**Business Analytics and Data Science: Churn Prediction Assignment**

**1. Introduction**

*1.1 Data Science in Churn Analysis*

Classification methods in Data Science have a wide range of applications for business problems. In this assignment, machine-learning techniques are applied to analyze real word data of an unknown telecommunications company in the USA to develop a model that predicts, as accurate as possible, whether a customer will churn or not. It is well known that in industries like that one where there’s a lot of competition in the market, customer acquisition is very expensive and therefore trying to minimize the proportion of people that quit a contract is a very valuable thing to do. More accurate detection of people that will churn makes it possible to design more effective marketing campaigns or reward systems aimed at retaining these customers.

… Something a bit more scientific about churn analysis. Quote something (Papers from Nikoletta)

In the rest of this introduction the main characteristics of the dataset will be presented. Section 2 is dedicated to data preparation. The most important steps and decisions –and their justification– will be developed. The two different strategies used for variable selection will be explained in section 3. We tried a wide variety of models: Naïve Bayes, Logistic Regression, Support Vector Machine, Artificial Neural Networks, Random Forests and Gradient Boosting. After getting the first round of results, we concentrated our effort into improving the results for the last two. The description of this process will be the subject of chapter 4. In chapter 5 we summarize the results and in Chapter 6 we present a brief conclusion.

*1.2 Description of the data set*

The complete dataset consists of 100,000 “mature customers” –*i.e.* who were in the company for at least six months– that were sampled during 2001 and 2002. 171 variables capture customer’s information about 1) socio-demographic, economic and geographic characteristics and 2) their usage and experience with the service provided by the telco company. Our target variable is “churn” –only contained in the test dataset–, which was calculated, according to the description provided, “based on whether the customer left the company during the period 31-60 days after the customer was originally sampled”. 49.56% of the customers in the training set churned, while the rest, 50.53% did not churn. This means that, for this dataset there is no class-imbalance problem that had to be dealt with.

**2. Data Manipulation**

Every data analysis starts with importing the data. We used the option ‘stringsAsFactors’ = FALSE to deal with the data classes manually. The process of transforming the variables was not a one-step process. We first made some basic adjustments, calculated correlations and analyzed the information value given by the WOE procedure and then made some extra manipulations. Here just a summary of the changes will be presented.

To ensure that both the training and the test dataset contained the exact same variable transformations and factor levels, therefore avoiding problems for the prediction, the two datasets were merged into one using the command *smartbind* from the package *gtools.* After all the necessary transformations were made they were split again.

*2.1 Variable transformations*

– *age1* and *age2*. After transforming the large amount of zeros into NAs, we decided to bin this variable into the following categories: (0,30], (30,40], (40,50], (50-60], (60, max(age)] and relabel the NAs as “Missing”. For each case, a dummy variable was created that indicates whether the value was missing or not.

– Roaming-variables (*rmcalls*, *rmmou*, *rmrev*). These all had a high number of NAs (48083), we calculated the correlation between the non-NAs and churn and since the correlation coefficient was too low, decided to drop them from the analysis. We however included a dummy variable that indicates whether the observation had a valid number or not.

– *retdays* (48083 NASs) . NAs were transformed to missing and four categories according to the .25, .5 and .75 quantiles of the valid values were created. A dummy variable *retdays\_bool* was also created with the levels “Called” and “Not called” (the retention team)

– *csa*. This variable seemed to have potential explicative power –according to the importance scores calculated in the WOE Method–, however due to the fact that it consisted of over 700 levels, it turned out to be impossible to include it in our model. We extracted the city (contained in the first 3 characters) but this shortened version of *csa* (*csa\_cities*) lost its explicative power and did not make it into the pre-selected variable list for modeling.

– *last\_swap*. This variable gives the date of the last swap. To extract more meaning out of this information, we calculated the number of days that had passed between the last swap and an assumed date for the analysis of “2002-01-31”. Since this variable introduced too many NAs, we decided to bin it using the variable quantiles (.25, .5 and .75) of the valid observations and assigning “Missing” as an extra category for the rest.

*2.2 Missing values*

Two different strategies on how to treat missing values were applied depending on the type of the variable in question. For categorical variables –and also for numeric that were binned into categories–, we created an extra category labeled “missing” that enters normally to the models. As for the numeric variables, after the variable transformations described in the previous section, we were left with a dataset for which none of the remaining numeric variables had more than 3% NAs. This meant that we got a data set in the end with 48,127 observations without NAs to build our models. In the case of the test set used for prediction, we tested our models both the mean and the median value of the variable in question.

**3. Variable Selection**

Our first approach to variable selection was splitting the dataset into factor and numeric variables to get a first idea of their relationships of these variables with our target variable churn using appropriate methods in each case.

*3.1Filter approach*

*3.1.1 Factor variables: Weights Of Evidence (WOE) and Information Value Score*

The WOE Method is a useful start to analyze whether a level in one factor exhibits a considerable difference in the proportion of people that belong to one of the categories of the target variable relative to the other –in this case whether a person churned or not churned.

The Information Value (IV) is a measure derived from the WOE to assess not just a level of a variable but the whole variable and also to be able to compare among variables.

