LAGOSTE

connect fashion, sports and luxury

Ana Rita Marques R2016700

Ana Sofia José R2016718

Catarina Neves R2016723

Joana Neves R2016724

Pedro Alves R2016734

Contents

Introduction

The present report summarizes the project for building a predictive model, in order to anticipate which of the 5 000 LaGoste's customers (representing 247 500 of the company customers) are more willing to answer positively to the new marketing campaign and consequently buy the new gadget.

To be able to build the predictive model, a pilot campaign was carried out, using a sample of 2 500 clients that were exposed to the campaign. Having this sample data, we analysed and studied these customers' behaviours and characteristics (input variables) to understand which ones influenced the most their decision when facing the campaign (target variable).

This project has a relevant role, because if we don't predict the clients' decisions correctly and consequently the people to whom we send the campaign to, don't buy the product, each of the 4€ spent in contacting each person, will have no return to the company. Therefore, predicting correctly the customers that are willing to acquire the new gadget will not only save a lot of time and money but will also have a very positive effect on LaGoste's campaign profit.

Methodology

The model built in SAS Enterprise Miner represents the relationship between all the variables in the training dataset and the dependent variable - acceptance of the new campaign. Based on this information, it is possible to predict the behaviour of all the 5 000 clients.

In this project, we have two datasets of clients from LaGoste. The first one, the campaign dataset, contains a sample of 2 500 customers, that received the campaign, and their decisions – acceptance or not of the new campaign. The second, the score dataset, contains 5 000 customers, having the same variables as the first one, but without the dependent variable, that will be predicted. As the first one is used to build the best model, the second one is only imported to SAS and used at the end, to predict the score of each customer, representing the probability of acceptance of the campaign.

The methodology used was the SEMMA process, which is depicted in the next figure.

