Nova School of Business and Economics

Case 02

Author:
Benjamin C. Herbert (45775)
Carolina Domingues (31951)
David Neves (31927)
Rian Yan (28823)
Sebastian Rupp (46093)
Thomas Dornigg (41727)

Instructor:
Prof. Carlos Daniel SANTOS

An assignment submitted for the course
Big Data Analysis

Second Semester, T4
2021

May 14, 2021

1. Based on the results of the first assignment, we used the pre-processed dataset, *i.e.* data with removed outliers and pre-selected features, to model a Decision Tree and several other supervised machine learning classifiers.

Before building a model, the group defined the main research question to be addressed. As described by the Badgett & Stone (2005), the goal of Big Data Analytics is to achieve the broadest knowledge possible out of the information given about clients. That being said, the group concluded that the broadest value for the company can be generated by creating micro-segments [2] out of the customer base, predicting the response to each of the firm's offers.

At first instance, the group encoded the variables in a manner so that one model could predict different segments within the different responses and response combinations, which ultimately led to a outcome with 71.58% of non-responses (encoded as "0") and 28.42% of responses (encoded as "1"). However, due to lack of data and miserable prediction outcomes, we needed to change this approach. Resulting from this misleading encoding, we decided to fit three models instead, where each must be able to segment our customers into negative and positive response for each of the given response variables (making it a binary classification), answering the question: "Which customers are likely to respond to each of the company's product offers?". Generally speaking, the group considers the segmentation and model as strong, if it can predict a positive response of a customer.

Even though the result of the final encoding led to a binary target class - which simplifies the modeling procedure in the next steps -, the encoding for the *response_01*-prediction (*response_02*; *response_03*) model led to a class-imbalance with 92.07% (86.9%; 89.5%) for non-responses (encoded as "0") and 7.92% (13.1%; 10.5%) for responses (encoded as "1"). In this case, the model's ability to accurately predict the minority class's label or probability correctly is our goal. However, when working with an imbalanced classification problem, there are fewer examples of the minority class in the dataset. Most classification algorithms though are designed and demonstrated on problems that assume an equal distribution of classes. Therefore, it is more challenging for a model to learn the specific characteristics of examples from the minority class, and to differentiate instances from the majority class. If not handled in a meaningful way, one might face the issue that the classifier tends to predict the majority class without much problem but fails to predict the minority class [3]. Given this insight, the group will propose a bias-mitigation strategy towards this issue in a later part of this paper.

In the following steps, the group focused on implementing supervised machine learning classifiers both in SPSS and Python to get a proper comparison of the existing model performance. As given in the task, the main focus lies on Decision Trees which are one of the most popular algorithms for machine learning, given their straightforward interpretability power. It is called a Decision Tree because it starts the partitioning in a root node, expanding on further branches and internal nodes, constructing a tree-like structure [4]. They belong to the family of supervised learning algorithms that, unlike other supervised classifiers like Naive Bayes, can be used for both classification and regression problems. Since the main goal of our project is to define customers' features within a segment and exogenously delineate them, the group uses standard supervised classification-techniques to predict what kind of product our customers will respond to.

Before applying any model, selecting the right metric is the most critical step in any machine learning project. The metric will be the benchmark by which all models are evaluated and compared. The choice of the wrong metric can mean choosing a bad algorithm that fails to solve the problem initially defined. Also, the selected metric must capture the most essential predictions to the project or project stakeholders.

In order to choose the right classification metric for our business problem, one needs to know how to interpret a confusion matrix and its terminology. Generally, a confusion matrix is a table used to measure the performance of classification models. It contains columns with the actual classification and rows representing the predictions (or vice-versa):

- Recall or Sensitivity or TPR (True Positive Rate): Number of items correctly identified as positive out of total true positives
- Specificity or TNR (True Negative Rate): Number of items correctly identified as negative out of total negatives
- False Positive Rate or Type I Error: Number of items wrongly identified as positive out of total true negatives
- False Negative Rate or Type II Error: Number of items wrongly identified as negative out of total true positives

Arguably, the target for this business problem is to optimize the recall-metric and, most importantly, minimize the false negative rate since ultimately, the company will be more negatively affected by not targeting potential future customers (FNR) than sending out targeted marketing campaigns towards the ones not responding to a product offer (FPR). Thus, the goal is to have a model that supports the most accurate profile of the customers that are likely to accept the product offers, minimizing the percentage of “missed customers” thereby minimizing potential missed profits.

