Business Analytics Case Study
Customers Churn Prediction in the Telecommunication Industry

Seminar: Business Analytics and
Predictive Processing
Dr. Stefan Lessman
March 2015

Laura Gabrysiak Gomez
Humboldt−Universität zu Berlin
Master in Information Systems
laura.gabrysiak@gmail.com
MatrNo: 555091

Frederick Hegemann
Humboldt−Universität zu Berlin
Master in Information Systems
hegemanf@hu−berlin.de
MatrNo:551938

Table of Contents

Introduction (Task Description)

The following report describes the procedure and the final results of a project for the seminar Business Analytics and Predictive Modelling. The project and developed model are based on a real life case study of customer churn of a large US Telecommunications enterprise. The aim of this report is to document the approach undertaken and to share some experiences from the teams point of view while analysing the task and developing the final model.

On Business Analytics

Business Analytics is a recent discipline which studies the use of statistical techniques as well as different technologies that can be applied to the analysis of (in many cases) large and complex datasets in order to gain deeper insights about the nature of the data. Business Analytics allows us to find and better understand complex patterns that are often times hidden in data. This is among other achieved through the recognition of patterns which are then used for future forecasting. Business analytics is closely related to another modern and quite popular discipline called "Business Intelligence (B.I.)"; which deals with relevant metrics and tries to adapt quantitative analysis to specific business cases in more strategic ways.

The scope of the seminar "Business Analytics and Predictive Modelling" is to provide the students with real life challenges of predictive modelling. In this case, the scope of the project was to develop a model for customers churn prediction.

On Customers Churn – Theory

The Customer Churn measure refers to the probability that current customers of a company stop using its services for different reasons, for example because they switch to rivalry companies. Customers churn companies mean a great loss to companies revenue, partly because acquiring new customers (as one measure to compensate leaving customers) is more costly than retaining existing customers (Colgate & Danaher, 2000, Lessman 2014) in many cases these customers are very unlikely to switch back (source). Also, long-term customers generate higher profits, are less sensitive to competitive actions, and may act as promoters trough positive word of mouth (e.g., Ganesh, et al., 2000; Reichheld, 1996; Zeithaml, et al., 1996). The prior forecasting of a customer churn can be used to prevent customer/revenue loss by reacting on time to the situation, for instance by launching marketing campaigns or other type of "bribing" the customer to remain (retention measures).

Customer churn is present in almost each industry specially in the service ones: such as banking, insurances and telecommunication. Specially in telecommunication services is an important factor to consider due to the flexibility characteristic of this field. For example, T-Mobile USA lost half a million of its most lucrative customers in the first quarter of 2012 (Bensinger & Tibken, 2012). It is known that contract churn rates for many types of communication services are in the three-percent-per month range (Kim, 2010). This means that a provider needs to refresh nearly 100% of its customer base about every three years. However, studies show there is an acceptable annual churn rate 5-7%[1] across all industries.

Voluntary vs. Involuntary churn

Literature differentiate between voluntary and involuntary customer churn. Customer churn has been a topic of vast study and discussion in the last years due to its relevance to the industry. Several models and theory has been developed and yearly there is a challenge to reach the optimal prediction model.

The scope of this seminar project is to make use of standard procedures and tools of the business analytics field to develop a prediction model that estimates churn risks accurately. A dataset with the customers churn rates of a large US Telecommunication enterprise was provided to train and test the model.

Data - General description

The data provided consists on a data set from an US Telecommunication enterprise of a total of 100 000 instances (customers) each described by roughly 171 variables. The data set was equally split into a training- and testing set with each 50 000 instances. The variables capture demographic, socio-, and also micro-geographic information from the customers such as for example "average call number per month", "ethnicity" or "type of car they use".

The training data set is additionally annotated with the information if the given customer has churned or not ("churn").

Variable types

From these 173 variables it can be distinguished between two groups of variables: the categorical and the numerical (also known as continuous) ones.

