WarGames Store Project

Data Mining II - Final Report

Maria Inês Pastor Pereira da Silva, D20170088

3rd May 2018

Contents

1 Introduction

The WarGames company is a well-established store which specializes in war games and miniatures. Their offer includes card games, painting material, specialized magazines, miniatures and playing scenarios, which are sold through three main channels - physical stores, a quarterly catalogue and their website.

In order to increase efficiency and to guarantee good prospects of future revenue, the marketing team was asked to build a predictive model for direct marketing campaigns. Before officially launching the sixth campaign of the year, they sent it to a sample of 2500 customers and registered their response. To complement this dataset, they also gathered information about these customers and their shopping habits. The goal was to use this data to build a model that predicts how a customer will respond to this marketing campaign (i.e., a classifier), apply the model to the remaining customers and send the campaign to the customers more likely to purchase the campaign's offer. This way, instead of sending the campaign to the entire customer database, the marketing team can pick the subset of customers expected to result in the highest profit for the campaign and thus optimize their budget.

This report describes the process and methodological choices done to build a classifier for this sixth marketing campaign. I'll present and discuss the dataset, explain the transformations applied to the data, clarify how the classifier was selected and built and present the main results and conclusions. It is important to note that most analyses were done in SAS Enterprise Miner (SAS EM).

2 Data exploration and pre-processing

The original dataset contained 26 variables, the target variable DepVar, which is binary, 8 categorical input variables and 17 numeric input variables. It had personal information about 2500 customers, their shopping habits for the last 18 months and their response to the previous five marketing campaigns. Table 1 gives more details about the variables and their meaning.

Variable	Description	Type
AcceptedCmp1	Flag indicating customer accepted offer in campaign 1	Binary
AcceptedCmp2	Flag indicating customer accepted offer in campaign 2	Binary
AcceptedCmp3	Flag indicating customer accepted offer in campaign 3	Binary
AcceptedCmp4	Flag indicating customer accepted offer in campaign 4	Binary
AcceptedCmp5	Flag indicating customer accepted offer in campaign 5	Binary
Complain	Flag indicating if customer has complained in the last 18 months	Binary
DepVar	Flag indicating customer accepted offer in the current campaign	Binary
Dt_Customer	Date of customer's enrolment with the company	Date
Education	Customer's level of education	Nominal
Income	Yearly income of the customer's household	Interval
Kidhome	Number of kids in the customer's household	Interval
Marital_Status	Customer's marital status	Nominal
MntMiniatures	Amount spent on miniatures in the last 18 months	Interval
MntCard_Games	Amount spent on card games in the last 18 months	Interval
MntPainting_Material	Amount spent on painting material in the last 18 months	Interval
MntMagazines	Amount spent on magazines in the last 18 months	Interval
MntScenario	Amount spent on playing scenarios in the last 18 months	Interval
MntBrandA_Material	Amount spent on items from Brand A in the last 18 months	Interval
NumCatalogPurchases	Number of purchases made through the catalog in the last 18 months	Interval
NumStorePurchases	Number of purchases made through physical stores in the last 18 months	Interval
NumDealsPurchases	Number of purchases made with discounts in the last 18 months	Interval
NumWebPurchases	Number of purchases made through the catalog in the last 18 months	Interval
NumWebVisitsMonth	Average number vistis per month to the website from the last 18 months	Interval
Recency	Number of days since the last purchase	Interval
Teenhome	Number of teenagers in the customer's household	Interval
Year_Birth	Customer's year of birth	Interval

Tab. 1: Description of variables in the provided dataset

2.1 Summary statistics and plots

In order to better understand the dataset, I did a simple exploratory analysis. For the categorical variables, a bar plot shows the number of classes of each variable and the number of customers in each class. For the numeric variables, I computed some summary statistics and plotted histograms to visualized their distribution.

Before analysing the input variables, I took a look at the target variable, DepVar. It is a binary variable that takes the value 1 if a customer accepts the current campaign's offer and takes the value 0 if a customer does not accept the offer. It is an unbalanced variable since only 13.4% of the customers accepted the offer. Additionally, this variable does not have missing values.

Figure 1 displays the distribution of the input categorical variables. There are six binary variables and two nominal variables, both with five classes. All the binary variables are highly unbalanced, which is not surprising since the AcceptedCmp variables indicate whether a customer accepted the offer of previous campaigns and Complain indicates whether a customer has made a complaint in the last 18 months. Education and Marital_Status are slightly more balanced than the binary input variables. *Graduation* is the mode of Education and *Married* is the mode of Marital_Status.