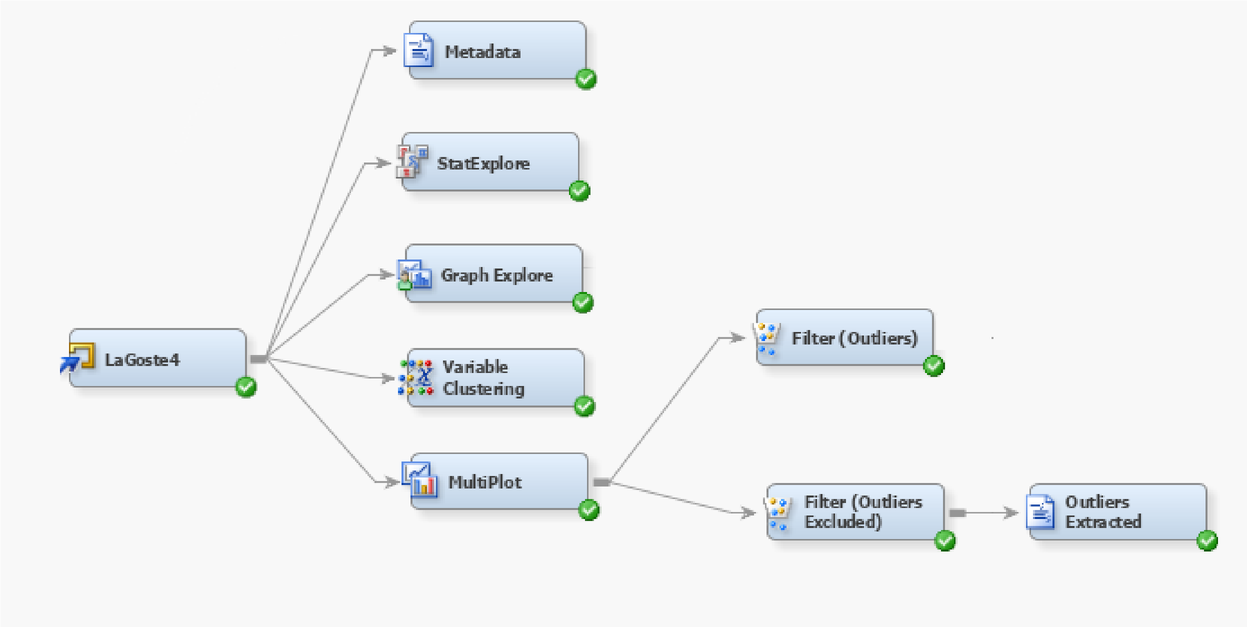


Figure 1 - Sample and Explore

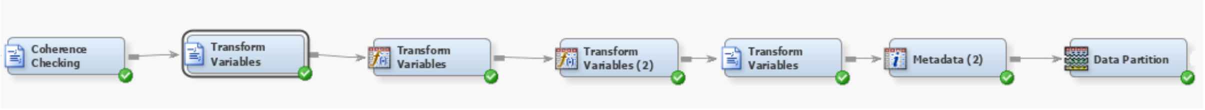


Figure 2 - Modify Part 1

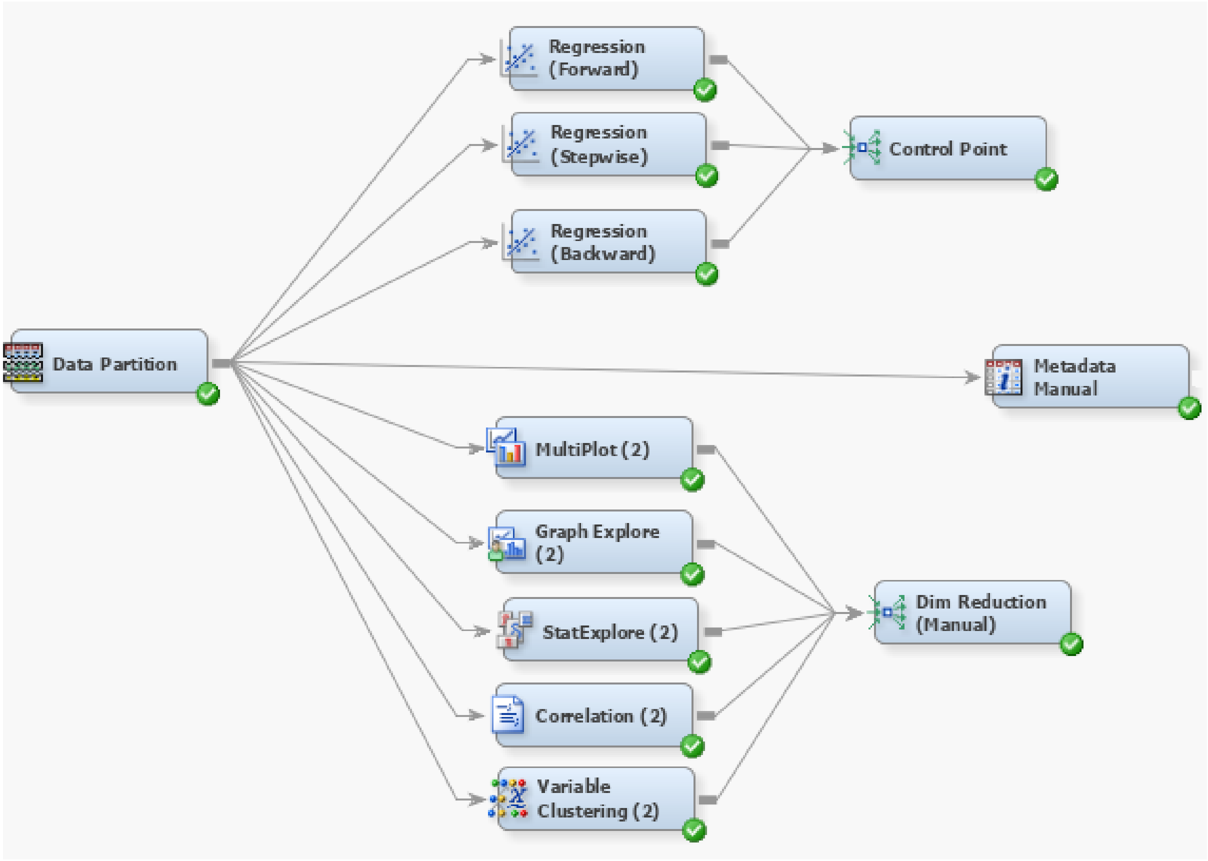


Figure 3 - Modify Part 2

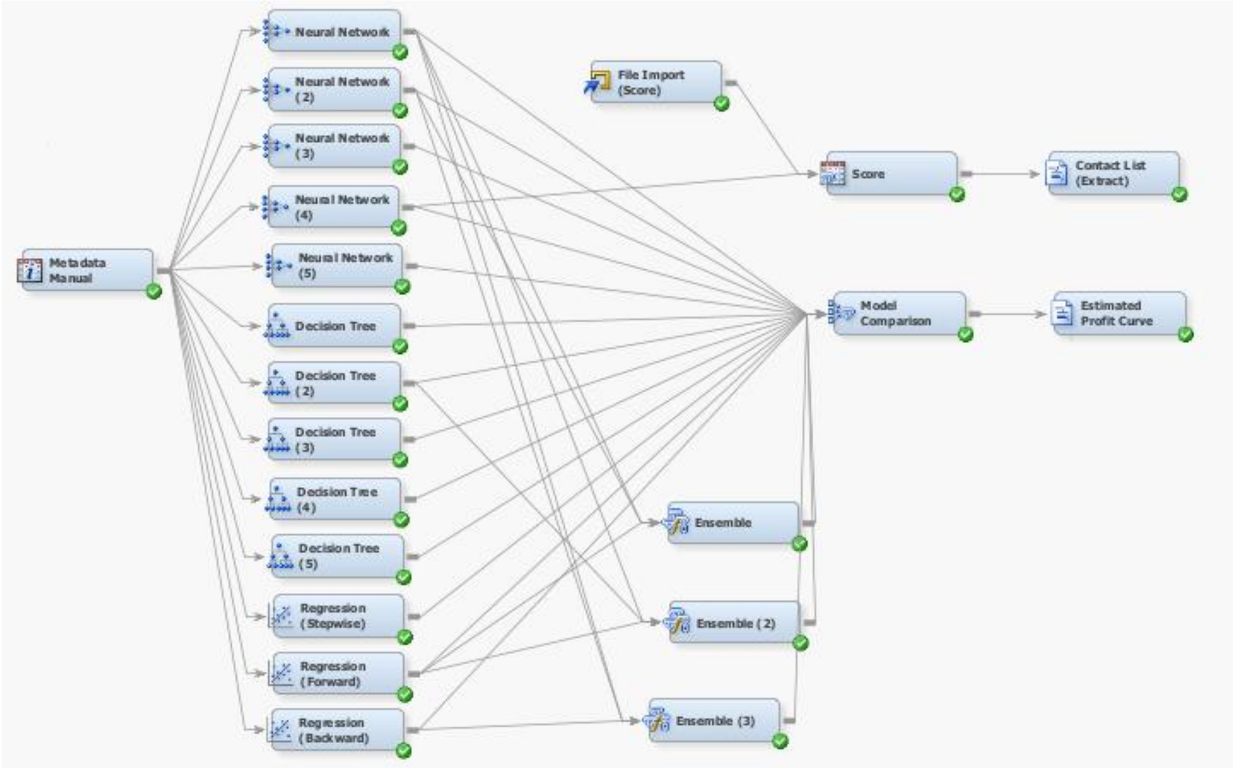


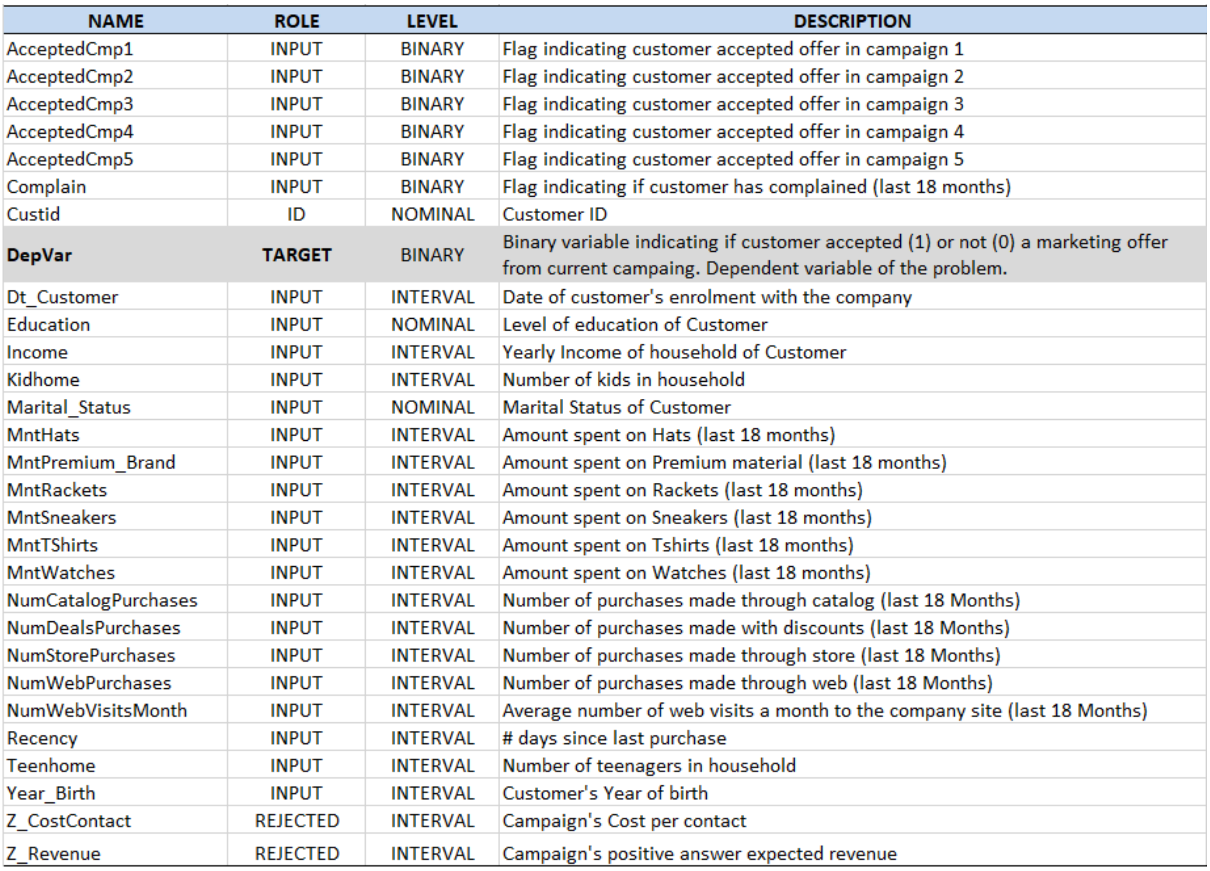
Figure 4 - Model and Assess

SEMMA – Sample Explore Modify Model Assess

Sample

In the first approach we started understanding our data and defined the level and role of each variable in the campaign dataset, according to their measure scale (binary, nominal and interval variables were present) and the part they play in the model (ID, input, target and rejected were attributed). The variables *Z_CostContact* and *Z_Revenue* were excluded because they are not used to build the model and we will only get back to them to calculate the final estimated profit.

The next table has all the attributes that can be found in the datasets and their role and level.



NAME	ROLE	LEVEL	DESCRIPTION
AcceptedCmp1	INPUT	BINARY	Flag indicating customer accepted offer in campaign 1
AcceptedCmp2	INPUT	BINARY	Flag indicating customer accepted offer in campaign 2
AcceptedCmp3	INPUT	BINARY	Flag indicating customer accepted offer in campaign 3
AcceptedCmp4	INPUT	BINARY	Flag indicating customer accepted offer in campaign 4
AcceptedCmp5	INPUT	BINARY	Flag indicating customer accepted offer in campaign 5
Complain	INPUT	BINARY	Flag indicating if customer has complained (last 18 months)
Custid	ID	NOMINAL	Customer ID
DepVar	**TARGET**	BINARY	Binary variable indicating if customer accepted (1) or not (0) a marketing offer from current campaing. Dependent variable of the problem.
Dt_Customer	INPUT	INTERVAL	Date of customer's enrolment with the company
Education	INPUT	NOMINAL	Level of education of Customer
Income	INPUT	INTERVAL	Yearly Income of household of Customer
Kidhome	INPUT	INTERVAL	Number of kids in household
Marital_Status	INPUT	NOMINAL	Marital Status of Customer
MntHats	INPUT	INTERVAL	Amount spent on Hats (last 18 months)
MntPremium_Brand	INPUT	INTERVAL	Amount spent on Premium material (last 18 months)
MntRackets	INPUT	INTERVAL	Amount spent on Rackets (last 18 months)
MntSneakers	INPUT	INTERVAL	Amount spent on Sneakers (last 18 months)
MntTShirts	INPUT	INTERVAL	Amount spent on Tshirts (last 18 months)
MntWatches	INPUT	INTERVAL	Amount spent on Watches (last 18 months)
NumCatalogPurchases	INPUT	INTERVAL	Number of purchases made through catalog (last 18 Months)
NumDealsPurchases	INPUT	INTERVAL	Number of purchases made with discounts (last 18 Months)
NumStorePurchases	INPUT	INTERVAL	Number of purchases made through store (last 18 Months)
NumWebPurchases	INPUT	INTERVAL	Number of purchases made through web (last 18 Months)
NumWebVisitsMonth	INPUT	INTERVAL	Average number of web visits a month to the company site (last 18 Months)
Recency	INPUT	INTERVAL	# days since last purchase
Teenhome	INPUT	INTERVAL	Number of teenagers in household
Year_Birth	INPUT	INTERVAL	Customer's Year of birth
Z_CostContact	REJECTED	INTERVAL	Campaign's Cost per contact
Z_Revenue	REJECTED	INTERVAL	Campaign's positive answer expected revenue

Table 1 - Variables Description, Role and Level

Explore

In this step, the nodes Multiplot, Stat Explore, Variable Clustering, Graph Explore and Filter were used. Here, the objective is to explore the data from a statistic and graphical point of view, in order to better understand our data. We searched for outliers and missing values and tried to apply the best methods to deal with them, either by elimination or replacement of those values.

Starting with the Multiplot (Bivariate analysis), in this node we analysed the different histograms to have some insights on the empirical distribution of the variables, have an idea about the existence of outliers and missing values and to relate the input variables with the target.

Missing values were identified in the following variables: *Income*, *MntHats* and *MntPremium*.

The next histograms display the distribution of the variables that, after visual analysis, may bring us relevant insights.

Regarding the *Income*, the presence of possible outliers was detected, and excluding them, we realized that the individuals with a higher income had a greater acceptance of the pilot campaign than the individuals with a lower income.

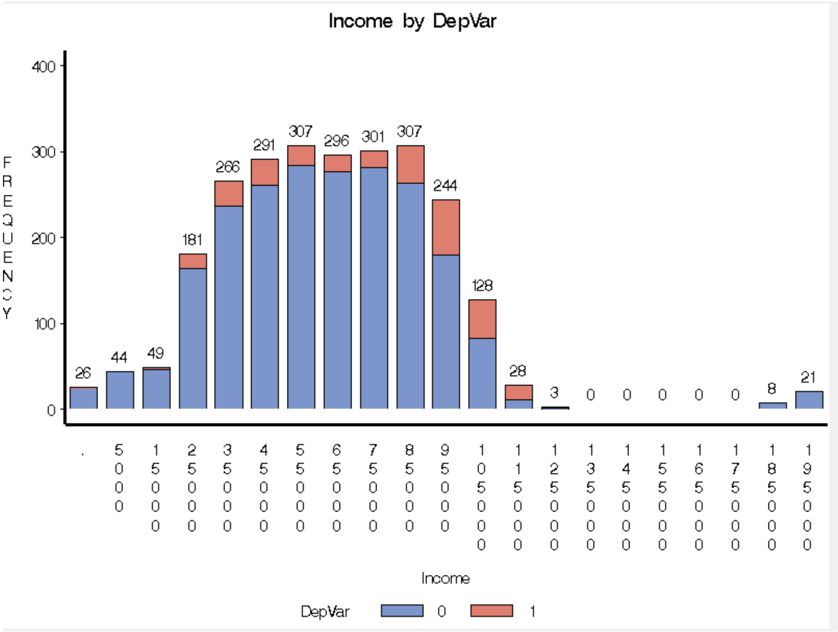


Figure 5 - Income Histogram

In the *Recency*, it is possible to state that the more days had passed since the last purchase, the lower the buying trend is.

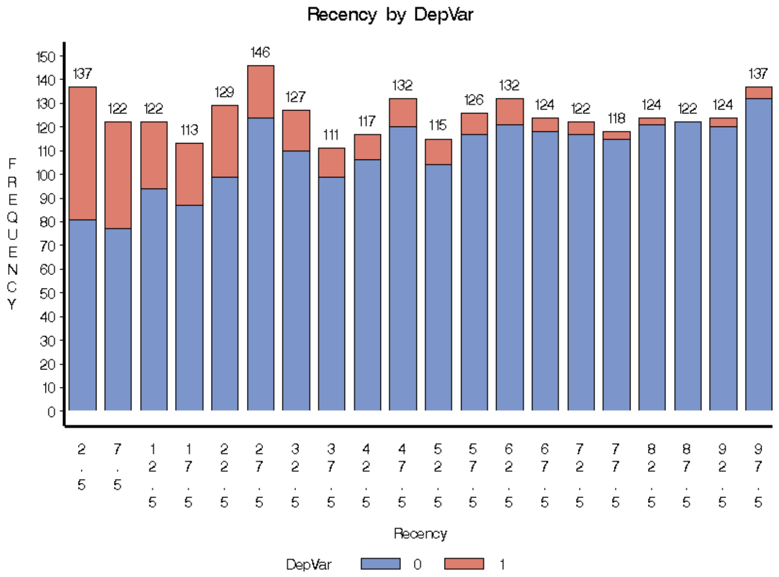


Figure 6 - Recency Histogram

It is also possible to conclude that *Age* does not have a large effect on the acceptance or not of the campaigns, but still we can say that people in the extremes tend to be more receptive to the campaign. Looking at the *Marital_Status*, we can observe that single people are the ones that accept the most the campaign and married people are the ones who accept the least, proportionally.

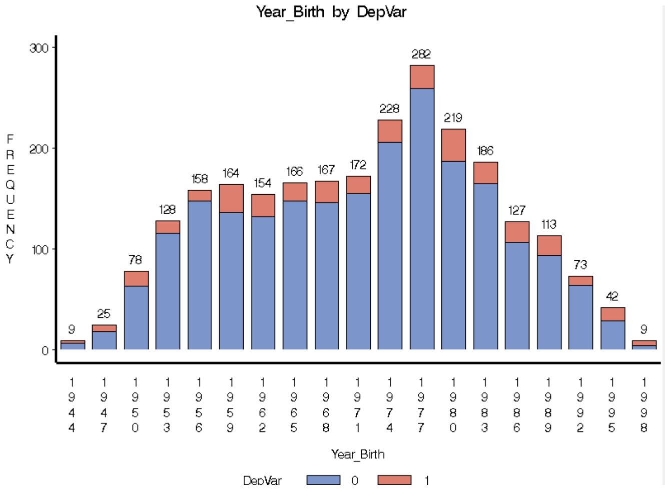


Figure 8 - Year_Birth Histogram

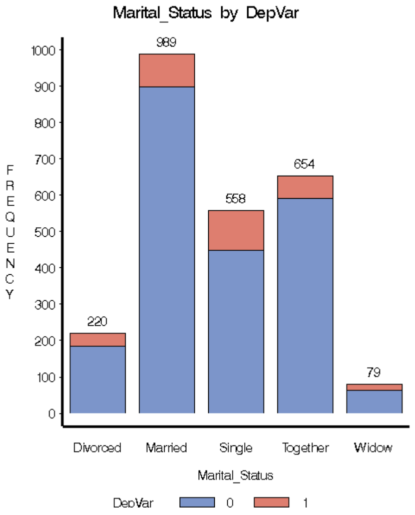


Figure 7 - Marital_Status Histogram

We also realized that generally, who has two children or teenagers at home is less receptive to the campaign.