Out of the 173 variable provided in the training and test-dataset, 29 variables were categorical (17%) and 144 numerical (83%). Categorical variables were provided with a description of for example the rate of missing values. Numerical variables on

[1] http://sixteenventures.com/saas-churn-rate

the other hand, were provided together with a description of average value, missing values rate and min and max values. This information will be shown to be very helpful specially when preprocessing the data.

While continuous/numerical variables express a numerical measurement such as "average calls per months" which also can be processed mathematically without any further concerns, categorical variables express (as the name indicates) discrete categories such as "type of car they use". This type of variables cannot be directly processed mathematically and it will be later shown the different possibilities to handle them.

The section below provides a description on how each step of the data processing was implemented and the intuition behind each decision.

Data Preparation and Transformation

A large part of the total assignment was to prepare and pre-process the data before building and applying the prediction model. By this, we refer to the standardization of the data so that it can be uniformly be used by the different algorithms.

This part of the assignment was very time consuming and required an overview and intuition of the desired outcome. Although the task of the assignment was to construct a prediction model of churn, the most effort was undertaken by cleaning the available data and selecting the important features. Moreover, the task of data preprocessing consisted of missing value-handling, feature selection, data normalisation (#numericals), data discretization(#numericals) and dummy-encoding. The handling of missing values is necessary for any kind of statistical or analytical applications since many of the common used methods cannot naturally deal with missing values. Feature selection is especially important in data mining, since data scientists often cope with data-sets of hundreds and thousands of features any many of them are correlated, interdependent or simply not useful.

The difficulties were manifold. On the one hand side, the literature is full of methods and techniques to deal with missing values and feature selection of highdimensional data. On the other hand, there is often no rule of thumb, which cut−off−value or threshold has to be chosen. Hence, the data cleaning−process was an iterative process. The feature−sets were adjusted several times, depending on the outcome and limits of the model. Another difficulty arose from the fact, that the success of used approaches was only explorable a posteriori. We applied different approaches to clean and preprocess the data, but many failed in the end and did not improve the prediction success.

STEPS

In KDD there is a given procedure of how to handle when processing the data. The procedure is defined by the following steps, which were adopted in this project as well. The tasks previously mentioned can be seen as substeps/tasks of the different phases of the process

1. Selection (this step was skipped since data was already provided)
2. Cleaning
3. Transformation
4. Reduction

It should be remarked that the process is not necessarily a linear one and the lack of independence between the steps lead to further difficulties while processing the data. For instance the first intuition when analysing the data, was to first reduce the large amount of variables and later to clean and transform them. For this aim, it was intended to make use of a Pearson's correlation matrix which turned out to be useless due to the many missing values. Generally speaking, the idea of reducing variables first was obstructed in most of the cases by the missing values or outliers.

For this reason it was decided to create a sort of hybrid, iterative model. Starting with a pre−selection of data which was then cleaned, always differentiating between the categorical and numerical variables which are handled differently as we will see in the following sections. As a further step, we proceeded to reduce dimensions and finally to transform the variables in one type or the other for example via dummy encoding.

Selection and Cleaning:

There are several ways of dealing with outliers, missing values and noise generally. In all cases, and quite intuitively, it can be decided to keep-, delete- or replace the noisy values. Each of these approaches bring along their own advantages and disadvantages such as bias, information loss etc.

As already stated, there are different ways on how to treat data from one type or the other. The following sections describe the cleansing of the data based on their structural features (I.e if categorical or numerical data)

Categorical variables

1. Harsh Pre-selection:
   In order to avoid a larger pre-processing and later feature reduction process the first intuition was to run a manual harsh pre selection and test each variable's semantic and availability due also to the fact that the amount of categorical variables (17%) wasn't very large.

   The first aspect to consider for the pre selection was the rate of missing values (which ranked from 0.001% to 96%). It was decided to chose as a first subset only variables with missing values rates below 20%[2]. Other variables didn't have large missing value rates yet their values contained a very high amount of "unknown" values which can be considered as missing.

