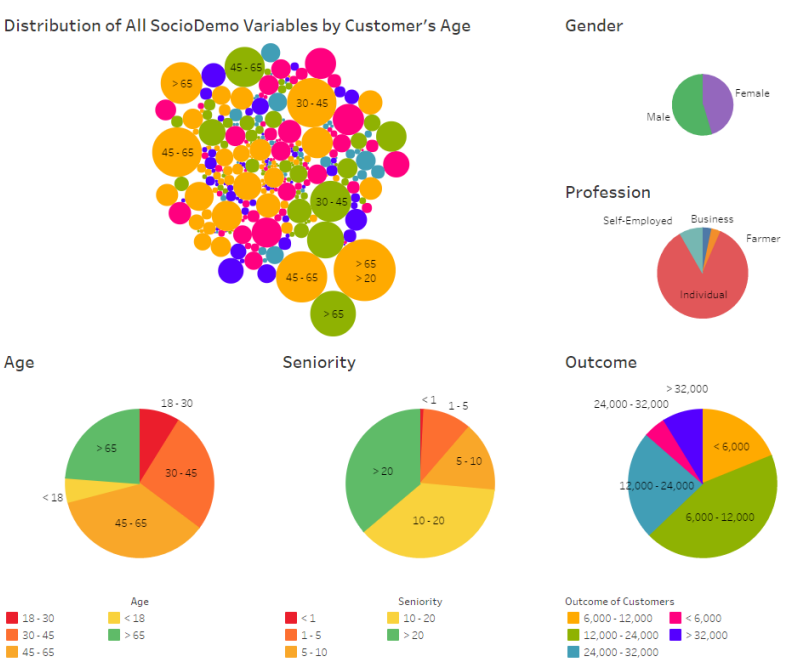
*Thanks to this we found some **inconsistencies** in the original data, resulting in Cajamar **providing new datasets** for this challenge.*

2. Visualization of Data

Customers

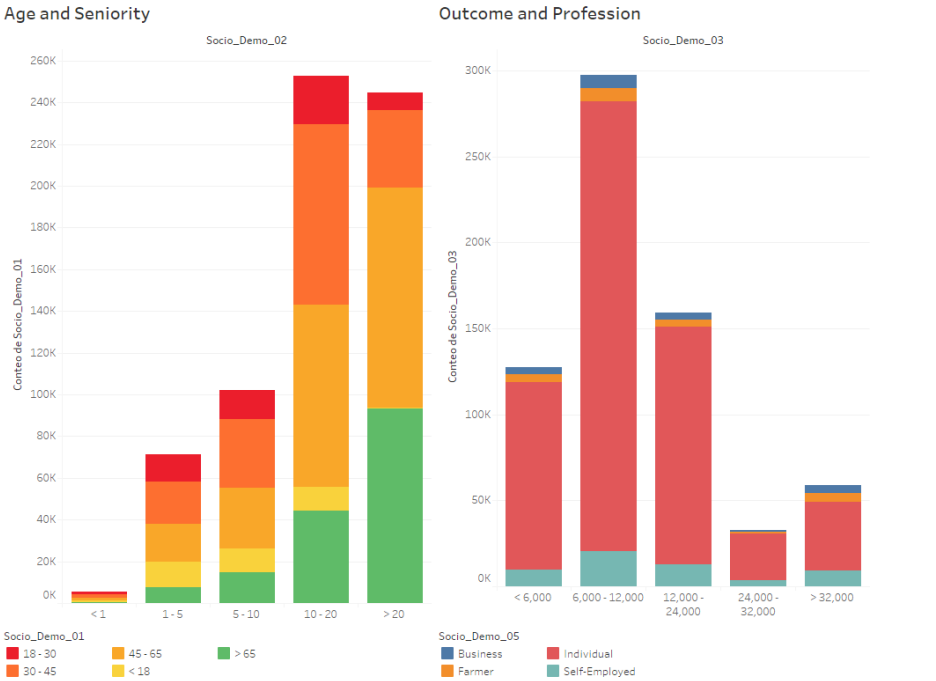
Click on the images to interact with the plot

#UniversityHack #CustomersI



Visualization of the 1000 (5 x 5 x 5 x 2 x 4) families of customers by its sociodemographic variables.