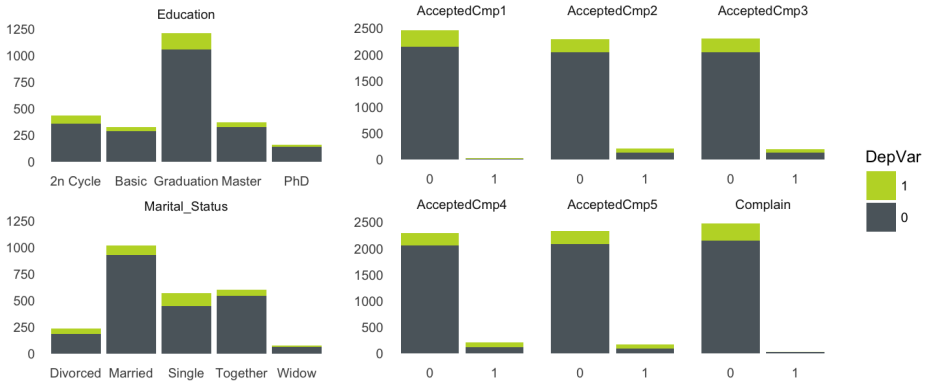


Fig. 1: Distribution of categorical input variables - total number of customers per class

Table 2 includes the mean, coefficient of variation, skewness, minimum and maximum of all the numeric input variables. It should be noted that, since Dt_Customer is an interval variable, some of the summary statistics are not defined for this variable.

The first observation I made was the coherence of the variables' extreme values and means, as there

are no variables with strange or impossible values. For instance, Year_Birth varies between 1944 and 1999, which means that the customers are aged between 19 and 74, the variables related with the number of purchases (the "Num" variables) in the last 18 months vary between 0 and 25 and the number of kids and teens in the household vary between 0 and 2, which is reasonable.

Secondly, the variables related to the purchase amounts (the "Mnt" variables) are the most disperse, which can be observed by their high coefficients of variation. Kidhome and Teenhome also have high coefficients, however, this is caused by their mean being very close to zero.

Thirdly, the "Mnt" variables and some of the "Num" variables show distributions with extremely pronounced skews to the right, a fact that will be very relevant in the identification of outliers. This right-skew can be observed in their high positive skewness and the shape of their histograms, which can be found in the annexe. In addition, a visual inspection of the histograms allows to conclude that the variables Income and Year_Birth have a distribution close to a Gaussian and the variables Dt_Customer and Recency have distribution more or less uniform.

Variable	Mean	Coef. Var.	Skewness	Min.	Max.
Income	63728,46	0,46	0,85	2216	196437
Dt_Customer	2015-07-21	n.a.	n.a	2014-07-30	2016-06-29
Kidhome	0,45	1,22	0,66	0	2
MntBrandA._Material	55,22	1,16	1,88	0	324
MntCard_Games	288,72	1,43	2,52	0	2628
MntMagazines	25,61	1,44	1,94	0	179
MntMiniatures	302,01	1,1	1,17	0	1494
MntPainting_Material	41,81	1,48	2,11	0	299
MntScenario	54,02	1,49	2,1	0	396
NumCatalogPurchases	4,93	0,71	2,28	0	25
NumDealsPurchases	2,43	0,96	2,93	0	16
NumStorePurchases	6,69	0,49	0,58	0	14
NumWebPurchases	7,91	0,36	0,3	0	15
NumWebVisitsMonth	5,19	0,52	0,89	0	20
Recency	49,11	0,6	0,03	0	99
Teenhome	0,48	1,12	0,46	0	2
Year_Birth	1970,98	0,01	-0,07	1944	1999

Tab. 2: Summary statistics of input interval variables

2.2 Outliers

The first step of pre-processing was to identify outliers in the numeric variables[1] and exclude them from the analysis. Outliers consist of unusual or atypical observations which are too far from the rest of the observations or are inconsistent with the process that generates the data. Because the classifiers I used are data-driven, outliers in the training set can potentially lead to unstable results and biased predictions. Therefore, the common practice is to remove them from the training dataset.