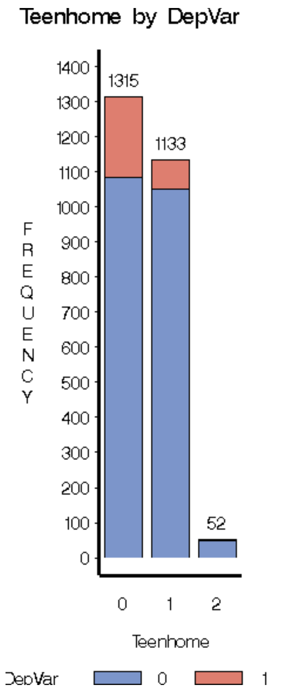


Figure 9 - TeenHome Histogram

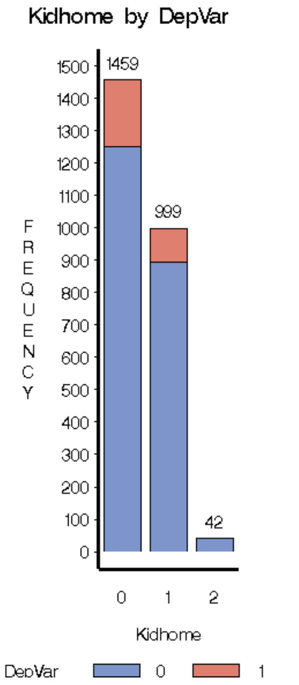


Figure 10 - KidHome Histogram

In the *NumDealsPurchases* graphic, we noticed that there is a trend: people who buy more products with discount, in general, do not accept the campaign.

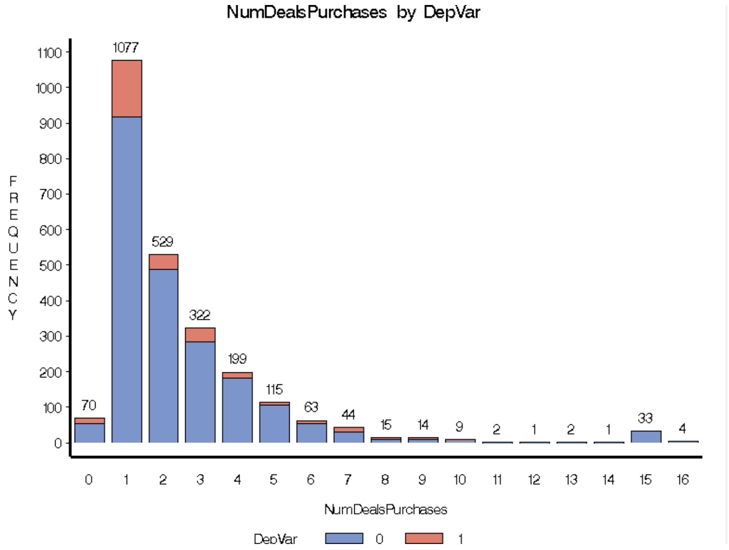


Figure 11 - NumDealsPurchases Histogram

Relatively to the *NumWebVisitPerMonth*, we understand that the clients that have never gone to the website and those who visit it a lot of times (>18) tend to not accept the campaign.

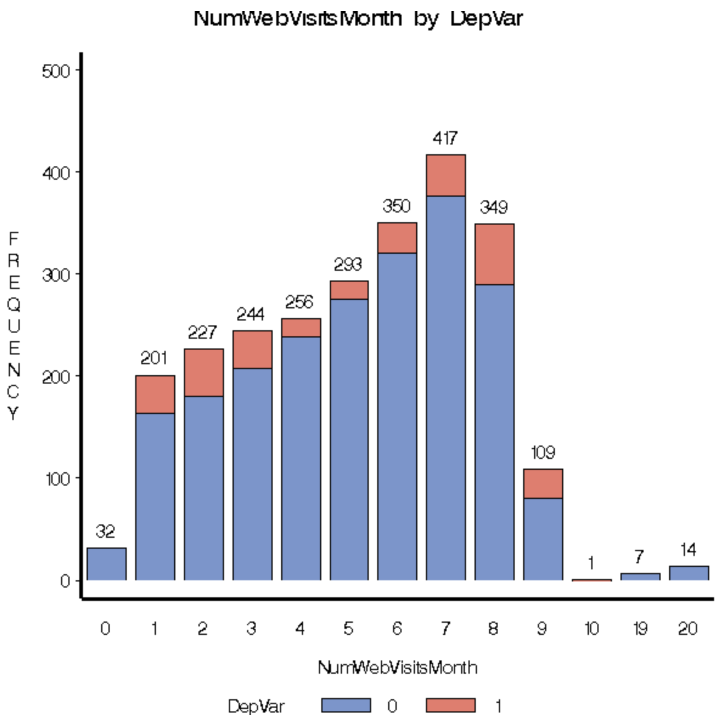


Figure 12 - NumWebVisitsMonth Histogram

About the *Complain* variable, the conclusion is clear, the ones who complained did not buy the advertised product.

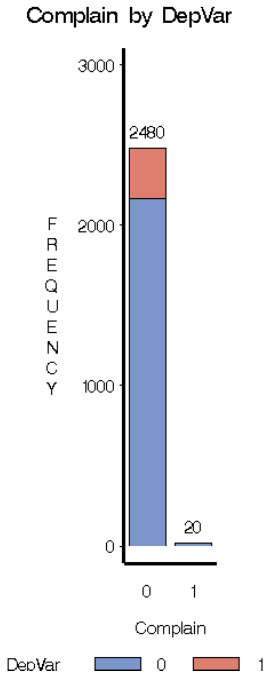


Figure 13 - Complain Histogram

Regarding to the *MntRackets*, we verified that this variable has outliers. Excluding them and analysing the chart, we understand that the people who spend more on rackets are the ones who accept more the campaign, proportionally.

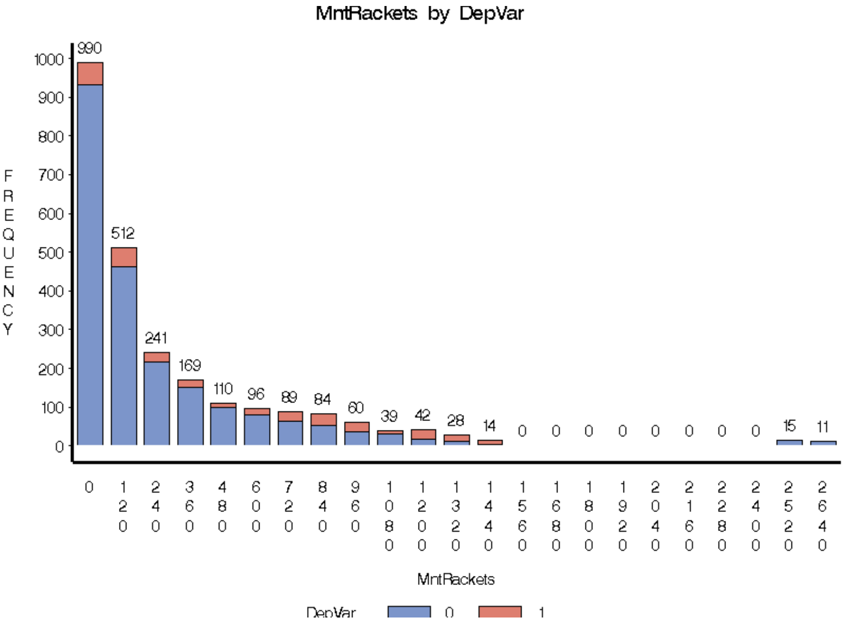


Figure 14 - MntRackets Histogram

About the *NumCatalogPurchases*, we concluded that who made 0, 1, 24 and 25 (the extremes) purchases through the catalog, did not buy our product in the pilot campaign.

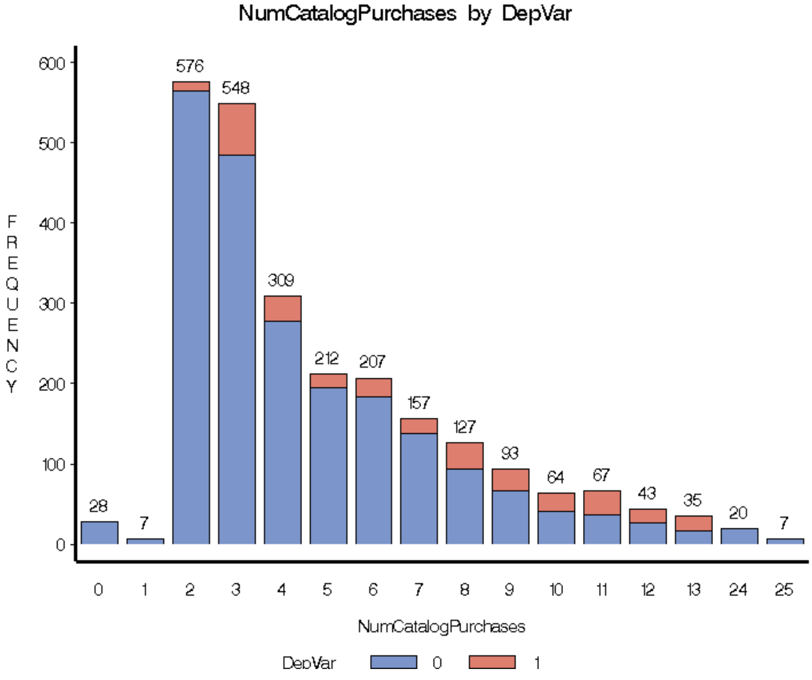
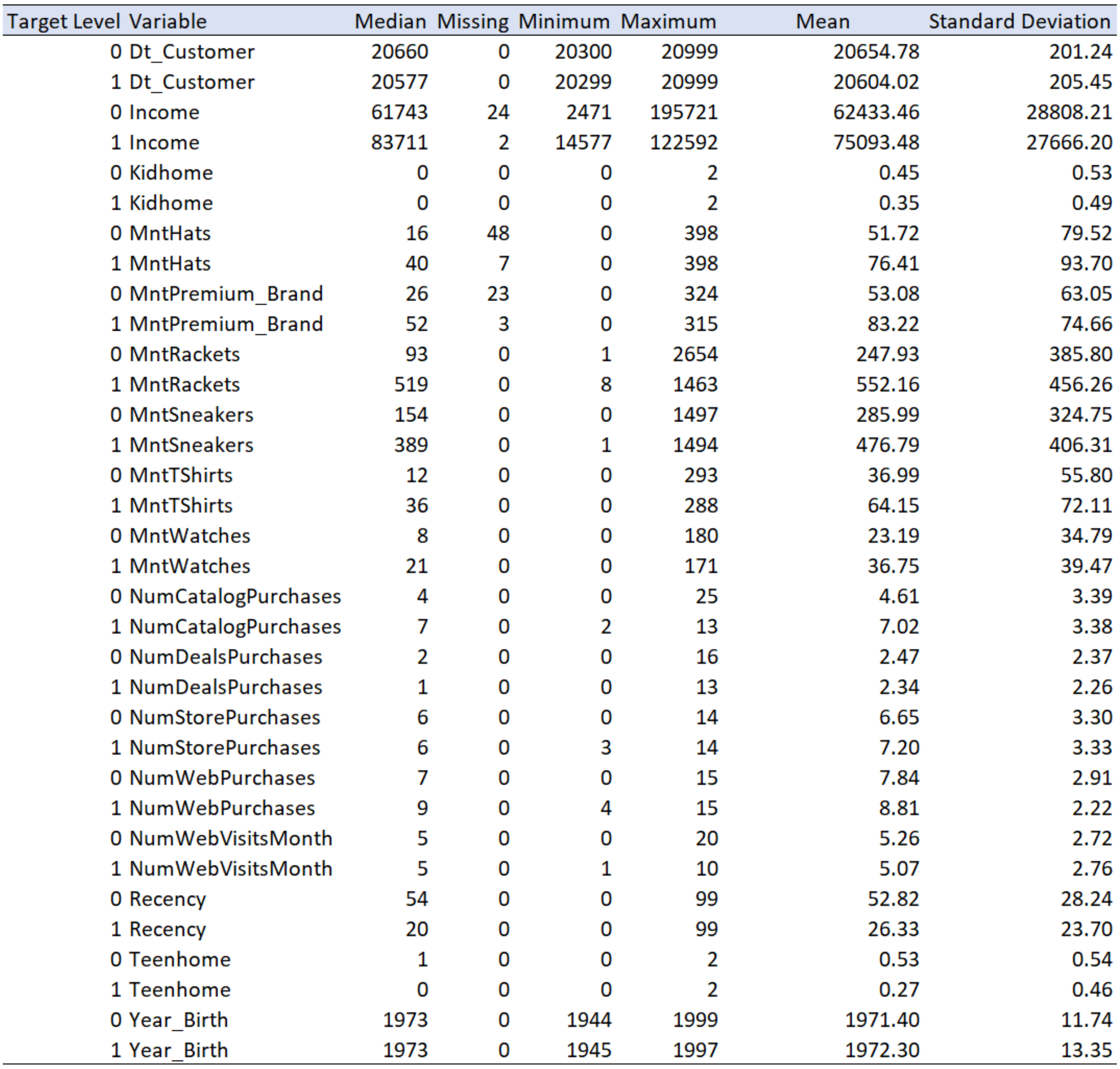