A useful rule of thumb is picking the variables with an IV bigger than 0.02 (which are considered to be at least weakly explicative). In our case, out of the 63 variables that entered to the model, only 9 met this criterion.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 1. Variables’ Information Value (WOE)** | | | | | | | | |
| csa | hnd\_price | hnd\_webcap | last\_swap | crclscod | retdays\_factor | tot\_acpt | retdays\_bool | tot\_ret |
| .072 | .052 | 0,03 | 0,029 | 0,027 | 0,029 | 0,023 | 0,021 | 0,21 |

*3.1.2 Numerical variables: correlation coefficient*

For the numerical variables we decided to use, as a first approach, the simple correlation coefficient, using the option that all the non-missing values for each pairwise set of variables are used. Normally we would have set a rule for which we would have kept all variables with at least a .5 correlation coefficient with the target variable. However out o the 111 variables, not even one had a correlation coefficient bigger than the established value. We therefore concluded that the relationship between the numerical variables, if any, could not be linear and therefore cannot be captured by correlation. Therefore this approach would not be right for variable selection.

*3.2 Random Forest for variable selection*

Given the above results, we decided to use the Random Forest Method directly to select our variables. This had also the extra benefit that we use a single method that can handle both numeric and categorical variables. To do this we used the option “importance = TRUE” to get the scores on Mean Decrease of Accuracy for each variable.

The big challenge in this data set is that there is not one or a small set of variables that really help predict churn. We have to include several variables in model assuming that each will help to explain little parts. This, of course, with the risk of over fitting our data. We decided to reduce our data set to contain the top 40 variables and apply the models and for each one decide the best amount of variables.

Here we present the top 40 variables and their scores:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 2. Variables’ Importance Scores (Random Forest)** | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| eqpdays | months | mou\_Mean | totmrc\_  Mean | last\_swap | hnd\_price | adjrev | change\_mou |
| 0,0068 | 0,0041 | 0,0020 | 0,0016 | 0,0013 | 0,0013 | 0,0012 | 0,0012 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| mou\_cvce\_  Mean | avg3mou | totrev | mou\_Range | mou\_opkv\_  Mean | totcalls | phones | avg3qty |
| 0,0012 | 0,0011 | 0,0011 | 0,0010 | 0,0010 | 0,0009 | 0,0009 | 0,0008 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| complete\_Mean | peak\_vce\_  Mean | comp\_vce\_  Mean | totmou | opk\_vce\_  Mean | mou\_peav\_  Mean | adjqt | avgqty |
| 0,0008 | 0,0008 | 0,0008 | 0,0008 | 0,0008 | 0,0008 | 0,0008 | 0,0007 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| adjmou | rev\_Mean | ovrmou\_  Range | rev\_Range | ovrmou\_  Mean | plcd\_vce\_  Mean | attempt\_  Mean | mou\_rvce\_  Mean |
| 0,0007 | 0,0007 | 0,0007 | 0,0007 | 0,0007 | 0,0006 | 0,0006 | 0,0006 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| models | pre\_hnd\_  price | ovrrev\_  Mean | avgrev | ovrrev\_Range | ethnic | avg6mou | avg6rev |
| 0,0006 | 0,0006 | 0,0006 | 0,0006 | 0,0006 | 0,0006 | 0,0006 | 0,0006 |

**4. Modeling**

One of the things that had the most impact in our modeling decisions was the computational capacity of running our models. This is why we invested time in learning how to parallelize the computation of our models –this will be discussed in more detail later in this chapter.

We decided, as a first approach, to try several different models by dividing our training dataset into 70% of the observations to real training and the rest to predict and assess the predictions. This first round of results let us to three competing models – Artificial Neural Networks, Random Forest and Gradient Boosting–, for which a more accurate assessment of the performance of the model was needed. For these three models we performed 10-fold Cross-Validation and averaged their performance metrics –lift score and area under the ROC-Curve–.

*4.1 Naïve Bayes*

Naïve Bayes is a statistical classification method based on probability theory that models the conditional probability of class membership given attribute values. The decision rule for classification is to assign the observation into the group that results in the maximum posterior probability. This method relies on the assumption that the attributes are conditionally independent.

Three models were tested with different number of variables –5, 15 and 30 respectively. The results for the AUC and the lift score were the lowest ones from all models, therefore no further time was dedicated to improve the models.

|  |  |  |
| --- | --- | --- |
| **Table 3. Naïve Bayes Results** | | |
|  | AUC | Lift Score |
| 5 variables | 0.561 | 1.164 |
| 15 variables | 0.558 | 1.146 |
| 30 variables | 0.564 | 1.198 |

4.2 Logistic Regression

4.3 Artificial Neural Networks

4.4 Support Vector Machines

4.5 Random Forest

4.6 Gradient Boosting (the winner)

**5. Results**

Table with results . Measures of AUC & LIFT.

Optional – Graph ROC-Curve for a random fold of the cross-validation?

How did we make the decision.

**6. Conclusion**

**7. Appendix**

7.1 List of packages that were used in the analysis

– leaps

– randomForest

– klaR

– Hmisc

– caret

– gtools