Universiteit Antwerpen

Academic year 2021/22

Faculty of Business and Economics

AXA DATA SCIENCE CHALLENGE

Data Mining

Prof. xxx

Group members:

XX

1. Business Understanding

We as students from the Faculty of Business and economics at the university of Antwerp are asked to carry out the AXA data science challenge. The challenge falls within the framework of the course 'Data Mining' lectured by Prof. xxx. This assignment has been constructed following the CRISP-DM (Cross-industry Standard Process for Data Mining) model. In this brief overview we will discuss those steps conforming to this model. AXA Bank Belgium is the sixth Belgian bank by balance sheet total and occupies a strong position in the market for professional loans. Despite covid19, AXA Bank Belgium again recorded strong commercial and financial results for 2020: an increase in the production of professional loans (12750 new applications or +3%). This growth can be countered with strong predictive models, which underlines the relevance of our work. The aim of this challenge is to find out the creditworthiness of the customers. An application score evaluates the risk of a credit application based on information about the business information, socio-demographic data, financial data and behavioural data. We were provided with the dataset Axa_Training.xlsx which holds a bunch of variables. Those features will predict the target variable. In the next part (data understanding) we will elaborate the data and indicate some insights.

We as students from the Faculty of Business and economics at the University of Antwerp were asked to carry out a group.

2. Data Understanding

We examined the dataset and looked at the number of data instances (25913) which represented professional loans. For each loan there are 44 corresponding attributes and a target variable. We began with observing the most important column namely 'Label_Default'. For this characteristic, we can distinguish two classes: the instance has a 'N' (meaning that they paid back their loan) or a 'Y' (meaning that they defaulted the loan and could not meet all payments within a period of 24 months from the start of the loan). Furthermore we observed that both continuous and categorical variables were present in the data set. Therefore we will have a split into three components. Out of 25 913 recorded observations 726 could not meet their financial responsibilities of the loan. This results in a default rate of 2.8%. What immediately struck us is that there are far fewer defaulters than non-defaulters. We will therefore have to pay enough attention to minority groups and recude the risk of oversampling.

We dug deeper into the data to find potential relationships among the data. We first made a correlation table, we noticed that most variables do not correlate with each other. Nevertheless are there some exceptions which we will come back to later. In a next phase we will divide our dataset into a train-, a test- and a validation set. As the test set is mostly a random part of the full dataset, we expect that the covariance structure is the same. Problems concerning multicollinearity is therefore not a topic in the data understanding. Secondly we used boxplots to visualize a number of continuous variables. We plotted for each variable two boxplots: one for the non-defaulters and the other for the defaulters. We assume that the effect of a variable on the prediction is big if the first, second and third quartile deviate significantly between the non-defaulters and defaulters. Finally we calculated the standard statistics (mean, standard deviation) of the different variables, and we could see that the continuous variables contain very high standard deviations. This indicates a dataset with many outliers. We will take this into account later in the preprocessing of the data.

3. Data Preparation

The first step into our data preparation process was dividing the variables of the dataset so that similar variables could be prepared in the same way. The variables were divided into three groups, namely continuous variables, discrete variables and variables that had a missing-value percentage higher than 50%.

As required, we also had to choose the indices of our training-, testing- and validation set. We chose a test size of 20% on the full 'DSC_2021_Training.xlsx' dataset. The remaining 80% was split into training (80%) and validation (20%). In total, out of our full 'DSC_2021_Training.xlsx' dataset with 25.912 observations, 16.583 were used for training the model, 5.183 were used for testing the model and 4.146 were used for validation purposes.

Because it was crucial to obtain the best possible pre-processed dataset for the later modelling stage, we did several iterations during the data preparation phase. Each iteration used different data pre-processing techniques that led to different pre-processed datasets. In order to see which pre-processed datasets performed best, we tested these after each iteration on several default models and noted their AUC. Default models were models of which the parameters weren't optimized (i.e. using the default Scikit-learn model parameters) and were used to reduce the computation time needed to optimize parameters. The classification algorithms that were used are Logistic Regression, Random Forest, Support Vector Classifier, Stochastic Gradient Descent and Decision Tree. By doing the data preparation phase iteratively, we made sure that our final pre-processed dataset would be close to optimal for the next phase.