#UniversityHack #CustomersII



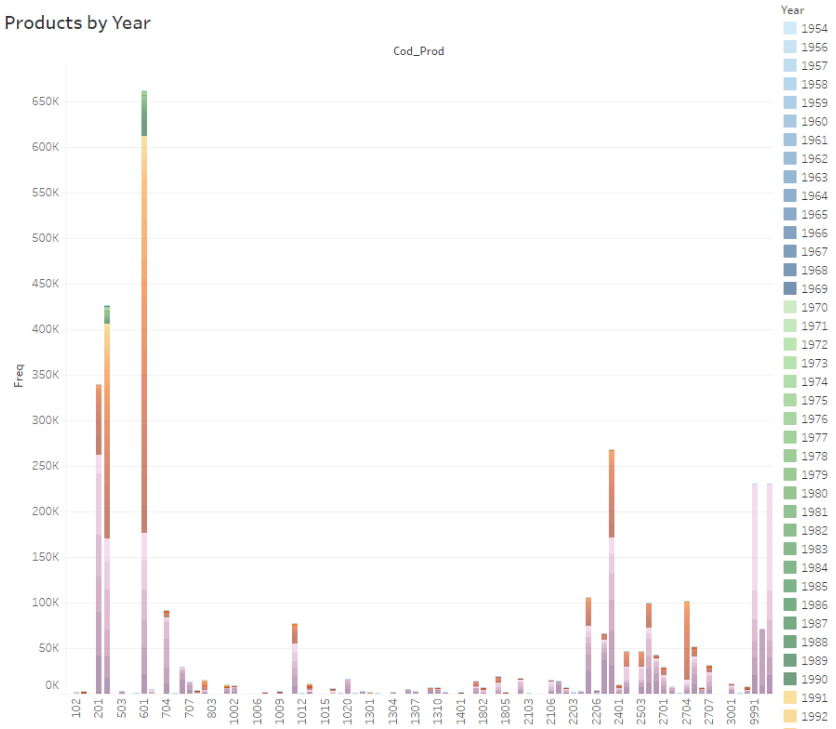
Age versus Seniority (left) and Outcome versus Profession (right).

2. Visualization of Data

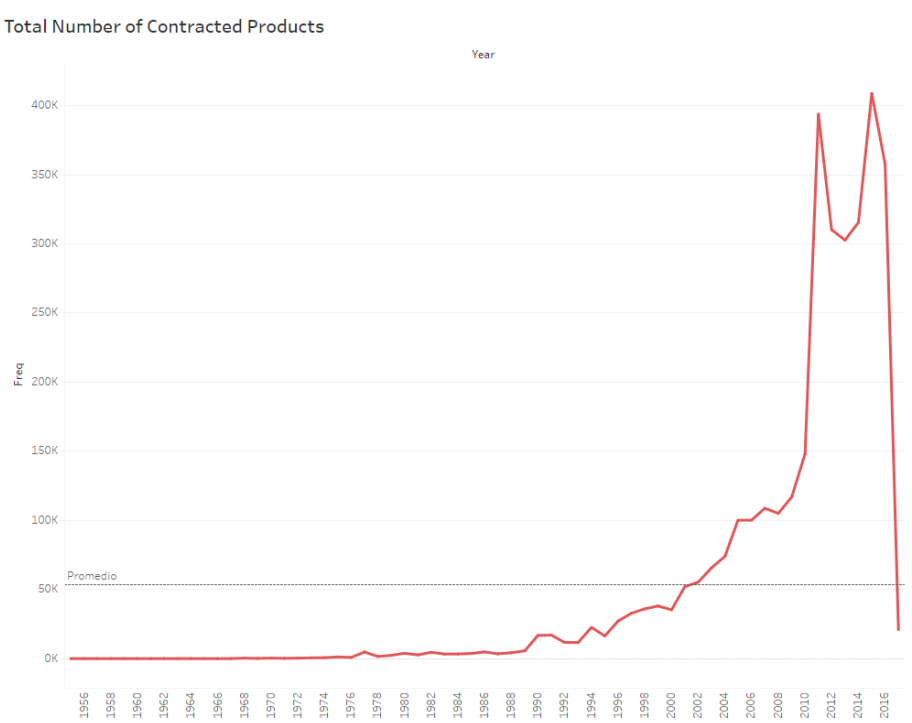
Products

Click on the images to interact with the plot

#UniversityHack #ProductsI



#UniversityHack #ProductsII



Contracted products by year. This plot helps to study if a product is old, new or regular over time.