   The variables that were discarded after the Harsh pre-selection were:

```
ADULTS,AGE1,AGE2,CARTYPE,CHILDREN,CRTCOUNT,DIV_TYPE,DWLLSIZE,DWLLTYPE,EDUC1,HHSTAT
IN,INFOBASE,Alle KID-
Variablen,LOR,MAILFLAG,MAILORDR,MAILRESP,NUMBCARS,OCCU1,OWNRENT,PCOWNER,PRE_HND_PR
ICE,PROPTYPE,REF_QTY,SOLFLAG,TOT_ACPT,TOT_RET,WRKWOMAN

#In addition to the variables with large amounts of "unknown" values:
AGE1,AGE2,New_Cell,MARITAL,KID2-17,
```

2. As a second aspect, we used a black-box-software "Salford TreeNet[3]" to run a regression on the k-folded training dataset to find out the so called "Variable Information". This measurement is based on the weight of evidence (WoE) and the goodness or badness of the fit to model. Generally, it provides a good intuition on how relevant the variable is for the model. The

[2] Unfortunately we didn't find any "golden-rule" for deciding on a value to discriminate when analysing the missing value rates.

[3] www.salford-systems.com/products/treenet

Salford TreeNet Software adjusted all values to an optimal model and provided us thus with the IV of each categorical variable.

3. Special cases
   The variables “CSA” and “last_swap” were discarded even though they had large IV due to the large number of levels they presented. The variable CSA referred to the “local area of service” and its (approx. 708) values were conformed by a nine digit sequence. CSA was very difficult to aggregate in order to reduce the amount of variables. For this aim, an approach was to run a clustering analysis on both variables together with other demographic and sociographic data such as income, ethnics etc. in order to come up with supra-categories. For this aim we made use of Pearson's correlation matrix however the process was interrupted after 1 h due to the large amount of missing values. Last swap on the other hand referred to the amount in days since the customer had changed his mobile phone. The difficulty with this variable was that each amount of days was considered a factor and not a numerical value.

Finally, 17 variables (descriptors) in addition to customer_id and churn remained. Below, the R command to create the first subset of relevant variables.

```
#CATEGORICAL: manual Cleanup - delete categorical variables as written in the
report
subset1=subset(trainingset, select=-
c(csa,adults,age1,age2,car_buy,cartype,children,crtcount,div_type,dwllsize,dwlltyp
e,educ1,HHstatin,infobase,kid0_2,kid3_5,kid6_10,kid11_15,kid16_17,lor,mailflag,mai
lordr,mailresp,marital,mtrcycle,numbcars,occu1,ownrent,pcowner,pre_hnd_price,propt
ype,REF_QTY,solflag,tot_acpt,tot_ret,wrkwoman,new_cell))
```

Continuous variables

1. The same Harsh pre-selection approach was undertaken with the numerical variables with the difference that (on the contrary to the categorical variables) only few had missing values rates above 20%.
2. Also, a certain number of variables were discarded which could be inferred by other variables. An example for these variables that could be derived from other were among others: all aggregated variables, average measurements, etc.

```
retdays,rmcalls,rmmou,rmrev,attempt_Mean,attempt_Range, complete_Mean,
complete_Range,drop_blk_Mean,drop_blk_Range,ovrrev_Mean,ovrrev_Range
```

3. As a third step, all variables correlating with a min of 0.75 were discarded. For this aim we made use of the Pearson's correlation matrix which assumes the linear relation between the parameters.

4. Search for important variables:
   In the next step various algorithms were tested to find the best possible explanatory variables in the subset. The approaches include: Best.First.Search, Backward.Search and hill climbing. Depending on the process, some variables were more important than others, however, it is to be remarked however, that the variable `eqpdays` turned out to be the most important variable every time. The explanatory value (1−error−value) for the continuous variables was always in the range 0:57 to 0:58.