Figure 15 - NumCatalogPurchases Histogram

Analysing the Stat explore node, one of the most important things to notice is the Variable Worth, where a ranking of the existing variables is defined, helping us understand which ones better explain the dependent variable. In this node, we verified that all the class variables have the levels well defined and, also, the existence of missing values. After the analysis of the results we concluded that there are missing values in the variables *Income* (26), *MntHats* (55) and *MntPremium_Brand* (26).



Target Level	Variable	Median	Missing	Minimum	Maximum	Mean	Standard Deviation
0	Dt_Customer	20660	0	20300	20999	20654.78	201.24
1	Dt_Customer	20577	0	20299	20999	20604.02	205.45
0	Income	61743	24	2471	195721	62433.46	28808.21
1	Income	83711	2	14577	122592	75093.48	27666.20
0	Kidhome	0	0	0	2	0.45	0.53
1	Kidhome	0	0	0	2	0.35	0.49
0	MntHats	16	48	0	398	51.72	79.52
1	MntHats	40	7	0	398	76.41	93.70
0	MntPremium_Brand	26	23	0	324	53.08	63.05
1	MntPremium_Brand	52	3	0	315	83.22	74.66
0	MntRackets	93	0	1	2654	247.93	385.80
1	MntRackets	519	0	8	1463	552.16	456.26
0	MntSneakers	154	0	0	1497	285.99	324.75
1	MntSneakers	389	0	1	1494	476.79	406.31
0	MntTShirts	12	0	0	293	36.99	55.80
1	MntTShirts	36	0	0	288	64.15	72.11
0	MntWatches	8	0	0	180	23.19	34.79
1	MntWatches	21	0	0	171	36.75	39.47
0	NumCatalogPurchases	4	0	0	25	4.61	3.39
1	NumCatalogPurchases	7	0	2	13	7.02	3.38
0	NumDealsPurchases	2	0	0	16	2.47	2.37
1	NumDealsPurchases	1	0	0	13	2.34	2.26
0	NumStorePurchases	6	0	0	14	6.65	3.30
1	NumStorePurchases	6	0	3	14	7.20	3.33
0	NumWebPurchases	7	0	0	15	7.84	2.91
1	NumWebPurchases	9	0	4	15	8.81	2.22
0	NumWebVisitsMonth	5	0	0	20	5.26	2.72
1	NumWebVisitsMonth	5	0	1	10	5.07	2.76
0	Recency	54	0	0	99	52.82	28.24
1	Recency	20	0	0	99	26.33	23.70
0	Teenhome	1	0	0	2	0.53	0.54
1	Teenhome	0	0	0	2	0.27	0.46
0	Year_Birth	1973	0	1944	1999	1971.40	11.74
1	Year_Birth	1973	0	1945	1997	1972.30	13.35

Table 2 - Summary Statistics of the original variables

Next, we ran the Variable clustering node, which is used to identify similar variables. Here, we saw the correlation matrix, depicted below, and verified that the highest correlation was between *MntRackets* and *NumCatalogPurchases* (approximately 77%).

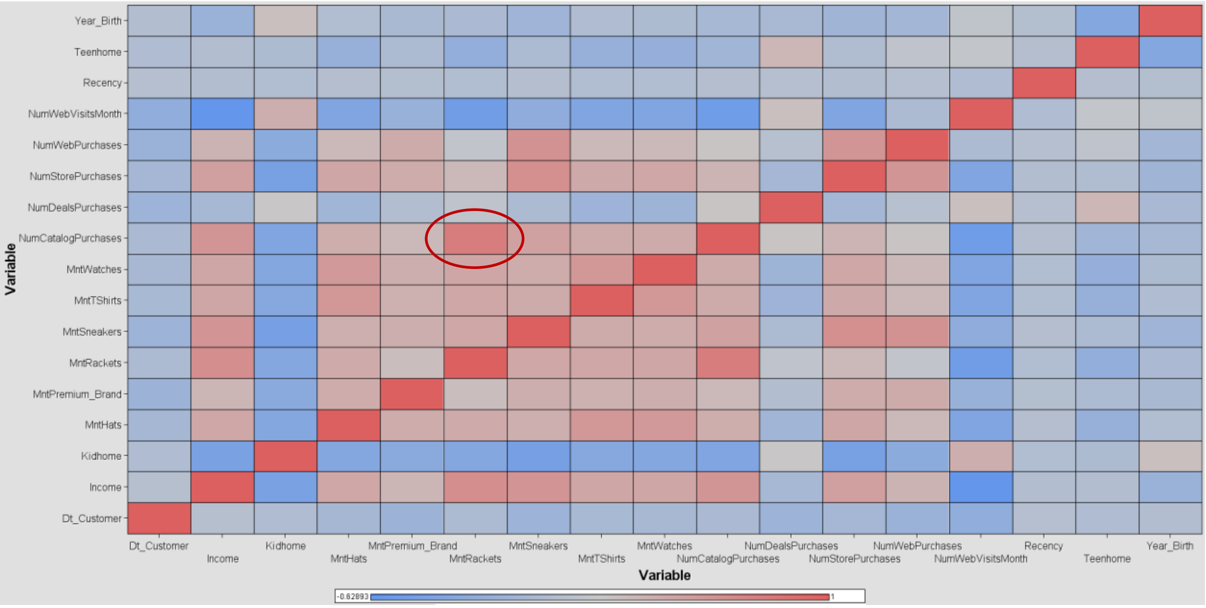


Table 3 - Variable Correlation matrix

Adding just a note about the Metadata SAS code node, it was used to export the metadata referred in the table 1, except the description, to excel, one of the deliverables.

Outliers

In the Filter node, we identified the outliers and then used the interactive interval filter (yellow area) to exclude them from our dataset. The outliers that were excluded were in the *MntRackets*, *Income*, *NumCatalogPurshases* and *NumWebVisitPerMonth* variables.

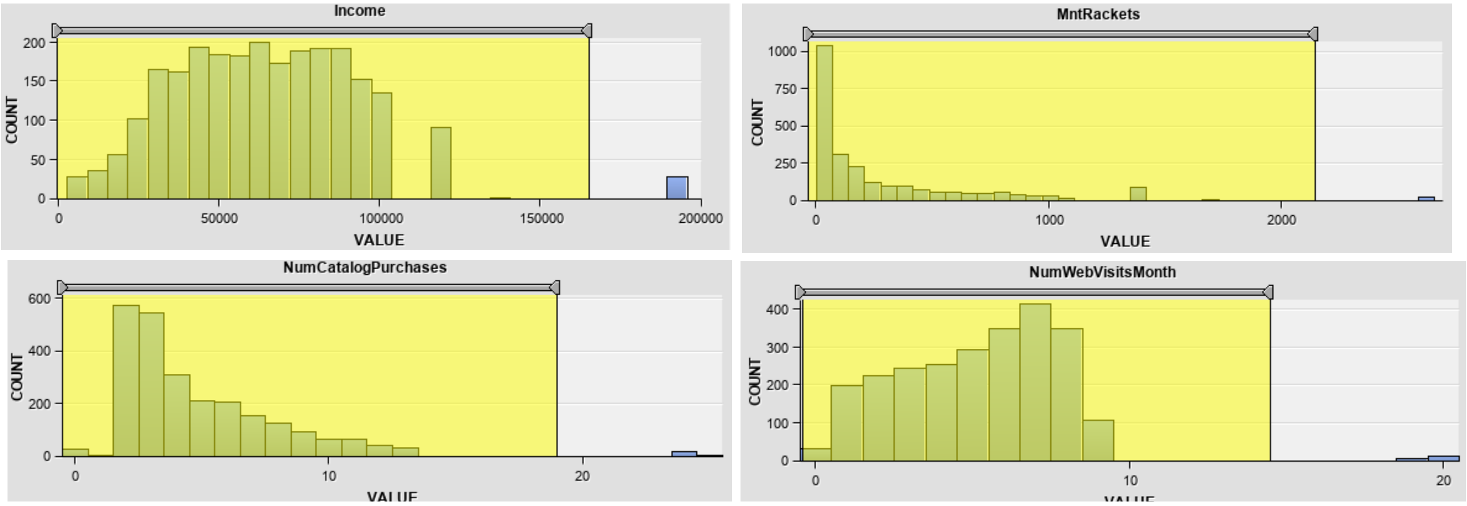


Figure 16 - Outliers Removal

The elimination of outliers is important, because they represent exceptions in the data, and many times they can even derive from errors. If we build a model using them, we will possibly have a result that is not very representative of the population, owing to the fact that we will have a bias due to the existence of extreme values. Some methods can deal with outliers, but others are very sensitive to them.

After the outlier elimination, we verified that the eliminated observations were 63, that represents 2.48% of our dataset, what meant that we were safe to move forward.

Missing Values

As previously said, we found missing values in the variables Income (26), MntHats (55) and *MntPremium_Brand* (26). It is important to decide how to manage observations with missing values and if the correction is the same for the different variables, or if different strategies are in order for each variable. We ultimately decided to impute the missing values, and to achieve this objective, an impute node was used, and its workings and theory are described in the annexes.

Modify

Coherence Checking

A necessary step before further analysis is to check for incoherencies in the data, that can lead us to wrong conclusions. SAS code was written in order to execute these coherence checks.