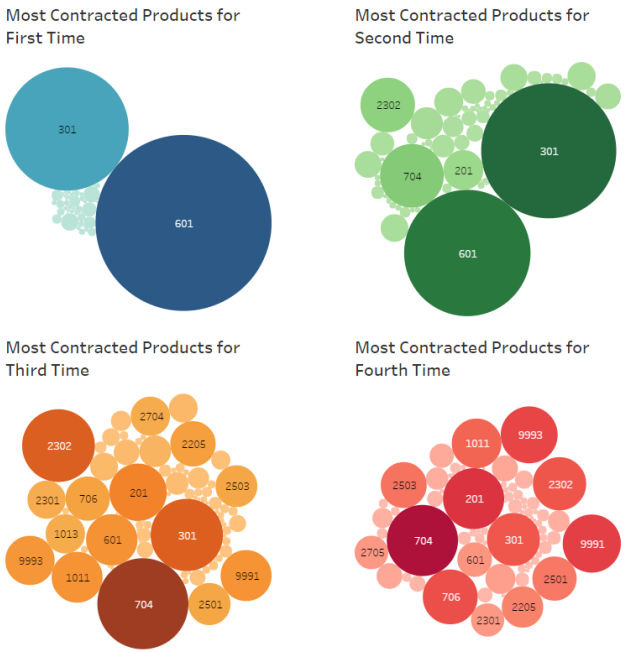
Number of total contracted products by year.

2. Visualization of Data

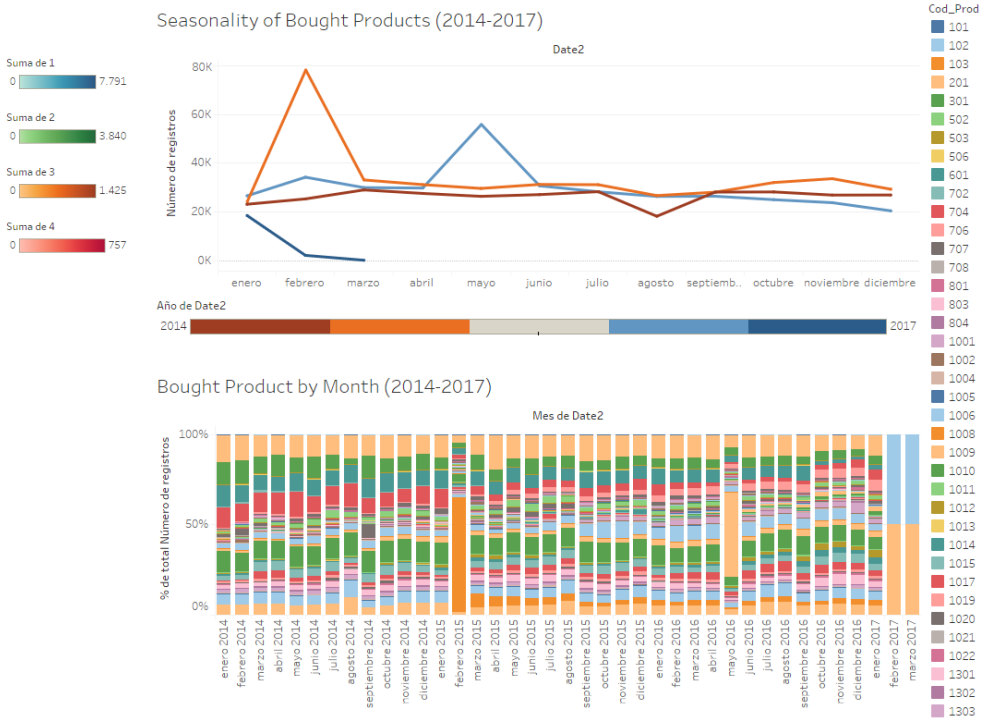
Products

Click on the images to interact with the plot

#UniversityHack #ProductsIII



#UniversityHack #ProductsIV



Fourth most contracted products by order.

601 and 301 are the most contracted products by first and second time.

Seasonality of contracted products (top) and ratios of contracted products of last three years.