Our model evaluation is therefore based on minimizing the false negative rate (optimization of recall). To choose the tree with the best predictive power, the group used 10-fold cross validation (both for SPSS and Python). Empirically, setting k equal to 10 has shown to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance [5]. Apart from cross validation, the group was also concerned about the parameters which are used to fit the decision trees in the first place. For the so-called hyperparameter-tuning, we used different specifications as given by the *scikit-learn* library: For the maximum depth (1 to 12), the minimum number of samples required to be at a leaf node (1 to 12) and the criteria to measure the quality of the split (Gini or Entropy). Selecting these parameters has two reasons: We want to build a model that has high predictive power as well as a model which does not overfit. Drawing on a too strong model in the training set can lead to a misspecification in the test set due to the inclusion of too many non-relevant noise parameters.

For comparison purposes, the group decided to train also state-of-the art Machine Learning models in the course of this assignment (Random Forest, Logistic Regression, CatBoost and

Light Gradient Boosting Model). Likewise, for all these other models, a similar approach was chosen. The parameters to be specified for each of the specific models like CatBoost or Logistic Regression differ from those of the Decision Tree. The approach of minimizing the false negative rate, however, remains the same.

In SPSS, the analysis was conducted using CRT (Classification and Regression Tree) as growing method. The CRT technique starts by examining the input fields and finding the best split, measured by the reduction in an impurity index that results from that split. In CRT all splits are binary, and the method is applied recursively. CRT allows to select cross-validation to assess the goodness of the fit more accurately. Moreover, CRT trees give the option to first grow the tree and then prune based on a cost-complexity algorithm that adjusts the risk estimate based on the number of terminal nodes. This method is well suited for datasets with large numbers of fields as they typically do not require long training times to estimate.

For this analysis, a decision tree was computed for each of the three target variables: *response_01*, (*response_02* and *response_03*). As output, each simulation gives a model summary (indicating variables included and number of nodes), a classification table, an associated risk factor table and a decision tree (in tree and table format). The independent variables selected are the ones conceptually chosen as in the first assignment and which were approved in the data validation process to guarantee they can be operationalized from a practical perspective. However, SPSS will still only select a subset of the included variables for the tree.

In terms of validation, a 10-fold cross-validation was selected, the maximum tree-depth was kept at 5 (the standard for CRT) and the minimum number of cases was kept at default levels to avoid complexity (100 for parent node and 50 for child node). The impurity measure applied was Gini and pruning was not employed. To deal with the highly imbalanced dataset and to avoid having only “zero” predictions, the misclassification costs were set as to penalize the false negatives occurrences for the reasons stated previously.

To find a reasonable weighting scheme, we started by calculating the ratio between the bigger and the smaller category (number of “No” observations divided by number of “Yes” observations) and inputting that number as the misclassification cost. Sometimes to avoid the reversal of the problem (having only “Yes” predictions), that penalization had to be reduced. Then, by trial and error, the group attempted to reach a fair trade-off between a good rate of false negatives (and value of overall percentage of correct prediction) and the cross-validation risk.

However, as the software significantly lacks “Yes” data points, the misclassification costs have to be quite severe, creating inflated cross-validation risk estimates which make the interpretation less straightforward and the benefits of the trade-off hard to assess. On top of that, we ended up with a high percentage of false negatives, despite the efforts.

An overview of all model results, both from SPSS and Python can be seen in Table 1 below.

Product	Model	TNR	FPR	FNR	TPR	# TP	Recall	Risk CV
1	DT (SPSS)	70.3%	21.75%	4.2%	3.75%	32	47.3%	0.67
	DT (Python)	46%	46%	3%	5%	48	62.3%	
	RFC	73%	19%	5%	3%	26	33.8%	
	LR	58%	34%	4%	4%	37	48.1%	
	CatBoost	88%	4%	7%	1%	7	9.1%	
	LGBM	91%	1%	8%	0%	1	1.2%	
2	DT (SPSS)	78.4%	8.25%	11.23%	2.13%	21	15.4%	0.55
	DT (Python)	52%	33%	8%	5%	48	37.5%	
	RFC	75%	11%	11%	2%	20	15.6%	
	LR	47%	39%	8%	6%	56	43.8%	
	CatBoost	79%	8%	12%	2%	16	12.5%	
	LGBM	86%	0%	13%	0%	0	0.0%	
3	DT (SPSS)	71.12%	18.55%	6.33%	4%	40	40%	0.55
	DT (Python)	33%	56%	3%	8%	73	75.3%	
	RFC	72%	18%	7%	3%	30	30.9%	
	LR	53%	37%	5%	5%	50	51.5%	
	CatBoost	84%	5%	9%	1%	11	11.3%	
	LGBM	90%	0%	10%	0%	1	1.0%	