By looking at histograms of the numeric variables, it is clear that some features had outliers. For instance, there were a few observations with an Income so high that they formed a sort of cluster separated from the other observations. However, not all outliers can be so easily spotted and thus I used some well know criteria for identifying outliers them.

A frequent approach is to use Tukey's method[2], where thresholds based on quantiles identify candidates to outliers. This is the standard method of computing the whiskers in common box-plots. Although being a reliable method in general, for highly skewed distributions, Tukey's method may lead to incorrect results. Thus, in these cases, one can use an adaptation of this method that takes into account the variable's skewness[3]. If there is no skewness, then this method is equivalent to Tukey's.

[1] Does not make sense to talk about outliers for categorical variables

[2] McGill, R., Tukey, J. W. and Larsen, W. A. (1978) Variations of box plots. The American Statistician 32, 12-16

[3] Hubert, M. and Vandervieren, E. (2008). An adjusted boxplot for skewed distributions, Computational Statistics and Data Analysis 52, 5186–5201

Variable	Left threshold	Left outliers	Right threshold	Right outliers
Income	-23720	0%	145029	1.2%
MntBrandA_Material	-6	0%	369	0%
MntCard_Games	-21	0%	4079	0%
MntMagazines	-2	0%	361	0%
MntMiniatures	-120	0%	2865	0%
MntPainting_Material	-3	0%	560	0%
MntScenario	-4	0%	629	0%
NumCatalogPurchases	0	0%	27	0%
NumDealsPurchases	-2	0%	6	5%
NumStorePurchases	2	2.48%	29	0%
NumWebPurchases	3	2.48%	20	0%
NumWebVisitsMonth	-3	0%	13	0.76%
Recency	-44	0%	164	0%
Year_Birth	1913	0%	1999	0%

Tab. 3: Results of the modified Tukey's method for identifying outliers

Table 3 proposes some candidates to outliers. To confirm them, I looked at the distribution of the correspondent variables. Figure 3 displays the plots I used to make this confirmation and the final thresholds I defined to exclude the outliers. For almost all the variables, I plotted each against a second meaningful variable that supported the visualization.

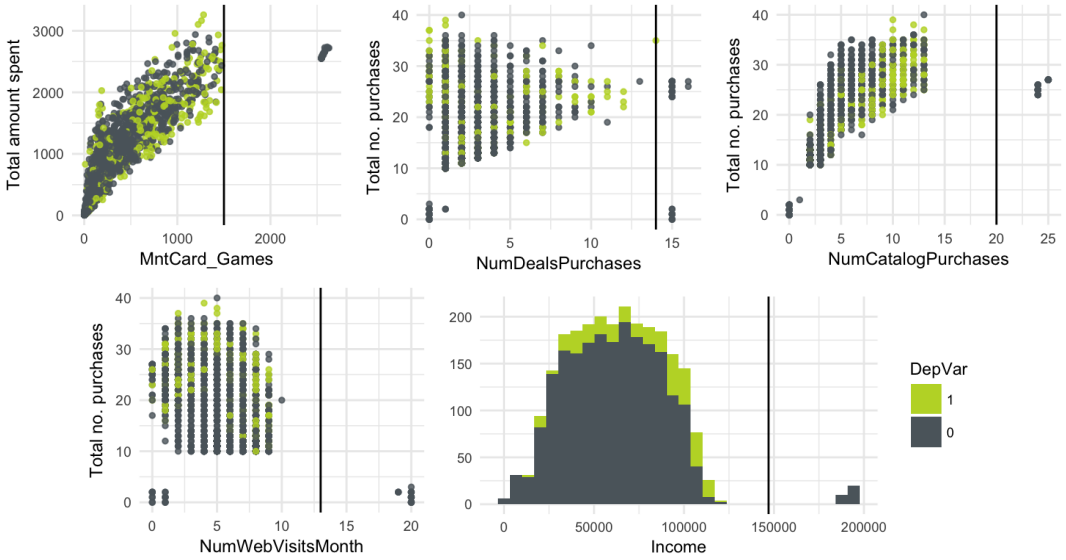


Fig. 3: Identification of outliers - variables' distribution and final thresholds

The thresholds for Income and NumWebVisitsMonth seemed appropriate as the outliers formed a clear group far from the rest of the observations. NumStorePurchases and NumWebPurchases also had a group on the left side that was separate from the rest of the data, however, I decided to do not consider them outliers as it seemed very plausible to have customers with few purchases in the last 18 months. Coincidently, this group of customers was excluded as outliers of other variables.