In order to detect, in an easier way, the possible incoherencies, one variable was created for each of the restrictions (incoherentX, where X=1, 2, ...,14). Here are the coherence rules used:

Incoherence number / Variable created*	**Incoherence code**	**Why this restriction**
Incoherent1	IMP_Income*1.5<SUM(MntSneakers,MntRackets, MntTShirts,MntWatches,IMP_MntHats)	It's not possible for a client to spend more money in LaGoste's products than the total money gained (income)
IncoherentY, where Y=2,3,4,5,6	(AcceptedCmpx NE 0) AND (AcceptedCmpx NE 1), where x=1,2,3,4,5	The campaigns are accepted (1) or not (0), there is no other possibility
Incoherent7	(DepVar NE 0) AND (DepVar NE 1)	
Incoherent8	NumDealsPurchases>SUM(NumCatalogPurchases,NumStorePurchases,NumWebPurchases)	A purchase can be made with a deal or not, but the contrary isn't true. It's not possible to have a deal purchase if a purchase wasn't made
Incoherent9	Year_Birth>YEAR(Dt_Customer)	A person that wasn't born at a certain time can't be registered as a client at that moment
Incoherent10	(AcceptedCmp1= 1 OR AcceptedCmp2=1 OR AcceptedCmp3=1 OR AcceptedCmp4=1 OR AcceptedCmp5=1) AND Recency>365	All the campaigns were done in the past year so if a client accepted one or more of the campaigns he has necessarily to have a recency smaller than 365
Incoherent11	SUM(NumCatalogPurchases, NumStorePurchases,NumWebPurchases) <SUM(AcceptedCmp1,AcceptedCmp2, AcceptedCmp3, AcceptedCmp4, AcceptedCmp5)	If a client accepts x campaigns, he has necessarily made at least x purchases, because each accepted campaign means a purchase made
Incoherent12	NumWebPurchases>0 AND (NumWebVisitsMonth*18)=0	A person can't have any web purchases if he hasn't visited the website
Incoherent13	SUM(NumCatalogPurchases,NumStorePurchases, NumWebPurchases)=0 AND SUM(MntSneakers,MntRackets,MntTShirts, MntWatches,IMP_MntHats)>0	It's not possible for a client to have spent money in LaGoste's products if he didn't any purchase.
Incoherent14	SUM(NumCatalogPurchases,NumStorePurchases, NumWebPurchases)>0 AND SUM(MntSneakers,MntRackets,MntTShirts, MntWatches,IMP_MntHats)=0	It's not possible for a client to do a purchase and spend no money.

Table 4 - Incoherence Specifications

We think it would also be important to apply this same coherence checking to the score set, because if we are predicting an individual's behaviour based on wrong or non-sense information we have about him, we will certainly not get to the best results. The same happens to the missing values of the score dataset, that should also be target of the same imputation as the missing values of the campaign dataset. If an observation of the score dataset was affected by some of these processes, the observation should be flagged, for the analyst to take special attention to it.

After running the coherence check code node, we found 6 incoherencies. All of them with the same problem and related to the same coherence check rule (Incoherent12) - the person did not make any web visit, however, made web purchases. This can be the result, not of an error on insertion, but from the format selection for the storage of this variable. The variable should, in fact, be a floating point value obtained by dividing the total visits by the time span, but was instead saved as an integer through rounding. The rounding applied causes loss of information, making someone who already visited us up to 9 times, to have an average visit number rounded down to 0.

Having this in mind, we decided to change the value of the *NumWebVisitsMonth* variable, on the incoherent observations found. In order to differentiate the clients that had already a value of 1, in the original dataset, we chose to change the incoherent values to 0.5.

Transform Variables

In this step, the objective is to transform existing variables or create new ones based on the originals to obtain a higher predictive power.

There are many input candidates to models that can or should be transformed in order to remove skewness or to continue processing them by applying business rules. In our case, we standardized some variables before adding them, as explained in more detail later in the report.

The following table presents the transformations made and respective new variables.

ID	Transformed Variable	Description
1	Age=2017-Year_Birth	Age of the customer when the database was extracted
2	AmountSpentperPurchase= TotalAmountSpent/TotalPurchases	In average, how much the client spent on each purchase made
3	BIN_Age = Age in 3 buckets	Variable Age divided in categories, created with bucket binning (3 bins)
4	BIN_IMP_Income = IMP_Income in 5 buckets	Variable IMP_Income divided in categories, created with bucket binning (5 bins)
5	BIN_Recency = Recency in 3 buckets	Variable Recency divided in categories, created with bucket binning (3 bins)
6	HigherEducation: 1, if Education in (“Graduation”, “Master”, “PhD”) 0, otherwise	Binary variable that tells if the person has a higher education degree/high level of education (1)
7	ImportanceA=STD_InvRecency + STD_TotalAmountSpent	A measure of the clients value to the company, based on recency and total amount spent. The bigger are the two parcels, the better
8	ImportanceB= STD_TotalAmountSpent + STD_TotalPurchases	A measure of the clients value to the company, based on the total amount spent and the total purchases. The bigger are the two parcels, the better
9	ImportanceC=STD_TotalPurchases + STD_InvRecency + STD_TotalAmountSpent	A measure of the clients value to the company, based on the total amount spent, the recency and the total purchases. The bigger are the two parcels, the better
10	Income_18months=IMP_Income*1.5	Average income per 18 months
11	IncomeSpentOnUs=TotalAmountSpent/Income_18months	Proportion of the total income, that was spent on purchases in the company, in the last 18 months
12	InvRecency=99-Recency	Importance of the client in terms of the last purchase
13	Log_IMP_Income=log(IMP_Income)	Logaritm of the IMP_Income variable
14	Log_TotalAmountSpent=log(TotalAmountSpent)	Logaritm of the TotalAmountSpent variable
15	Log_TypeProductsPurchase=log(TypeProductsPurchase)	Logaritm of the TypeProductsPurchase variable
16	MntXRatio=X/TotalAmountSpent, where X = IMP_MntPremium_Brand, MntSneakersRatio, MntRacketsRatio, MntTShirtsRatio, MntWatchesRatio, MntHatsRatio	Proportion of money spent in each category over the total amount spent in the company.
17	OPT_Age = Age formed with created binning	Variable Age divided in categories, created with optimal binning (the resulting bins were 2)
18	OPT_IMP_Income = IMP_Income created with optimal binning	Variable IM_Income divided in categories, created with optimal binning (the resulting bins were 3)
19	OPT_Recency = Recency created with optimal binning	Variable Recency divided in categories, created with optimal binning (the resulting bins were 4)
20	RatioWebVisits_Purchases=NumWebPurchases/NumWebVisitsMonth	Proportion of times that the client visited the website and bought any products over the total number of visits
21	STD_InvRecency = InvRecency standardized	InvRecency standardized
22	STD_TotalAmountSpent = TotalAmountSpent standardized	TotalAmountSpent standardized
23	STD_TotalPurchases = TotalPurchases standardized	TotalPurchases standardized
24	TogetherStatus: 1, if Marital_Status in (“Married” or “Together”) 0, otherwise	Binary variable that tells if a person lives with someone or not
25	TotalAcceptedCmp=AcceptedCmp1+AcceptedCmp2+AcceptedCmp3+AcceptedCmp4+AcceptedCmp5	Total number of accepted campaigns, of the 5 done previously
26	TotalAmountSpent=MntSneakers+MntRackets+MntTShirt+MntWatches+IMP_MntHats	Total amount spent in all products
27	TotalChildrenBinary: 1, if Kidhome+Teenhome>0 0, if Kidhome+Teenhome=0	Binary variable that tells if there are any kids or teenagers in the household
28	TotalChildrenDiscrete=Kidhome+Teenhome	Total number of children and teenagers in the household
29	TotalPurchases= NumStorePurchases+ NumCatalogPurchases+NumWebPurchases	Total purchases made in last 18 months
30	TypeProductsPurchase=Sneakers+Rackets+TShirts+Watches+Hats	Sum of the previous 5 binary variables, that gives how many type of products the client has bought
31	Whatkids: 0, if Kidhome=0 and Teenhome=0 1, if Kidhome=0 and Teenhome>0 2, if Kidhome>0 and Teenhome=0 3, if Kidhome>0 and Teenhome>0	Multinomial variable, that tells if there are only kids in the household (2), only teens in the household (1), both (3) or none (0).
32	X: 1, if MntX>0 0, if MntX=0 where X=Sneakers, Rackets, TShirts, Watches, Hats	Binary variables for each of the product types, in order to know if the client ever bought that specific kind of product.
33	XPurchasesRatio=NumXPurchases/TotalPurchases, where X = Catalog, Store, Deals, Web	Proportion of purchases of each category over the total number of purchases.
34	YearsWithUs=2017-Year(Dt_Customer)	Number of years since the person has become LaGoste’s client

Table 5 - Transformed Variables

Based on the information referred in “Data Mining and Predictive Analytics”, 2nd Edition, Wiley (page 722 – 724), we understand that typically individuals who buy from just one or two categories tend not to adhere to campaigns, since they know exactly what they want to buy, and people who buy different products with us, tend to adhere more to campaigns. Therefore, we decided to create binary variables for each of the product types (*Sneakers*, *Rackets*, *T-shirts*, *Watches* and *Hats - variables 32*) in order to understand if a client ever bought that specific type of product and then, sum all of them (variable 30) to know how many type of different product categories the client has bought.

Regarding the variables *TotalAmountSpent*, *TotalPurchases* and *InvRecency (from Recency)*, we decided to standardize them since we were going to create composite variables that better represent customer importance (*ImportanceA*, *ImportanceB* and *ImportanceC* - 7, 8 and 9), and we wanted them to have equal weights, instead of unknown random weights due to different scales. We at first thought about multiplying the variables, but then realized that multiplying Z-standardized variables wouldn’t have the result we were expecting. Explaining further the problem, considering an observation with a value of 0 in the Recency variable, the new variable would always have the value 0, whether the customer was very good or really bad in the other dimensions. Considering a new case with 2 positive values, the multiplication would have the same result than if the 2 were negative. This, in retrospective, could have been solved by a different normalization. We ended up summing the variables to create the 3 new variables.

When analysing *Education* and *Marital_Status* we verified that we had different categories that could have a similar behaviour, so we decided to join the categories that were identical in the *Education* variable and then did the same with the ones in *Marital_Status*, creating the dummy variables 6 and 24, respectively.

We decided to apply the log transformation to the variables *TypeProductsPurchase*, *IMP_Income* and *TotalAmountSpent* (15, 13 and 14) since we wanted to see if it is possible to remove some skewness from the variables, so that we can get a different view of the data. Many researchers want to correct the skewness because of the assumptions many algorithms make regarding the shape of the data, namely normal distributions. The log transformation is perhaps the most often used transformation to correct for positive skew (all the variables had only positive values in the distribution).

Relatively to the binning variables, we chose this method because it can be a good strategy when in the presence of variables with long tails, multi-modal or with spikes. A binned version of a variable may be a better representation for modelling rather than correcting awkward distributed variables. So, we decided to divide the variables *Age*, *Recency* and *IMP_Income* into bins (variables 3, 4 and 5), in order to capture regions centred on some peaks, forming groups.

In this case, we used the bucket option that creates buckets by dividing the input in n equal sized intervals and grouping the observations into the n buckets.

Then, we created another equal node, but here we did not use the bucket option for binning, but the optimal, in order to understand which method gives us better results (variables 17, 18 and 19). The Optimal Binning splits a variable into n groups regarding to the target variable. When we created the binning variables, we decided to also maintain the original ones, so that we could compare their performance.

We opted to use the year 2017, as the year when dataset was extracted from the database, to form the variables *Age* and *YearsWithUs*. We made this decision after analysing the dataset and realizing that the last purchases were made in 2017.