Table 1: Classification results (of the test-set)

---

2. In order to perform the micro-segmentation of our customers with the help of decision trees modeling, we decided to train three models to predict the following target variables:

- Target 1 - predict which clients are likely to buy product 1
- Target 2 - predict which clients are likely to buy product 2
- Target 3 - predict which clients are likely to buy product 3

Considering the resulting confusion matrices of the models for the three different targets (see Table 1), one can see no significant differences in the performance. Exemplary for the first response/product, the best recall out of all trained models had the Decision Tree with around 62.3%. It can be interpreted in the following way: our decision tree manages to predict 62.3% of positive responses cases out of all true positive responses. Additionally, it should be noted that the recall score of the simple Decision Tree has been better than the recall score of the Random Forest, which is a more sophisticated algorithm. Further more complex tree-based classifiers such as an LGBM classifier - which is a gradient boosting framework that uses tree-based learning algorithms -, and a CatBoost algorithm for classification problems, have been trained on the data. While the LGBM achieved a recall rate of merely 1.2%, the CatBoost did only slightly better by yielding a rate of 9.1% when predicting the response rate for product 1. As shown, the recall scores of all the tree-based models have not been good enough to be deployed yet, because none of them is capturing enough of the true positives. When looking at the ROC and AUC graph (see Figure 1), one can observe a similar result. However, the Random Forest for predicting clients who might be interested in the product has the highest ROC of 0.63. The AUC generally displays how well an algorithm is able to distinguish negative from positive instances. An AUC of 0.50 therefore means that the algorithm allocates classes completely at random and a value of 1 means that the algorithm is distinguishing both classes perfectly. However, as one can observe the algorithms we trained ranged all between 0.63 AUC in the best case to 0.49 in the worst case.

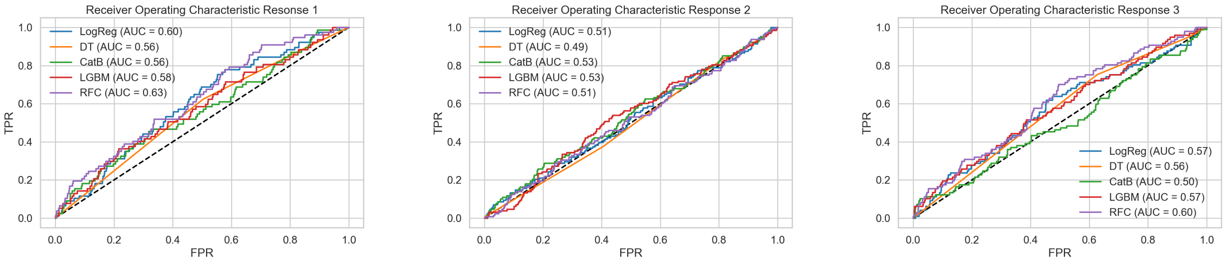

FIGURE 1: ROC-curves for fitted models

The reasons for the poor AUC and recall rates are that considering the approach of predicting the customer's response per product, the dataset has merely around 4000 instances. For example, for product 1, only 307 were interested in it, which shows the massive imbalance of the dataset. For setting off this effect, the synthetic minority oversampling technique has been applied right from the beginning. SMOTE is an approach, which creates synthetic observations based upon the existing minority observations. Despite using the library in Python, the group was not able to even out the class-imbalance to achieve better results.