NumDealsPurchases had a right threshold that excluded too many observations and that was inconsistent with the distribution of the variable. However, this variable did have a tight group of observations away from rest and, therefore, I set a new threshold to exclude this group. Additionally, even though they had no outliers' candidates, MntCard_Games and NumCatalogPurchases also had groups of observations on the right side of the distribution that looked as outliers and, thus, I decided to exclude them as well.

After applying the threshold to these five variables, I excluded 62 customers as outliers and reduced the initial dataset to 2438 customers.

2.3 Missing values

There are 3 variables with missing values, namely Income, MntBrandA_Material and MntScenario. In total, 103 customers had missing values in at least one of these variables, which represents 4.12% of the initial dataset.

There are two main strategies to deal with missing values - remove observations/variables or input new values. Since the percentage of outliers is higher than 3%, I need to be careful with how I treat this data in order to avoid bias. Thus, removing observations or variables is not an optimal strategy in my opinion. In addition, since I plan to make transformations to some input variables, I need to guarantee that any imputation of missing values won't lead to inconsistencies, which immediately excludes the use of fixed values or central tendency measures.

Therefore, I decided to use the node from SAS EM called *Impute* to fill in all the missing values. In particular, I chose the *Tree* method which trains a decision tree for each input variable with missing values to predict that variable based on all the other input variables. More details on this SAS node and the imputation method can be found in the annexe.

2.4 Variable transformation

For this project, I applied two types of transformations to the input variables - I combined different variables to create new ones that were more meaningful to the problem and I transformed variables into a more suitable format for training the classifier. More specifically, the following transformations were implemented:

- Created the new variable campaign_acceptance, which is the total number of previous campaigns accepted.
- Created the new binary variable Childhome, indicating whether or not a customer has a kid or a teen in the household.
- Created the new variable Mnt, which is the total amount spent by a customer in the last 18 months.
- Created the new variable Frq, which is the total number of purchases made by the customer in the last 18 months.
- Created the new variable average_purchase, which is the average amount spent per purchase, computed based on purchases from the last 18 months.
- Year_Birth was replaced by age. This variable represents the age of a customer, assuming that the current year is 2018.
- Dt_Customer was replaced by loyalty. This variable represents the number of months since a customer joined the store.
- The "Mnt" variables were all divided by Mnt and thus began to represent the percentage of spending in the corresponding type of items.
- The "Num Purchases" variables were all divided by Frq and thus began to represent the percentage of purchases in the corresponding channel.

After making these transformations and exploring the new variables, I found some inconsistencies. Particularly, one customer had a value higher than 100% in MntBrandA_Material and 12 customers reported fewer website visits than purchases through the website, which is impossible. For the first, since there was only one customer, I decided to simply exclude him from the analysis. As for the second, I supposed that the number of purchases was more reliable and thus changed the number of web visits to equal the number of purchase through the website.

Finally, in order to avoid the presence of different scales in the dataset, which can be problematic for some classifiers, I applied the min-max normalization to all the numeric variables that were not percentages.

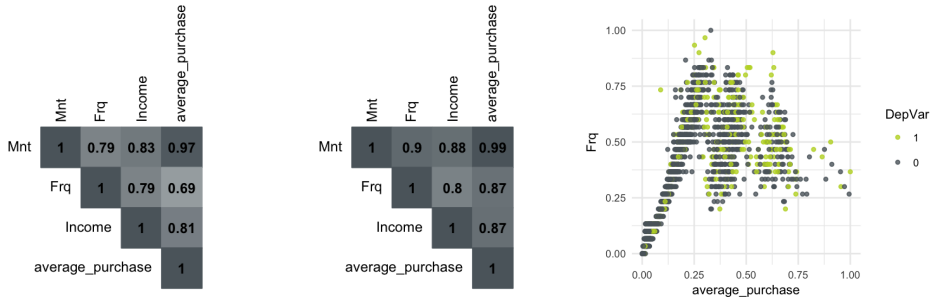
3 Variable selection

After the variable transformation step, I had 4 categorical variables and 19 numeric variables as inputs to build a classifier. Clearly, it was too much for the problem at hand. Therefore, I had to select the most meaningful variables to predict DepVar.