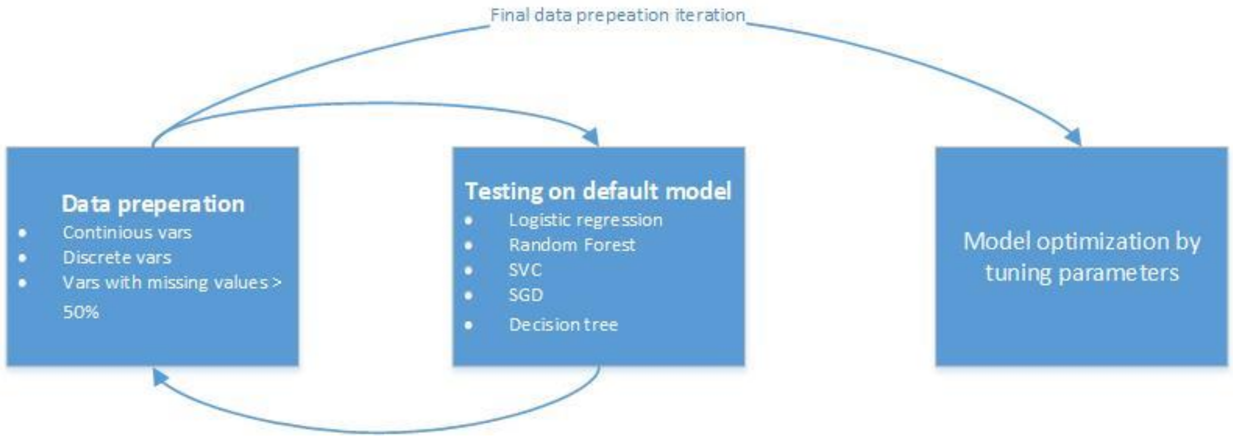

Figure 1: Iteration process of data preperation phase

In what follows, we'll discuss the data preparation techniques that were used during the final iteration (i.e., the techniques that led to the final pre-processed dataset). We'll also discuss some of the considerations that were made during the iteration process.

3.1 Continuous variables

The group of continuous variables contained a total of 25 variables. The first step we took was leaving out the variables that had a correlation coefficient greater than 0,90. The variables with which those variables were correlated were obviously left in. By doing this, we made our model more robust. The variables that were left out were A1_AVG_NEG_SALDO_PROF_3_AMT and A1_AVG_POS_SALDO_PROF_12_AMT. We also did iterations using a correlation coefficient cut-off of 0,85 (6 variables left out), 0,8 (7 variables left out) and 0,5 (9 variables left out) but noted that the overall performance of the default models was best at the coefficient of 0,9.

Furthermore, we filled the missing-values of the remaining variables by using the mean of the training set as the fill-in value. We then standardized those variables by using the formula $z = \frac{(x-\mu)}{\sigma}$. Using standardized data gave us the opportunity to handle outliers by changing z-scores that were greater than 3 or less than -3 to the values 3 and –3 respectively.

3.2 Discrete variables

The group of discrete variables contained 10 variables (excluding Managing_Sales_Office_Nbr and Postal_Code_L). During preparation of the discrete variables, we used several techniques which will be discussed in what follows. As we've mentioned earlier, this process was done iteratively.

k-1 dummy encoding

Most of the discrete variables were pre-processed using k-1 dummy encoding. The missing values of those variables were filled using the mode of the training set. One of the iterations was using k dummy encoding instead of k-1, but we noticed that the models performed better using k-1 dummy encoding.

INDUSTRY_CD_3 and INDUSTRY_CD_4

The variables INDUSTRY_CD_3 and INDUSTRY_CD_4 contain NACE-codes which are industry standard classifications of economic activities. NACE-codes are organized in 5 levels where level 1 is the broadest classification of an economic activity and level 5 is the narrowest classification of an economic activity[1]. Because of the hierarchical nature of NACE-codes, you could take any NACE-code from a certain level and trace it back to its broadest classification (level 1). Take for example the NACE-code '222' from INDUSTRY_CD_3 (level 3) which represents the economic activity 'Manufacturing of plastic products'. When we trace it back to its broadest classification we get the NACE-code 'C' which represents the economic activity of manufacturing as a whole.

Based on this knowledge, we were able to write a function that reduced the high cardinality variables INDUSTRY_CD_3 and INDUSTRY_CD_4 to only one variable INDUSTRY_CD_1 which contained the level 1 NACE-codes of the separate instances. The result was that the number of separate discrete values was significantly reduced to only 21 and that the variable INDUSTRY_CD_1 could be dummy encoded.