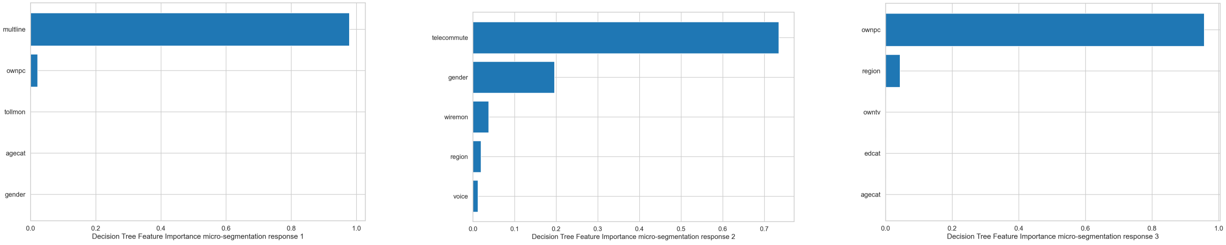

FIGURE 2: Feature Importance

The graphs above display the feature importance which results from the Decision Tree modeling. As shown in Figure 2, the client segments who would respond to the *offering 1* can be distinguished by whether they have multiple lines ("multiline") and their respective level of education ("edcat"). Thereby multiple lines have the highest impact on the segmentation in this case. Moreover, clients interested in *offering 2* can be segmented primarily by whether they telecommuted last month, by their gender, by whether they called the previous month wirelessly, the region they are from, and lastly, whether they own a voice mail or not. For *offering 3*, the Decision Tree segments customers by whether they own a personal computer or not and by their region.

Taking a closer look at the Decision Tree predicting *offering 1*, one can observe that clients were divided into two groups. One group did not have multiple lines and was more likely to subscribe to product 1, and the other did have multiple lines. However, in the left node, which predicts class 1, the entropy has been 0.948, which shows high impurity. Theoretically, an entropy of 1 means that both classes we are trying to predict are present equally. That's one of the strengths of a decision tree. We can now divide our customers into different micro segments, and know which ones to offer each product, expecting a positive response.

---

3. Realistically, in a potential meeting with the company's CEO, these shortcomings of the models would have to be addressed, and suggestions for an improvement be made. Firstly, one would need to roughly introduce the concept of a Decision Tree model to the CEO while providing a simple description of how the model is built and how it works. In this context, a Decision Tree could be explained as a model that takes in an individual customer and predicts how he or she has to be classified based on the customer's characteristics. By applying all three models, they predict whether the customer is supposed to be targeted and with which product offer. It is worth mentioning that the algorithm is trained according to the instances and their respective features it was exposed to initially. Therefore, the model is equipped to identify characteristics associated with target customers and non-target customers. Secondly, one should point out the models' performance. To do that, the AUC would be too complex to explain. Hence, the recall would be the metric used to elaborate on the model's performance to a CEO due to its greater simplicity.

Arguably, the results could be better which means the CEO would need to explained the models' performance. It has to be explained that the data set with which the models were trained had not enough examples of clients who bought certain products and that the database could be enhanced by adding features with more predictive power. Ultimately, the CEO should not base the companies final decisions only on the statistical models.

Taking this into consideration and assisting the CEO in making a specific decision regarding a future marketing plan to increase response rates, one could suggest either of the following courses of action:

1. Improve the models' performance, which could be achieved by collecting data about more customers and increasing the number of features. As mentioned before, a more balanced data set with a higher presence of customers who bought products in combination with more meaningful features could improve the predictive abilities.
2. Resort to other managerial tools for market research that may help the company understand consumer behavior and target market segmentation according to market research. For example, massive post-purchasing surveys through e-mail with a 3%-5% response rate can be cost-effective. Focus groups can be used to get a more robust and detailed result on understanding consumer behavior. Additionally, focus groups might indicate which further features of customers should be integrated into the model. Observations and field trials should also be used as a more effective market research model talking with the consumers face to face to understand consumer society more profoundly.

References

[1] Melody Badgett and Merlin Stone. 'Multidimensional segmentation at work: Driving an operational model that integrates customer segmentation with customer management'. In: *Journal of Targeting, Measurement and Analysis for Marketing* 13.1 (2005), pp. 103–121.

[2] Yoon Ho Cho, Jae Kyeong Kim, and Soung Hie Kim. 'A personalized recommender system based on web usage mining and decision tree induction'. In: *Expert systems with Applications* 23.3 (2002), pp. 329–342.

[3] Jason Brownlee. *A Gentle Introduction to Imbalanced Classification*. Dec. 2019. URL: https://machinelearningmastery.com/what-is-imbalanced-classification/. (accessed: 12.05.2021).

[4] Marina Milanović and Milan Stamenković. 'CHAID decision tree: Methodological frame and application'. In: *Economic Themes* 54.4 (2016), pp. 563–586.

[5] Jason Brownlee. *A Gentle Introduction to k-fold Cross-Validation*. May 2018. URL: https://machinelearningmastery.com/k-fold-cross-validation/. (accessed: 12.05.2021).