Regarding the variable *InvRecency*, the objective of its creation was to consider the value 0 as the worst possible value and have the higher numbers representing better clients. So, we inverted the *Recency* variable by subtracting the value of each observation, from the maximum of the variable.

At this point we decided to reject some variables, so that they didn't enter in the next phase, where they would or could negatively affect the correlations. This said, we rejected the variables that were only created in order to achieve a final one (the variables *Sneakers*, *Rackets*, *T-shirts*, *Watches* and *Hats*) and the ones created to help us knowing which ones of the coherence check rules were giving us incoherencies.

Data Partition

In this node, the campaign dataset that has been processed so far, was divided in two: the training dataset (70% of the original dataset), that will help us to build the model, and the validation dataset (30% of the original dataset), used to control and tune the model parameters mainly to avoid overfitting, by checking if the model is not too fitted only to the training data. Thus, we can have a model that is suitable to all other individuals. The partitioning method was changed to stratified so we could have the same proportion of accepted and non-accepted in both datasets.

Regressions

Forward Regression

Using this procedure, the selection of variables is done computationally with three steps. First, the model starts with no variables and then selects the first one to enter the model using the one most correlated with the target variables. Then, for the remaining variables, is calculated the sequential F-statistic. After that, the variable with the larger sequence F-statistic is selected and then the model is tested. If the model is not significant, the algorithm stops and the model discards the variable. Otherwise, that variable enter in the model and the process continues to find other variables.

Backward Regression

This procedure starts by performing the regression with all variables in the model. Then, the partial F-statistic is calculated for each one and the smallest partial F-statistics is noted. After, if the variable with that minimum F is not significant, it is removed from the model and all the steps are repeated until all the variables in the model are significant. If is significant, then the procedure stops and we have a model to report.

Stepwise Regression

This approach is similar to the forward regression, but here the variables are tested twice. In this procedure, a significant variable can pass to nonsignificant because it performs at each step a partial F-test, using the partial sum of squares, for each variable currently in the model. If there is a variable in the model that is no longer significant, then the variable with the smallest partial F-statistic is removed from the model.

We decided to use these three regressions to compare the results with the variables worth and the correlations in order to find the most suitable variables for our model.

	Forward	Stepwise	Backward
AcceptedCmp1			X
AcceptedCmp2			X
AcceptedCmp3			X
AcceptedCmp4			X
AcceptedCmp5			X
AmountSpentperPurchase			X
BIN_IMP_Income			X
BIN_Recency	X		
CatalogPurchasesRatio	X	X	X
DealsPurchasesRatio	X	X	X
Education			X
HigherEducation		X	
IMP_Income	X	X	X
IMP_MntHats			X
ImportanceA			X
InvRecency	X	X	
Marital_Status			X
MntHatsRatio			X
MntPremiumRatio	X	X	X
MntRackets			X
MntRacketsRatio	X		X
MntSneakers			X
MntSneakersRatio			X
MntTShirts			X
MntTShirtsRatio			X
MntWatchesRatio	X		
NumCatalogPurchases			X
NumDealsPurchases	X	X	
NumWebVisitsMonth	X	X	X
OPT_Recency	X		
TogetherStatus	X	X	
TotalAcceptedCmp	X	X	
TotalChildrenBinary	X	X	X
TotalChildrenDiscrete	X	X	

Table 6 - Regressions

Variable Clustering and Correlation Matrix

In this node we could see and check the variables that had higher correlation. We used the Spearman method because not all our variables follow a linear distribution. This is important to analyse, due to the possible problems of redundancy.

After analysing the spearman correlation matrix, we identified the correlations with a score equal or higher than 0.8 (table with the correlations annexed).

After that, we used the StatExplore node to see the Variable Worth of each variable when it comes to their discriminating power over the dependent variable. The graph below shows the variables with the higher worth, and we identified the ones used, the ones not used and the ones we were unable to use due to high correlations.

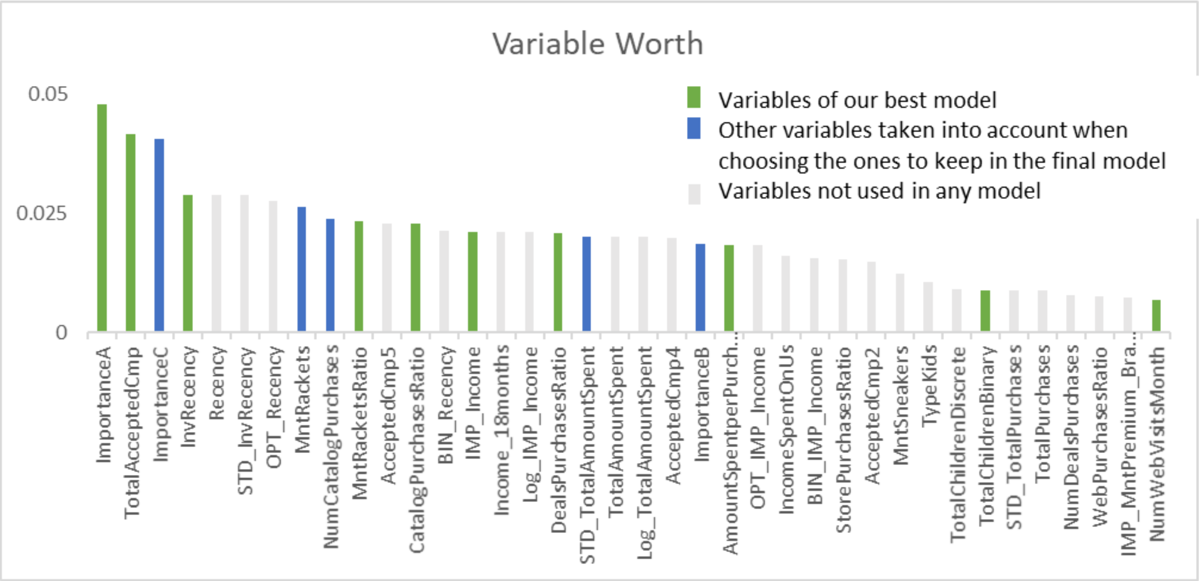


Figure 17 - Variable Worth

Metadata node - Variable Selection

In this step we must select the variables to include in our model. It is important to remember that our goal is to achieve the highest profit possible, through the created model, taking into consideration that any additional variable above 10, will penalize the model profit by 50 €.

For this selection several criteria were used, obtaining information from various sources namely the correlation matrix, the variable worth and regressions. We pre-selected a small subset of variables to test in several models to compare the results and confirm which one would be the best model.

Model

In this part, after analysing the variables to select them, we used several modelling techniques such as neural networks, decision trees, regression models and the Ensemble node (explained in annex), in order to find the best model. Therefore, we made several models with different parameters and some different variables, to evaluate them (the final part of the SEMMA - Assess) and verify which one had the best performance. Information on settings selected for each node are available in assess where they are evaluated as well.

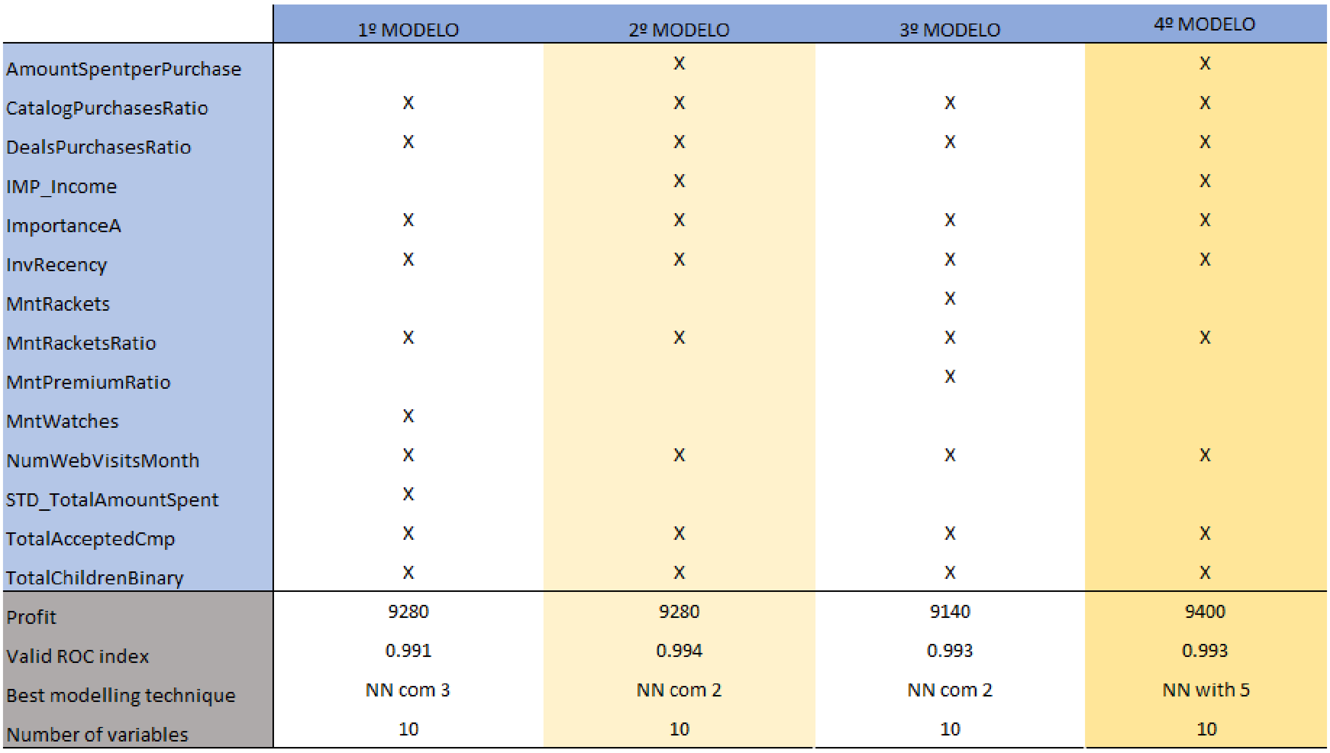
Assess

In this part of the SEMMA process, after creating several models, they were compared in terms of the profit, the ROC index and the number of errors. The ROC index over the validation set is the area under the test ROC (Receiver Operating Characteristics) Curve and tell how good is the model on distinguishing between accept/not accept. So, it measures the discrimination power. Then, after choosing the best model, that one will be applied to the score dataset in order to find the customers that are more willing to accept the campaign (so that they can be contacted).

In order to find the highest profit, we used an iterative process, testing sets of variables, trying to find the best ones, and then running multiple models and tuning the parameters. This process was repeated, comparing validated ROC and profit, until a final set of variables and best model fit was found.

Beside this, we tested the best model with more variables to see if we could achieve higher profits even with the penalty incurred. We were unable to get higher profits with this attempt.

The best models achieved were:



	1º MODELO	2º MODELO	3º MODELO	4º MODELO
AmountSpentperPurchase		X		X
CatalogPurchasesRatio	X	X	X	X
DealsPurchasesRatio	X	X	X	X
IMP_Income		X		X
ImportanceA	X	X	X	X
InvRecency	X	X	X	X
MntRackets			X	
MntRacketsRatio	X	X	X	X
MntPremiumRatio			X	
MntWatches	X			
NumWebVisitsMonth	X	X	X	X
STD_TotalAmountSpent	X			
TotalAcceptedCmp	X	X	X	X
TotalChildrenBinary	X	X	X	X
Profit	9280	9280	9140	9400
Valid ROC index	0.991	0.994	0.993	0.993
Best modelling technique	NN com 3	NN com 2	NN com 2	NN with 5
Number of variables	10	10	10	10

Table 7 - Final Model comparisons

After the models tested, we confirmed that the best model, with certain definitions, has a profit of 9280, achieving a Valid ROC index of 0.994. After obtaining that model, it was fine tuned with other definitions/parameters, with the same variables, obtaining even greater results. Therefore, with the following definitions, we reached the best solution.