3.1 Redundancy

An important step in variable selection is to avoid redundancy. In other words, we should only consider variables that bring new information to the model. One can spot redundant variables by computing their correlation. If two variables are highly correlated (either positively or negatively) then we should only use one to build the predictive model.

For this project, I used two widely known measures of correlation. The Pearson correlation coefficient is a measure of linear correlations and thus is a good indicator of linear relationships between two variables. On the other hand, the Spearman's rank correlation coefficient is a measure of rank correlation and thus can be used to assess whether the relationship between two variables is described by a monotonic function. One can say that Spearman's correlation is broader than Pearson's as it considers non-linear relationships between variables.



(a) Pearson correlation (b) Spearman's correlation (c) Scatter plot

Fig. 5: Correlation plots for highly correlated variables and relationship between Frq and av_purchase

Figure 5 shows the results of this analysis for the highly correlated variables, namely, the pairs of variables with an absolute correlation higher or equal to 0.8.

Income, Mnt and av_purchase formed a group of highly correlated variables, which is evident by both their Pearson and Spearman's correlation coefficients. Therefore, I had to choose only one of them to train the classifier. As for the variable Frq, even though it shows some high correlation with these three variables, their relationship seems to have a quite curious shape. As I believe it seems to add some interesting information to the dataset, I decided to keep it in the analysis.

3.2 Relevancy

Relevancy is related with the ability of a variable to differentiate or predict the target variable. As an example, if the distribution of the target variable through a specific input variable is uniform, then this input variable is irrelevant as it has no discriminative power over the target variable.

There are many ways of measuring relevancy, however, in this project, I focussed on the *Variable Worth*. This is a measure provided by the SAS EM's node *StatExplore* that allows a ranking of all variables from the most relevant to the least relevant. Its computation is based on the *reduction in impurity* (which is based on the Gini index) produced by the best split of the target variable based on single input variables. Figure 7 presents the variable worth of the 23 input variables. Since the project penalizes the use of more than 10 variables, I chose the 10 variables with the highest variable worth, excluding Income and Mnt to avoid redundancy.

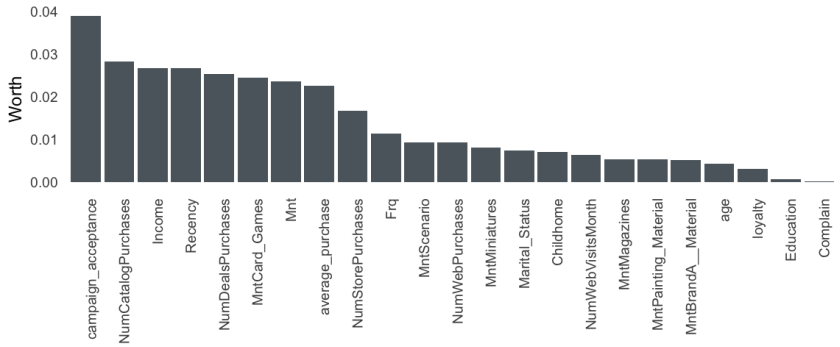


Fig. 7: Variable worth of input variables

SAS EM also has a very interesting node called *Variable Selection*. This node has some pre-defined routines to automatically select the best variables based on either R-square or Chi-square selection criteria. More details about this node can be found in the annexe. Although having a small variable worth, Marital_Status was selected by both R-square and Chi-square criteria. Therefore, I decided to add this variable to the training dataset and remove MntScenario.

3.3 Training dataset

The final variables used to train the predicted model are displayed in figure 8. I plotted the histograms of the first eight variables and the bar plots of the remaining two variables.