Because reducing the dimensionality meant that we had lost some of the valuable data contained in INDUSTRY_CD_3 and INDUSTRY_CD_4, we also decided to include INDUSTRY_CD_4 by using weight-of-evidence encoding as described by D. Martens and J. Moeyersoms in their article about handling high-cardinality variables.[2]

[1] https://www.vlaanderen.be/economie-en-ondernemen/een-eigen-zaak-starten/nace-code
[2] https://www.kdnuggets.com/2016/08/include-high-cardinality-attributes-predictive-model.html

3.3 Variables with missing-value percentage higher than 50%

The group of variables with a missing-value percentage higher than 50% contained 8 variables which were all preprocessed in the same manner. These variables were prepared by transforming each variable into a binary variable, where 1 and 0 where defined respectively as:

- 1 = has a value that is not equal to Nan
- 0 = has a value that is equal to nan

The reasoning behind this technique also actually makes sense. Take for example the variable MONTHS_SINCE_LAST_REFUSAL_CNT, which contains the number of months since the last time an application has been refused. One could presume that if an instance has a missing value for this variable, that the instance hadn't had an application refused in the past. We could therefore reinterpret this variable when applying the binary transformation as APPLICATION_REFUSED_IN_PAST with 0 being 'no application refused in past' and 1 being 'one or more applications refused in past'.

4. Modeling and Evaluation

4.1 Tuning for optimal parameters

In this phase, we went ahead and took our final pre-processed dataset to optimize the different model-parameters. We obtained the optimal parameters by performing a grid search that used the training and validation set and used the AUC as its metric. The obtained results are summarizes in the table below.

Classification algorithm	**Parameters**	**Optimal parameters**	**AUC**
Logistic Regression	solver C	liblinear 0.5	**0,86843**
Random Forest	n_estimators criterion	701 entropy	**0,91877**
Support vector classifier	C	96	**0,86895**
Stochastic Gradient Descent	penalty	l1	**0,86554**
Decision Tree	criterion min_samples_leaf	gini 260	**0,80232**
K-Nearest Neighbors	n_neighbors	331	**0,86748**

Table 1: testing results of different models

Oversampling

Because there was a significant underrepresentation of defaulters in the dataset (default rate = 2,8%), we also trained our model using an oversampled training set. Oversampling means duplicating instances from the minority target class in order to increase the number of minority classifications in the training set and thus balance out the dataset. By doing this, the default rate in our oversampled training set equalled 23%. Even though the AUC-scores obtained using the oversampled training set didn't vary much from the ones using the normal training set, it is important to mention that the Logistic Regression and Stochastic Gradient Descent models did record lower false negative rates using the oversampled training set. This could be important to consider when doing churn prediction since false negatives (not detecting a defaulter that is a defaulter) are generally considered to be much more costly than false positives (not granting someone a loan who should have gotten a loan)

4.2 Generating predictions

Pre-processing the prediction set

What we've done previously was the pre-processing and training of our model using the dataset that contained the target variables. Because one of the goals of this assignment is using the trained model to make predictions about data with an unknown label, we had to go through the same pre-processing step with our data upon which we had to make predictions. This was done by combining both files (DSC_2021_Training and DSC_2021_Test) into one aggregated dataset on which we could run through the same pre-processing steps as done before.

Training the model

In order to make predictions, we had to retrain our model. This time however, we could use the entire training dataset containing 25.912 instances. As discussed previously, the Random Forest classification algorithm gave us the higher AUC-score. Consequently, we used this model to make the final probability predictions. To reduce false negatives, we also applied oversampling on our training set.

5. Conclusion

The AXA bank Data Science challenge was our first real hands-on experience into the application of machine learning and data mining. Using the study material that was provided in our master course 'Data Mining', we were able to make predictions about whether or not a professional loan would result in default. Whilst going through the assignment, the importance of the CRISP-DM model became very clear: data mining projects require a circular approach instead of a linear approach in order to complete the task successfully. This became especially noticeable during the data preparation phase we went through several iterations in order to get the best pre-processed dataset.

The assignment offers a great learning experience by being confronted with a real-life business problem. Solution-oriented thinking and working efficiently were for us the keystones to get the job done.