Neural Networks
Model Selection Criterion - Profit/Loss
Continue Training - No
Optimization (Maximum Iterations)- 100
Optimization (Training Technique)- Back prop

Regression
Selection Criterion - Profit/Loss
Use Selection Default – Yes

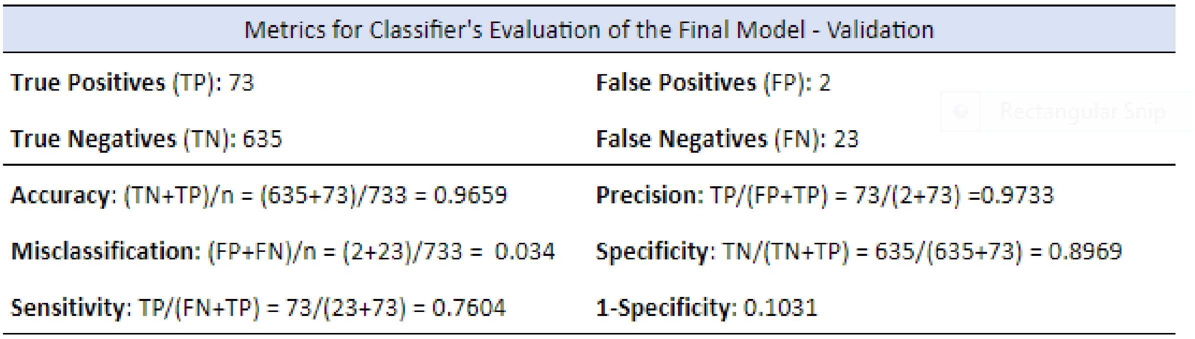
Ensemble
Predicted Values - Maximum
Posterior Probabilities - Voting

The other definitions are the default ones.

The best model

After all those models, we found the best one. Therefore, this model as the following variables: *AmountSpentperPurchase, CatalogPurchaseRatio, DealsPurchaseRatio, IMP_Income, ImportanceA, InvRecency, MntRacketsRatio, NumWebVistisMonth, TotalAcceptedCmp* and *TotalChildrenBinary.*

The best result was achieved using a Neural Network with 5 hidden units, allowing us to have an expected profit of 9 400 € based on the validation set and a Valid ROC index of 0.993. Relatively to other measures based on the validation set (733 observations), to figure out if the model is a really great one, we can observe:



Metrics for Classifier's Evaluation of the Final Model - Validation	
True Positives (TP): 73	**False Positives** (FP): 2
True Negatives (TN): 635	**False Negatives** (FN): 23
Accuracy: (TN+TP)/n = (635+73)/733 = 0.9659	**Precision**: TP/(FP+TP) = 73/(2+73) =0.9733
Misclassification: (FP+FN)/n = (2+23)/733 = 0.034	**Specificity**: TN/(TN+TP) = 635/(635+73) = 0.8969
Sensitivity: TP/(FN+TP) = 73/(23+73) = 0.7604	**1-Specificity**: 0.1031

Based on these measures, we can conclude that our best model seems to be a good model, because the misclassification value is low and the accuracy is really high.

Moreover, we must check the overfitting to see if the model is too accurate to the training dataset. Therefore, we must compare the ROC indexes between the validation set and the training set. Using the data from the ROC indexes, we can conclude that there is no overfitting, because the model fits very well both to the training and, most importantly, to the validation set.

Train ROC index = 0.995 Valid ROC index = 0.993

Conclusion

Based on the model, we can conclude that 750 customers should be contacted (15% of the 5000 customers). The last customer to be contacted was assessed has having a probability of acceptance of 0.189. This is the "cut-off" selected by the model. It derives from the relation of cost to contact and profit from acceptance (20 to 4, a 5:1 ratio) that states that 4 non-acceptances will be compensated by a sale. Having this in mind, all the other customers, which have an equal or lower probability of 0.189, are not worth contacting.

So, the model predicted that 620 customers will accept the campaign and 130 will not accept, predicting 9 400 € of estimated profit (620*20 -750*4).

To conclude, it is important to refer the relevance of this kind of analysis. Using the predictive modelling it was possible to find which costumers to contact, predicting their willingness to accept the campaign. Based on that, it was possible to estimate a good profit, something that was really difficult if the company had contacted the customers randomly, which would inevitably result in high losses, which was confirmed by the losses incurred when contacting the sample.

References

Dean Abbott (2014) "Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst", Wiley

Daniel T. Larose, Chantal D. Larose (2015) "Data Mining and Predictive Analytics", 2nd Edition, Wiley

Annexes

Impute Node

Although the differentiation between types of missing values is rarely used in predictive modelling, we find it important to know if we can determine what type of missing values we have in our dataset since different strategies are suitable to different types of missing values. Missing values are divided in MCAR (Missing Completely At Random), MAR (Missing At Random) or MNAR (Missing Not At Random). Although the names are similar there are nuances among them.

Missing values are considered MCAR, when the fact that there is a missing value, is not dependent on other variable values, being actually independent, and knowing the value of another variable does not gives us information on the actual missing value, although we can sometimes construct a model to try and estimate the missing values. These missing values typically derive from a problem in data collection or storage.

The main difference to MAR is that here, the value is missing not in a random way but actually with a relation to another variable or variables, or with a specific reason. As an example, on a questionnaire a question can be skipped depending on the values of previous questions, but we cannot directly determine the missing value. For instance, if at the time when the data was collected, we were not interested in knowing the total money spent, on a certain good, by people with an income above 100 000€/ year, as they were not part of target population, we know it is related with the income, but we cannot know the missing value.

When we have missing values from MNAR type, we know the values are related with a variable or a set of variables and we can directly know the value. We do not ask if a family with yearly income above 100.000€ /year would have state granted family support since we know that they are not eligible. In this case the missing values could be replaced by 0. In this case if we used the average, median or other replacement we would be causing a bias in our data.

If we consider the missing values to be MCAR, we can remove the observations, use some substitution like the median, mean, a matching distribution or another imputation method, since the missing values are random. If they are considered MAR or MNAR these substitutions will likely introduce a bias in our model. In all of these cases, advanced imputation should be used, the method applied will be dependent on the particular dataset.

We considered that in the 3 variables where we have missing values (*Income*, *Mnt_Hats* and *Mnt_PremiumBrand*), these are all considered MCAR. In the case of *Mnt_Hats* and *Mnt_PremiumBrand* we first thought that maybe the missing values corresponded to people that started being our clients when these kinds of products weren't sold yet. In this case we would have missing values because columns for the variable would not be available in the database at the time of entry, making these a case of MNAR. When consulting the dataset we noticed that these missing values had different dates for client creation, so the hypothesis was rejected, and we started treating the variables as MCAR.

We were then faced with the decision of which method to use to impute the variables. The mean is the first option but is not as robust as the median. The median although more robust than the mean, is subject to the same problem, imputing to many values with the mean can cause an increase in a single point of the distribution, causing errors in our model, not in regression algorithms, but mainly in neural networks.

The following best practice is to assign random values which would not change the distribution, but in some cases, we know we can get better results by using other variables (except the target) to predict the missing values, using a tree classification. We decided on this last option. Although SAS does not explicitly how he does the final imputation when the leafs are settled, it may use the mean, median, distribution or predictors weights, using a small regression to obtain the final estimate. This method also has the advantage of tending not to change the distribution of data.

Care must be taken that data that is presented to the model for prediction, should be subject to the same method of imputation, and we should be sure that the reason for missing values remain the same.

Other advanced methods are available, but we decided not to study them at this point.

Ensemble Node

The ensemble node tries to capture the best of each model, by assuming that different models can capture with different precision different relations between the variables. Then it is possible to implement a method that considers the predictions emerging from different models and define a single score. Since that in our case we are creating a model for the probability of a customer to adhere or not to the campaign, the values can be bound between 0 and 1, normalized.

Then we can establish different methods to try and determine if it will buy or not. SaS miner presents us with a few options:

A more aggressively positive method of selecting by choosing the maximum of the scores from different models, we simply select the highest score as the final score;

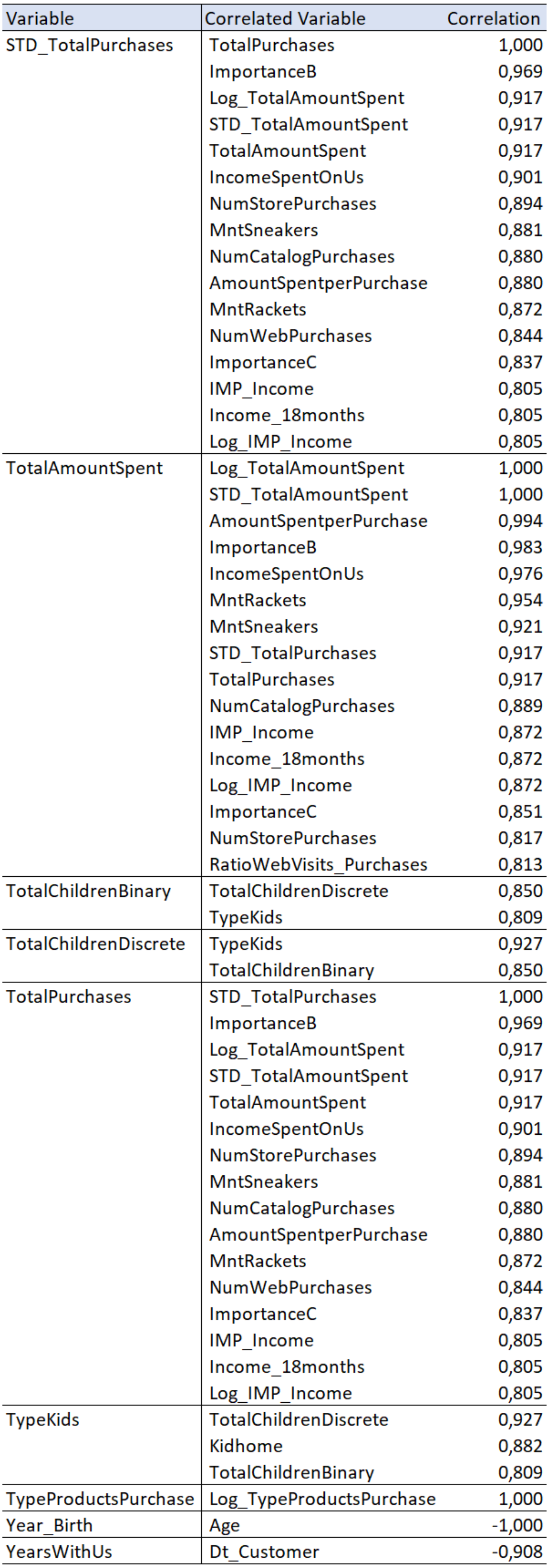
One that considers the score as a vote, a simple one, either a model votes 0 or 1 and we tally the votes. We then decide if it is a 0 or 1, and then we have to select the score, we either average the scores of the models who voted positive, or we use the proportion of models that voted 1;

The last model is a vote with a weight, the weight of each vote is actually the score of the observation, and they are averaged. The final vote is already a score, also normalized between 0 and 1.