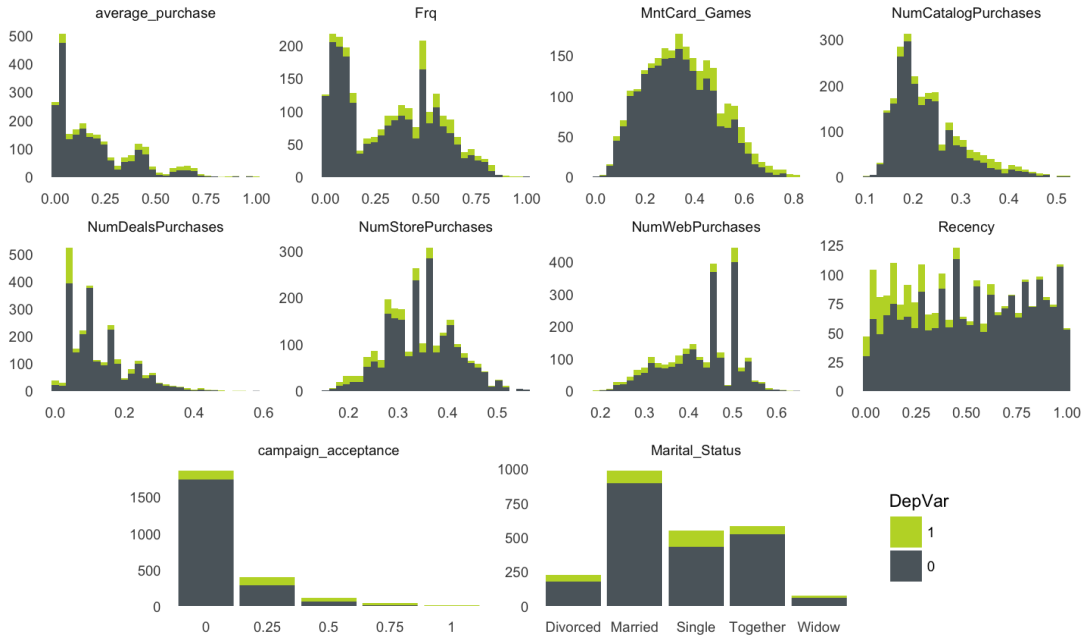


Fig. 8: Distribution of final input variables

4 Training and validation

In this phase of the project, I tested a few classifiers with a 10-fold cross-validation. This is a widely used learning schema that has the advantage of using all observations exactly once for validating the model. In this process, the dataset is randomly split into 10 equally-sized subsets and, for each subset, the remaining 9 subsets are used to train the model while the retained subset is used for validation. In other words, there's a loop where each subset is used exactly once as the validation set.

I chose to use this learning schema because, after a few training runs with the classic hold-out method (70% for training and 30% for validation), I realised that the classifiers showed performances so close that changing the dataset's partition would have an impact on the choice of the best model. Thus, by using cross-validation, I aimed at having a more robust estimate of the true best model for the problem.

4.1 Classifiers

During cross-validation, I tested three types of classifiers - logistic regressions, decision trees and artificial neural networks. During the initial training runs, I also tested the k-nearest neighbours (which can be implemented in SAS EM with the MBR node) and an ensemble of all the trained models (which can be implemented in SAS EM with the Ensemble node). However, since they were having much lower performances than the logistic regressions and the neural networks, I decided to exclude them from the cross-validation process.

Logistic regression

The logistic regression is a classifier rooted in classical statistics where the class probability is modelled as the inverse-logit of a linear combination of the input variables: $P(Y = 1|X) = \frac{e^{X\beta+\epsilon}}{1+e^{X\beta+\epsilon}}$.

In this model, the function that relates the probability of the target class to the linear combination of the input variables is called the link function. There are some implementations of the logistic regression using different link functions, such as the probit model. However, for this project, I used the default inverse-logit.

To train this model, the vector of weights β is estimated using maximum likelihood. This approach has the advantage of allowing to compute statistical significance tests to measure goodness-of-fit. SAS EM allows choosing a method for selecting the variables included in the regression. I used the *Stepwise* selection method, which iteratively adds variables that are significantly associated with the target variable and, after adding it, checks the significance of all the variables included in previous steps.

SAS EM also has the option of using two-factor interactions between input variables and polynomial terms. Thus, I included three logistic regression classifiers in the cross-validation test:

- Simple regression - logistic regression that only considers single variables as effects.
- 1-order regression - logistic regression that includes two-factor interactions as effects.
- 2-order regression - logistic regression that includes two-factor interactions and polynomial terms of order two as effects.

Decision trees

In predictive analytics, decision trees are collections of simple rules which are usually represented by a tree-like graph. The main advantages of this family of models are their interpretability and their robustness to different variable types and scales. There are many different algorithms to train decision trees, but the underlying idea is always to define a series of successive rules on individual input variables that split the dataset into groups as "pure" as possible in relation to the target variable.

