Technical Report – Direct Marketing Optimization

Table of Contents

Technical Report – Direct Marketing Optimization 1
2.1 Scope and problem definition 1
2.2 Methodology 2
2.3 EDA 2
2.4 Feature engineering 2
2.5 Modeling 3
2.6 Prediction 4
2.7 Limitations and next steps 5

2.1 Scope and problem definition

The purpose of this report is to provide a description of the steps undertaken in solving a direct marketing optimization problem on a sample of 1615 KBC customers, integrating the attached notebook (which provides a greater level of detail). This report will utilize the same variable names as the attached notebook for ease of reference. The notebook also contains notes and comments to guide the reader and make the code as interpretable as possible – to which this report functions as complement and explanation. For ease of reading, the shared notebook does not contain some steps (e.g. some histograms of the EDA, most grid searches, some algorithms tried and discarded).

Problem definition: identifying (flagging) 100 customers which exhibit higher propensity to purchase one of either three financial products (a Mutual Fund (MF), a Credit Card (CC), and a Consumer Loan (CL)), with the overall goal of optimizing targets so as to maximize revenues.

Tables & data:

- soc_dem: 1615 records, 3 null values (Sex column);
- prods_actbal: 1615 records, high number of null values in counts & balances fields;
- inf_outf: 1587 records, 28 records missing compared to the soc_dem table, no null values;
- sales_rev: 969 records, no missing values.

The following sections will provide a walkthrough of the main steps undertaken in solving the problem as defined above.

2.2 Methodology

Overall methodology in terms of top-line steps is explained in the executive summary. In this report the focus will be on giving an explanation while reading through the code. The steps will be:

- EDA: from loading the data to having a good understanding of the variables;
- Feature engineering: preparing the data as best as possible for ingestion (iteratively);
- Modeling: developing, training, and testing the models; and
- Prediction: applying the models to the records with no response variables.

2.3 EDA

To generate an analytical DataFrame ("full" in the code), the soc_dem, prods_actbal, and inf_outf have been merged, using a left join from soc_dem to have all the rows. The "response" DF is a right join of the resulting DF with the sales_rev DF, so as to only keep records with data for the target fields.

The next logical step in a data science pipeline is to take a first glance at the data to get a sense of the distributions involved, and then split the data in a training and test set. Splitting before doing a full-fledged EDA is done to prevent the "data snooping bias", where the analyst risks to be biased by patterns in the test data which could influence the choice of models and processing steps.

Preliminary analysis suggests that the distributions are approximately exponential, with a strong right skew and a very long tail. These two characteristics of the data are to be kept in mind during modeling.

Data is then split in a training and test set, with a customary 80/20 train/test split. Descriptive statics are observed at this point, reinforcing the observation that features exhibit a long tail and several outliers, which are however a feature in the data, not broken data points.

Analyzing the three products, it becomes apparent that they are not equally profitable: average revenues are higher for CL and CC, which also have higher customer counts. More details are available in the notebook comments & notes.

The correlations between features & response variables are also taken into consideration (see notebook for more details and the correlation map).

2.4 Feature engineering

The goal of this section, to be iterated in part with the modeling so as to identify the best-performing combinations of models and feature engineering steps, is to prepare the data as best as possible for ingestion in machine learning algorithms.

The approach for missing values is defined at this stage: counts & balances variables for many records are missing, even when we have a response variable. A reasonable assumption here is that features with a null count, given that the feature descriptions mention a "live" balance / count, pertain to records with no active product, so these values can be imputed with 0s. To do otherwise would either reduce tangibly the number of available features or datapoints (an attempt at removing the columns was made, which expectedly produced a material reduction in predictive power).

Volume and transaction features are filled with the mean (chosen because of greater performance and higher robustness versus extreme values). The few missing values for sex are removed because with a split near 50/50 of the two sexes, imputing would be fairly random.

Two features have been created: “balance_tot’ and “balance_CA”, as subtraction of the volume credit and volume debit fields. These features are different from ActBal features as the former portray an average over the last few months, while the latter are the actual balance at the moment of observation. Other feature combinations have been attempted, but these were the ones which added the most value.

In terms of scaling and outlier handling, a first glance at the distribution would have suggested that the features would have required either a power transformation or a log transformation to have their distribution approximated to a normal distribution. Some models tend to underperform if the distribution is not normal. However, using either of those transformation produced worse results than scaling the data with the StandardScaler. Log/power transformations could have helped a linear regression model for revenues, so this is a point to consider as further step.
The approach for managing extreme values was to Winsorize the top/bottom 1%. The transformation and threshold have been chosen as they outperformed alternatives (such as removing outliers or using a higher threshold). Winsorizing is preferable to removing in this case because extreme values are not spurious observations, rather valid ones.