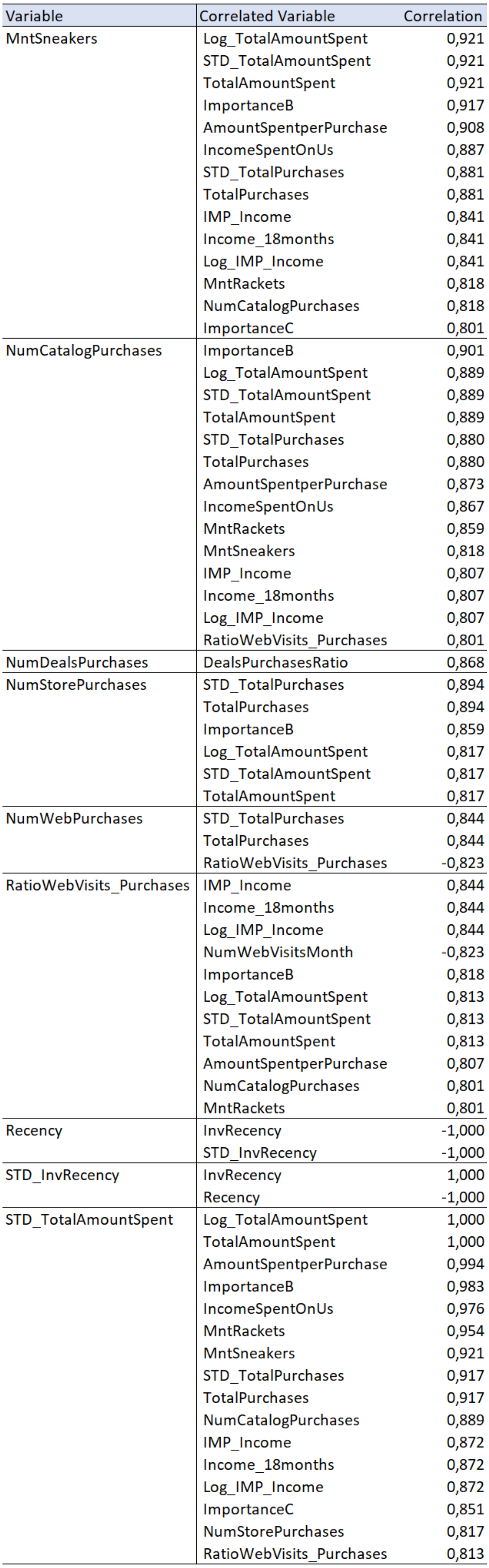
We can then describe the ensemble mode as a model that extract the best of each model, or the combined power of several models to be able to capture different nuances from the models.

Variable	Correlated Variable	Correlation
Age	Year_Birth	-1,000
AmountSpentperPurchase	Log_TotalAmountSpent	0,994
	STD_TotalAmountSpent	0,994
	TotalAmountSpent	0,994
	IncomeSpentOnUs	0,970
	ImportanceB	0,960
	MntRackets	0,952
	MntSneakers	0,908
	STD_TotalPurchases	0,880
	TotalPurchases	0,880
	NumCatalogPurchases	0,873
	IMP_Income	0,866
	Income_18months	0,866
	Log_IMP_Income	0,866
	ImportanceC	0,831
	RatioWebVisits_Purchases	0,807
DealsPurchasesRatio	NumDealsPurchases	0,868
Dt_Customer	YearsWithUs	-0,908
IMP_Income	Income_18months	1,000
	Log_IMP_Income	1,000
	Log_TotalAmountSpent	0,872
	STD_TotalAmountSpent	0,872
	TotalAmountSpent	0,872
	AmountSpentperPurchase	0,866
	ImportanceB	0,858
	RatioWebVisits_Purchases	0,844
	MntSneakers	0,841
	MntRackets	0,825
	NumCatalogPurchases	0,807
	STD_TotalPurchases	0,805
	TotalPurchases	0,805
ImportanceA	ImportanceC	0,908
ImportanceB	Log_TotalAmountSpent	0,983
	STD_TotalAmountSpent	0,983
	TotalAmountSpent	0,983
	STD_TotalPurchases	0,969
	TotalPurchases	0,969
	IncomeSpentOnUs	0,963
	AmountSpentperPurchase	0,960
	MntRackets	0,937
	MntSneakers	0,917
	NumCatalogPurchases	0,901
	ImportanceC	0,865
	NumStorePurchases	0,859
	IMP_Income	0,858
	Income_18months	0,858
	Log_IMP_Income	0,858
	RatioWebVisits_Purchases	0,818
Income_18months	IMP_Income	1,000
	Log_IMP_Income	1,000
	Log_TotalAmountSpent	0,872
	STD_TotalAmountSpent	0,872
	TotalAmountSpent	0,872
	AmountSpentperPurchase	0,866
	ImportanceB	0,858
	RatioWebVisits_Purchases	0,844
	MntSneakers	0,841
	MntRackets	0,825
	NumCatalogPurchases	0,807
	STD_TotalPurchases	0,805
	TotalPurchases	0,805
IncomeSpentOnUs	Log_TotalAmountSpent	0,976
	STD_TotalAmountSpent	0,976
	TotalAmountSpent	0,976
	AmountSpentperPurchase	0,970
	ImportanceB	0,963
	MntRackets	0,936
	STD_TotalPurchases	0,901
	TotalPurchases	0,901
	MntSneakers	0,887
	NumCatalogPurchases	0,867
	ImportanceC	0,838
InvRecency	STD_InvRecency	1,000
	Recency	1,000

Variable	Correlated Variable	Correlation
Kidhome	TypeKids	0,882
Log_IMP_Income	IMP_Income	1,000
	Income_18months	1,000
	Log_TotalAmountSpent	0,872
	STD_TotalAmountSpent	0,872
	TotalAmountSpent	0,872
	AmountSpentperPurchase	0,866
	ImportanceB	0,858
	RatioWebVisits_Purchases	0,844
	MntSneakers	0,841
	MntRackets	0,825
	NumCatalogPurchases	0,807
	STD_TotalPurchases	0,805
	TotalPurchases	0,805
Log_TotalAmountSpent	STD_TotalAmountSpent	1,000
	TotalAmountSpent	1,000
	AmountSpentperPurchase	0,994
	ImportanceB	0,983
	IncomeSpentOnUs	0,976
	MntRackets	0,954
	MntSneakers	0,921
	STD_TotalPurchases	0,917
	TotalPurchases	0,917
	NumCatalogPurchases	0,889
	IMP_Income	0,872
	Income_18months	0,872
	Log_IMP_Income	0,872
	ImportanceC	0,851
	NumStorePurchases	0,817
	RatioWebVisits_Purchases	0,813
Log_TypeProductsPurchase	TypeProductsPurchase	1,000
MntRackets	Log_TotalAmountSpent	0,954
	STD_TotalAmountSpent	0,954
	TotalAmountSpent	0,954
	AmountSpentperPurchase	0,952
	ImportanceB	0,937
	IncomeSpentOnUs	0,936
	STD_TotalPurchases	0,872
	TotalPurchases	0,872
	NumCatalogPurchases	0,859
	IMP_Income	0,825
	Income_18months	0,825
	Log_IMP_Income	0,825
	MntSneakers	0,818
	ImportanceC	0,810
	RatioWebVisits_Purchases	0,801



Variable	Correlated Variable	Correlation
STD_TotalPurchases	TotalPurchases	1,000
	ImportanceB	0,969
	Log_TotalAmountSpent	0,917
	STD_TotalAmountSpent	0,917
	TotalAmountSpent	0,917
	IncomeSpentOnUs	0,901
	NumStorePurchases	0,894
	MntSneakers	0,881
	NumCatalogPurchases	0,880
	AmountSpentperPurchase	0,880
	MntRackets	0,872
	NumWebPurchases	0,844
	ImportanceC	0,837
	IMP_Income	0,805
	Income_18months	0,805
	Log_IMP_Income	0,805
TotalAmountSpent	Log_TotalAmountSpent	1,000
	STD_TotalAmountSpent	1,000
	AmountSpentperPurchase	0,994
	ImportanceB	0,983
	IncomeSpentOnUs	0,976
	MntRackets	0,954
	MntSneakers	0,921
	STD_TotalPurchases	0,917
	TotalPurchases	0,917
	NumCatalogPurchases	0,889
	IMP_Income	0,872
	Income_18months	0,872
	Log_IMP_Income	0,872
	ImportanceC	0,851
	NumStorePurchases	0,817
	RatioWebVisits_Purchases	0,813
TotalChildrenBinary	TotalChildrenDiscrete	0,850
	TypeKids	0,809
TotalChildrenDiscrete	TypeKids	0,927
	TotalChildrenBinary	0,850
TotalPurchases	STD_TotalPurchases	1,000
	ImportanceB	0,969
	Log_TotalAmountSpent	0,917
	STD_TotalAmountSpent	0,917
	TotalAmountSpent	0,917
	IncomeSpentOnUs	0,901
	NumStorePurchases	0,894
	MntSneakers	0,881
	NumCatalogPurchases	0,880
	AmountSpentperPurchase	0,880
	MntRackets	0,872
	NumWebPurchases	0,844
	ImportanceC	0,837
	IMP_Income	0,805
	Income_18months	0,805
	Log_IMP_Income	0,805
TypeKids	TotalChildrenDiscrete	0,927
	Kidhome	0,882
	TotalChildrenBinary	0,809
TypeProductsPurchase	Log_TypeProductsPurchase	1,000
Year_Birth	Age	-1,000
YearsWithUs	Dt_Customer	-0,908



Variable	Correlated Variable	Correlation
MntSneakers	Log_TotalAmountSpent	0,921
	STD_TotalAmountSpent	0,921
	TotalAmountSpent	0,921
	ImportanceB	0,917
	AmountSpentperPurchase	0,908
	IncomeSpentOnUs	0,887
	STD_TotalPurchases	0,881
	TotalPurchases	0,881
	IMP_Income	0,841
	Income_18months	0,841
	Log_IMP_Income	0,841
	MntRackets	0,818
	NumCatalogPurchases	0,818
	ImportanceC	0,801
NumCatalogPurchases	ImportanceB	0,901
	Log_TotalAmountSpent	0,889
	STD_TotalAmountSpent	0,889
	TotalAmountSpent	0,889
	STD_TotalPurchases	0,880
	TotalPurchases	0,880
	AmountSpentperPurchase	0,873
	IncomeSpentOnUs	0,867
	MntRackets	0,859
	MntSneakers	0,818
	IMP_Income	0,807
	Income_18months	0,807
	Log_IMP_Income	0,807
	RatioWebVisits_Purchases	0,801
NumDealsPurchases	DealsPurchasesRatio	0,868
NumStorePurchases	STD_TotalPurchases	0,894
	TotalPurchases	0,894
	ImportanceB	0,859
	Log_TotalAmountSpent	0,817
	STD_TotalAmountSpent	0,817
	TotalAmountSpent	0,817
NumWebPurchases	STD_TotalPurchases	0,844
	TotalPurchases	0,844
	RatioWebVisits_Purchases	-0,823
RatioWebVisits_Purchases	IMP_Income	0,844
	Income_18months	0,844
	Log_IMP_Income	0,844
	NumWebVisitsMonth	-0,823
	ImportanceB	0,818
	Log_TotalAmountSpent	0,813
	STD_TotalAmountSpent	0,813
	TotalAmountSpent	0,813
	AmountSpentperPurchase	0,807
	NumCatalogPurchases	0,801
	MntRackets	0,801
Recency	InvRecency	-1,000
	STD_InvRecency	-1,000
STD_InvRecency	InvRecency	1,000
	Recency	-1,000
STD_TotalAmountSpent	Log_TotalAmountSpent	1,000
	TotalAmountSpent	1,000
	AmountSpentperPurchase	0,994
	ImportanceB	0,983
	IncomeSpentOnUs	0,976
	MntRackets	0,954
	MntSneakers	0,921
	STD_TotalPurchases	0,917
	TotalPurchases	0,917
	NumCatalogPurchases	0,889
	IMP_Income	0,872
	Income_18months	0,872
	Log_IMP_Income	0,872
	ImportanceC	0,851
	NumStorePurchases	0,817
	RatioWebVisits_Purchases	0,813