The *Decision Tree* node from SAS EM has quite a bit of flexibility for training decision trees as it allows to define the maximum number of branches in each tree node, the maximum generations of nodes (the depth of the tree) and the criterion for choosing the optimal split (p-value of the Pearson Chi-square statistic, information gain or reduction in the Gini index). For this project, I used the default splitting criterion for nominal targets - the Pearson Chi-square statistic with a maximum p-value of 0.2 - and included three decision tree classifiers in the cross-validation test:

- Decision Tree Small - Decision tree with a maximum of 2 branches per tree node and a maximum depth of 3 generations.
- Decision Tree Medium - Decision tree with a maximum of 3 branches per tree node and a maximum depth of 5 generations.

- Decision Tree Big - Decision tree with a maximum of 5 branches per tree node and a maximum depth of 6 generations.

Artificial neural networks

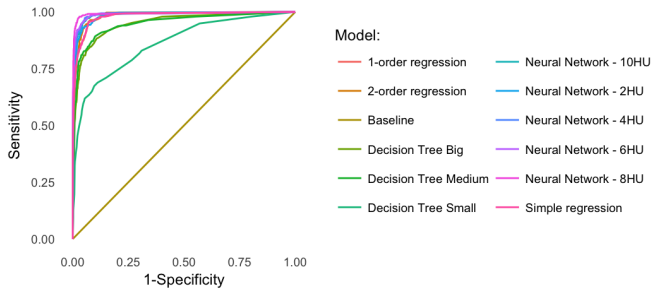
Artificial neural networks are systems inspired by the way biological brains function. Similarly to a brain, these systems are a grid of connected units (or artificial neurons) that transmit and process data. They are usually called "universal approximators" since they can approximate complex functions within any desired precision. Therefore, they are a powerful tool for modelling complex patterns and solving problems where there is not much information about the underlying phenomenon.

SAS EM has a node called *Neural Networks* where one can train artificial neural networks with a single hidden layer. For this project, I used all the default settings and chose a set of different hidden units (i.e., number of units in the hidden layer) to test different neural networks. In other words, I included five neural network classifiers in the cross-validation test, with 2, 4, 6, 8 and 10 hidden units respectively.

4.2 Cross-validation results and conclusion

Figure 9 presents the cross-validation performance of the 11 tested models. In particular, on the left, there're the ROC curves computed with the cross-validation scores and, on the right, there're the ROC indexes and the error rates, also computed with cross-validation scores.

From these results, we can conclude that there are two groups of models. The neural networks and the logistic regressions are the most performing and the decision trees are the least performing. This seems to indicate that the decision boundary of this problem cannot be well modelled with linear planes perpendicular to the axis. In addition, the best models have extremely low error rates and high ROC indexes, which can be only attributed to the artificial nature of the dataset.



(a) ROC curves

Model	Roc Index	Error Rate
Neural Network - 8HU	0.995	0.02
Neural Network - 6HU	0.992	0.02
Neural Network - 4HU	0.992	0.03
Neural Network - 10HU	0.989	0.04
Neural Network - 2HU	0.987	0.04
2-order regression	0.986	0.04
1-order regression	0.983	0.04
Simple regression	0.983	0.05
Decision Tree Big	0.953	0.05
Decision Tree Medium	0.95	0.06
Decision Tree Small	0.865	0.09

(b) ROC indexes and error rates

Fig. 9: Models' performance in cross-validation

Since the aim of this project is to optimize the budget of direct marketing campaigns, in the end, the most important performance indicator is the expected profit of the campaign. The problem statement indicates that, for this campaign, one contact to a customer costs 4 euros and a positive answer from a customer (i.e., the customer accepts the campaign's offer) leads to a revenue of 20 euros. With this information, we can then extrapolate the expected profit of contacting specific groups of customers.

One of the main outputs of this project is to indicate, from a list of 5000 customers, who should be contacted in order to maximize the campaign's profit. Based on the response rates (i.e. the percentage of customers accepting the offer) obtained in the cross-validation process, I estimated the response rate and the expected profit for a group of 5000 customers using a SAS code node presented in the practical classes. More specifically, the customers were ordered based on the predicted probability of accepting the offer and, for specific percentiles, the response rate and profit of contacting all customers in or above that percentile were estimated. This resulted in two curves, one for the expected profit and one for the response rate, which are displayed in Figure 10.

Consistently, the best performing models show higher profit and response curves and once again these models resulted in very similar profit curves. However, the highest profit was obtained with the neural

network with 8 hidden layers and by contacting the 750 most promising customers. In this case, the estimated profit was 10080 euros and the response rate was 87.1%.