In these graphics we detect that on Feb 2015 the product 9992 was launched. It looks like in May 2016 the product 2301 was relaunched.

2. Visualization of Data

Customers + Products

Click on the images to interact with the plot

#UniversityHack #Customer+ProductsI

#UniversityHack #Customer+ProductsII

Distribution of sociodemographics variables by product.

Who is contracting each product?

In the plots we detect that some products are exclusive for some customers: 1312, 2104 and 2105 were only purchased by seniors, 2901 by women, 1009 and 1019 by farmers and bussines…

3. Feature Engineering

Creating new variables

Variable	Description	Why?	Models
seq	This variable counts the order (sequence) of the contracted product. It is important to measure the order of customer purchasing (see).	The plot #UniversityHack #ProductsIII shows us that it is important the order of the product.	RSM & XGBoost
Diff_March2017	Time between last product purchased and March 2017.	We think that there are differences to predict next product between customers that purchased last product five years ago and today.	XGBoost
Products	Digit representation of all purchased products by the customer in the past. Each digit is a product, 1 if it was purchased and 0 otherwise. This variable sums up the purchase history of customers.	It is important to take into account the behavior of customer purchasing.	XGBoost
Age, Senior and Income	The average number for each group, eg.: if Socio_Demo_01 == 4 *(Edad >= 45 años y Edad < 65 años)* => Age == 55.	XGBoost does not accept categorical variables.	XGBoost

4. The Recommendation System Model (RSM)

Our first approach

IDEA. Predict the next product taking into account sociodemo variables as well as old purchased products. This idea melds the **Up Selling** technique, well-known in e-commerce area.
ASSUMPTION. We all behave like those who resemble us.

RSM 1

1. Obtain the customer (C) to predict in test.
2. Take all customer in train with the same sociodemo var.
3. Compute the most purchased products of these customers.
4. Select the first most purchased product of the list, if it was not purchased before by C.

RSM 2

1. Obtain the customer (C) to predict in test.
2. Take all customer in train with the same sociodemo var and same purchased history of C.
3. Compute the most purchased products of these customers.
4. Select the first most purchased product of the list, if it was not purchased before by C.

BUILD THE MODEL	PARALLELIZATION	EVALUATE RESULTS
RSM1 and RSM2.	BIG DATA solution: split the set in 8 blocks (one for each node). This strategy reduce the computational time.	Compute and check accuracy in the validation set.

Take a look to the code in:

```
RSM1 -> RecommendationEngine_exp1_X.R
RSM2 -> RecommendationEngine_exp3_X.R
```

5. XGBoost Model

Our second approach

IDEA. Predict/Recommend the next product building a classification model for each product and ensembling final probabilities. This idea melds the **Up Selling** and **Cross Selling** techniques, well-known in e-commerce area.

ASSUMPTION. We all behave like those who resemble us. Machine Learning models are more accurate than human engines.

XGBoost

1. Select the *k* most purchased products over last three years.
2. Build a dataset with the last purchased product in train customers.
3. Set classification variable: if the last purchased product is the product model == 1, otherwise 0.
4. Train k XGBoost models.
5. Select the highest probability of the product for each customer, if it was not purchased before.

CREATE DATASET

Select the *k* most purchased products over last three years. Build datasets with the last purchased product in the train set.

BUILD MODEL

Train a XGBoost for each *i* product. The classification variable is 1 if the last purchased product is *i*, and 0 otherwise.
For each customer, recommend the product with the highest probability of the *k* products.
Train with new variables for outperforming.
Validate the model with the validation set.

EVALUATE RESULTS

Compute and check accuracy in the validation set.

Take a look to the code in:

`XGBoost_2_Data_Preprocessing_Model.R`

6. Results

XGBoost rocks!

MEASURES:

- **AUC:** For training XGBoost models.
- **% accuracy:** For evaluating precision on the predictions.

% accuracy = $100 \cdot \frac{1}{N} \sum_i^n C_i$

where:

- n number of total customers to predict.
- C_i = 1 if the last purchased product is equals to the predicted, and C_i= 0 otherwise.

EVALUATION

- We evaluate the accuracy for predicting 1 and 5 products.
- The **validation set** is the set predicting the **last purchased product** in the **test set**.