A pipeline is done to perform the imputing and scaling steps, to make sure there is no leakage from training to test data.

One note on chosen features: all features have been used as removing some did not improve predictive performance. Counts and balances features for the target variables have been kept (e.g. ActBal_MF for the MF variable) because it was assumed that one customer can purchase more than one instance of the same product (e.g. multiple MFs). Were this not to be the case, it would simply be a matter of removing those features from the model (e.g., taking out ActBal_CC and Count_CC if customers cannot own more than a credit card). Whether this is the case should be verified with business stakeholders.

2.5 Modeling

Once the data is deemed ready, the next step is choosing a few algorithms to be trained and assessed on their predictive power. This is an iterative process where the hyperparameters are tuned to identify the best performing combinations, usually with a combination of grid searching and manual tuning. Multiple feature engineering steps and pipelines can also be tested to search for a best option.

The process is composed of two broadly defined steps:

- Training: the models are trained on the 80% of the data, with each model and hyperparameter combination being assessed through cross-validation to provide an estimation of their generalization capacity (balancing the bias/variance tradeoff); and
- Testing: once a few best-performing models are identified, they are evaluated against the test set, using data which the models never saw. This step usually involves a hopefully slight loss in model scores. Higher loss could mean overfitting.

Multiple algorithms have been tried (e.g. logistic regression, SVC, random forest), and the best performer was an XG Boost model. Hyperparameters were first tuned by grid search, then further adjusted manually.

As described in the executive summary, the algorithms were optimized for precision as no classifier scoring high on both precision and recall could be developed and given the constraint on targeting 100 customers precision was the straightforward choice. Thresholds have been chosen by comparing the precision and recall curves visually (see notebook for exact scores and curves).

Metrics are uneven across the 3 target variables: the model for CL outperforms significantly the other ones, and MF is the worst performer:

	precision	recall	f1-score	support
0	0.84	0.98	0.91	158
1	0.70	0.19	0.30	36
accuracy			0.84	194
macro avg	0.77	0.59	0.61	194
weighted avg	0.82	0.84	0.79	194

	precision	recall	f1-score	support
0	0.79	0.99	0.88	144
1	0.87	0.26	0.40	50
accuracy			0.80	194
macro avg	0.83	0.62	0.64	194
weighted avg	0.81	0.80	0.76	194

Figure 1. Confusion matrices for the MF (left) and CL (right) classifiers on test data.

2.6 Prediction

The chosen algorithms are thus validated, and their predictive power defined. Of the 1615 data points, 40% do not have response variables, and these are assumed to be customers who can be targeted with direct marketing.

The processing steps described above are applied to this subset of the data and propensity scores computed. To offer an easier to interpret propensity score, since for all models the predicted scores were between 49% and 51%, three fields with the standardized predicted probabilities have been added.

Customers have been "flagged" in this order (this is accessible in the "flag" variable of the full_pred_df DataFrame):

1) All prospects with score > threshold for CL have been flagged since the model is the best performing and the product the most profitable (45);
2) The same step is applied for CC, as the model is better than that for MF and the product is still more profitable than MF on average (37); and
3) The remaining 18 available "places" are assigned to MF prospects, ranking prospects with score > threshold by their "TransactionsCred_CA", which is correlated to purchasing MF.

Effectively predicting revenues through a regression analysis was not possible in the given time as no model with acceptable metrics could be produced. As a fallback option, expected revenues were thus estimated, by multiplying two variables:

- Average revenues for customers purchasing target products; and
- Model precision for target product.

The resulting value is used as expected revenue for a given customer having a higher propensity than threshold for a given product. An alternative could have been multiplying average revenues times propensity score, but the chosen option was favored as it was assessed as methodologically sounder and more conservative.

2.7 Limitations and next steps

Revenue estimation is the main practical limitation of the proposed solution, as it is a simplistic and most likely still inaccurate estimation, which is also not capable of differentiating high-value prospects (who could buy higher amounts) from low-value ones. Given more time, a better performing regression model could be developed, improving the revenue estimation.
Another limitation is scalability, as mentioned in the executive summary.

Concerning possible next steps, some points mentioned above should be clarified or validated with business stakeholders before deploying this solution (e.g., whether all three products should be targeted or only targeting CL is acceptable, or whether a customer can own multiple products of the same type).

Additionally, the models could be refined to improve the revenue estimation and their scalability – it is unlikely that, were the direct marketing campaign to have a good performance, the bank would settle with targeting 100 customers at a time.

Finally, working in tandem with marketeers a potentially worthwhile activity would be to use the insights derived from the developed models (i.e. feature importance) and the flagged customers to gain a deeper understanding of "customer profiles". This could be useful to develop data-driven personas to better tailor marketing messages to each identified category.