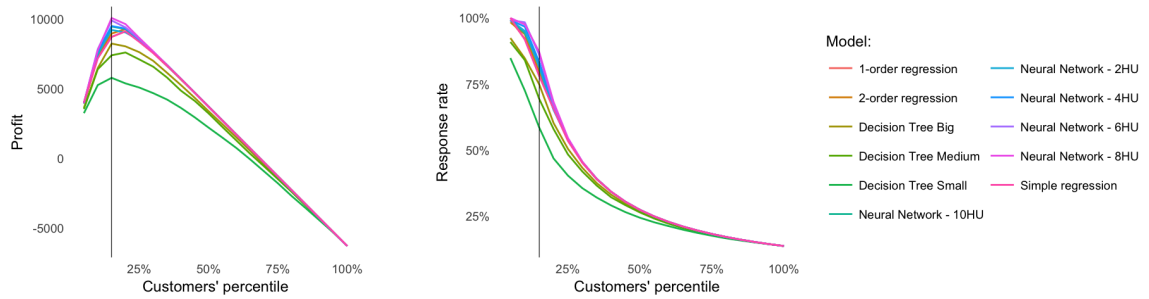


Fig. 10: Profit curves and response rate curves

In conclusion, because this is the best performing model according to all the considered indicators, the neural network with 8 hidden layers is the selected model to predict the response to this marketing campaign. Additionally, the analysis of the profit curve indicates that contacting the 15% of customers with the highest predicted probability of accepting the offer will result in the highest profit for the campaign. This means that, from the original 5000 customers, the marketing team should send the campaign to the 750 most promising customers.

In order to find these 750 customers, I trained a new neural network with 8 hidden layers using the classic holdout method (70% for training and 30% for validation) and used the SAS EM node called *Score* to estimate their probability of accepting the offer.

5 Annexe

SAS EM - Impute node

The *Impute* node is SAS EM standard for replacing missing values as it has many pre-defined methods, such as, the *Default Constant*, the *Mean* or the *Median*.

Given the specificities of the project, I decided to use the *Tree* imputation method with the default settings. According to the Reference Help, missing values are replaced by estimates based on a decision tree which is trained on all the other input variables. It is very important to note that the target variable is excluded from the training set of this decision tree. In fact, it would not make sense to use the target variable to impute missing values as it would lead to bias for the main model, i.e. the model to predict the target variable.

SAS EM - Variable Selection node

As previously explained, this SAS EM node allows to identify the most relevant input variables for predicting the target and perform variable selection in a automatic way. Thus, this node has two main functionalities:

- Some user-defined criteria to exclude variables that are considered to have poor quality. For instance, we can define a maximum percentage of missing values allowed in the training dataset and if a variable exceeds this threshold, then it would be rejected.
- Some pre-defined selection criteria methods that test the relevancy of the input variables. The node has four available options:
  - R-Square criteria: this criteria uses a least squares regression that maximizes the model's coefficient of determination (also called R-square). This coefficient is the standard for measuring goodness-of-fit in linear regressions as it measures the proportion of the target variable's variance which is explained by the model. To choose the variables, this criteria starts with the intercept and then adds the variable with the highest coefficient of determination. Then, it

successively adds the variable that most contributes to an increase in the model's coefficient of determination. This iteration stops when all the variables are included or a stop criterion is met. In the end, the users sees the variables that most contributed to the model's coefficient of determination and therefore are the most relevant for predicting the target variable with a linear model.

– Chi-Square criteria: this criteria can only be used with categorical target variables and is based on the Chi-Square test of independence, which is a highly used statistical test for measuring the association between two categorical variables. This classical test uses cross-frequency tables to describe the distributions of two categorical variables and assesses the degree of association by comparing the actual observations in the frequency table to what was expected, provided that the variables were independent. Thus, SAS EM creates binary splits in each input variable and builds a 2×2 frequency table to measure association of each variables with the target variable.
– R and Chi Square criteria: this method applies both criteria if the target variable is categorical and applies the R-square criteria if the target variable is numeric.
– None: no selection criteria is applied in this setting.

Given that I had already dealt with data issues in the dataset (such as missing values and outliers), I only used this node for the testing the relevancy of the input variables. Particularly, I used the R-Square and the Chi-square criteria separately.

Distribution of numerical input variables - initial dataset