Models	% accuracy for 1 product	% accuracy for 5 products	Comments
RSM 1	28.56 %	71.28 %	Computational time (~ 2.5 days).
RSM 2	26.51 %	70.36 %	Computational time (~ 2.5 days).
XGBoost 1	42.12 %	77.26 %	Computational time (< 2hours). The AUC average of 76 products is 0.84.
XGBoost 2	**45.35 %**	**79.16 %**	Computational time (< 2hours). **XGBoost + "Products" variable. The AUC average of 76 products is 0.87.**

6. Results

Step by step

1

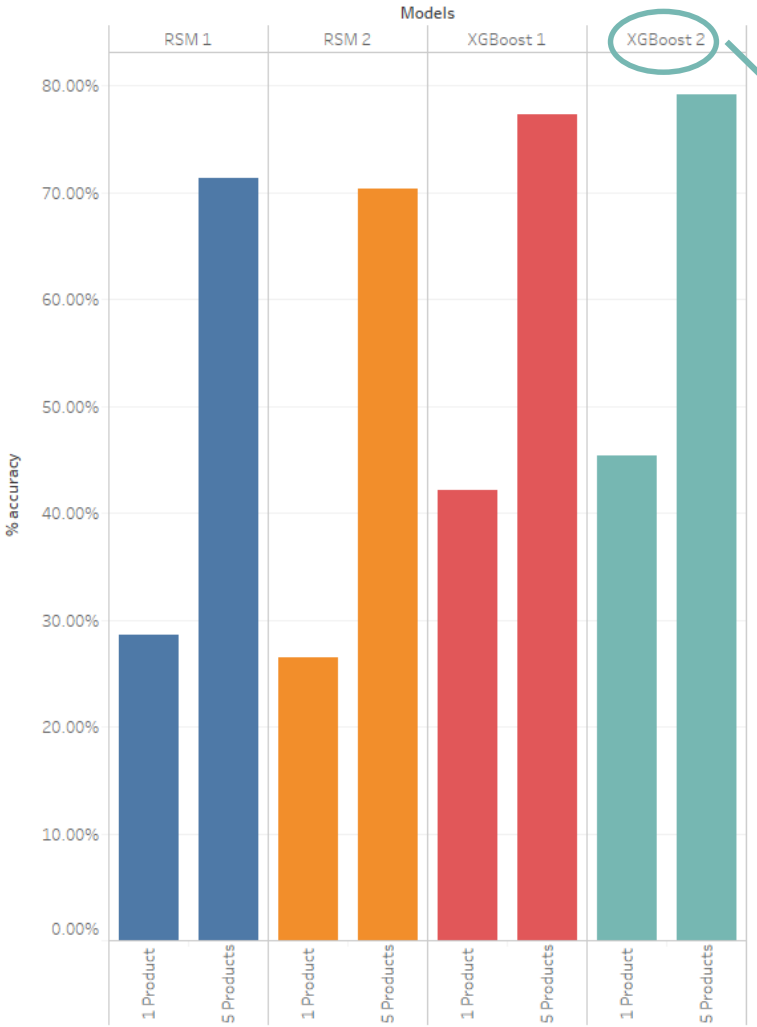
*We started trying with dummy variables and different algorithms. **Cleaning** specific **outliers** and also applying **association sequential** and **non rule-based techniques**, but results were not satisfying.*

2

*Then we moved to the first approach (**RSM**) and some interesting results showed up.*

3

*Our final approach with **XGBoost** proved to be the **most successful**, with almost **80% accuracy** for **5 products** and an **average AUC** of **0.87**.*



***XGBoost 2** is built with a **Cross Selling** strategy, adding the new variable "**Products**" to the model.*

6. Results

Final remarks

*We recommend **5 products** because we think that for the Entity is **more useful** to **chose** between a basket of **different purchase options**.*

*Our **models** have **improvement possibilities**. Fitting them with new variables like: customer address, product family, etc.*

***XGBoost** has **reduced the overall computational time** (< 2 hours). In contrast to the RSM engines (> 1 day).*

*The **XGBoost** product model **can be enhanced** setting the parameters of those models with lower AUCs.*

*Our plots in Tableau has helped this team to **understand the challenge** and **discover insight** from raw data. While **detecting problems in original datasets**.*

Last but not least, we have been working side by side as a team. This challenge has resulted in more analytics knowledge. ***Now, DataShots is waiting for new data to analyze.***