The data was later also sorted out with a similar R command as above:

```
#CONTINOUS: manual cleanup - delete continous variable as written in report due to
missing
subset1=subset(subset1, select=-c(csa,retdays,rmcalls,rmmou,rmrev)
```

As a remark, the R code was designed so that when any missing value encountered it should be skipped. This does not refer to variable selection but to the values/levels of all remaining variables.

Transformation

The transformation step mainly considers the standardization of the data. As previously mentioned, continuous and categorical data cannot be handled the same way due to their structural characteristics and thus they have to be brought to the same format/standard, which is also referred as standardization.

As a reminder, at this step of the process we have two subsets of a reduced clean data with no missing values (see remark above) One subset is conformed by 17 categorical variables and 25 numerical variables.

The main intuition for this part of the process was that we had three possibilities on how to proceed: either keep both variable types and handle them with different types of algorithms or we rather convert all variables into one type (standardized data) and proceed with one type of algorithm. Many sources point out to standardize the data in order to later be able to reduce as many dimensions as possible. After doing some research we decided to convert all the data into one type: either all data would be binary or numerical encoded. These were our two main approaches which will be described in this section.

Approach 1: Converting all variables to numerical

Since the categorical data was the smallest part of the total data (17%), the first intuition was to convert all categorical variables into binary ones and later to reduce the dimensions.

1. Categorical Variables: (Categorial - Binaries – Numerical)
   For this attempt, we first encoded all categorical data with a dummy variable. It was considered to then convert the binary variables into numerical ones with the Point biserial correlation approach[4], however we didn't pursue this approach due to the lack of time left. This method can be subject of future analysis.

   It was also the consideration to convert into numerical values with the weight of evidence measure (WoE). Some of our research pointed out that the Pearson matrix can be used to later reduce the dimensions. However, some problems arose with the high complexity and availability of the information.

2. Numerical Variables: Here there wouldn't be any transformation needed.

Approach 2: Converting all variables to categorical (binary)

The motivation for pursuing this approach was to convert all variables with dummy–encoding in order to run a logistic regression (and/or Neural Network) which is a quite simple and fast approach. Unfortunately the fact that most of the variables were continuous meant a drawback in our work specially because the amount of binary variables (dimensions) exploded. However, since they are saved as num (numeric) they require little storage and the performance of R didn't suffer much from this.

1. Categorical Variables (Categorical – Binary)
   To convert the categorical variables into binary they were encoded with dummy variables.

2. Numerical Variables (Numerical - Categorical - Binary)
   The intuition of this approach was to first convert numerical variables into categorical ones (aka discretization) and later convert the categorical

[4] https://stat.ethz.ch/pipermail/r-help/2008-July/168703.html

variables into binary ones. With discretization it is meant the mapping of continuous values into categories such as for example different ages mapped into age groups. The advantage of this approach is that the impact of outliers is avoided but on the other side the disadvantage is that information gets lost.

For the discretization we tried out 3 different algorithms. All algorithms (The functions `chi2, mdlp`(minimum discription length) and `ChiMerge`) belonged to the R package "discretization". Unfortunately all algorithms had to be aborted after almost 18h of running due to the decision of further reduce the variables to better to discretize. We are still unaware if the performance of R or rather the data quality played here a major role.

A following approach, as above explained, was to further reduce the numerical variables using the Pearson's correlation matrix. This turned out to work fine however when discretizing Pearson's correlation values (all values below 0.5 which were filtered out) the process had to be aborted after 10h due to time.

We did some research and we found out an unpublished PhD thesis from the University of Florida[5] which makes use of the Winnow algorithm[6] to discretize Pearson's correlation values and use them as factors. We tried that out and it turned out to work well, however after the Information Gain ratio was very low.

In this stand of the process we had a matrix with 195 binary variables.

Reduction

After, creating the new dataset with the 195 variables we weren't able to further reduce the dataset

Data Mining - Model/Algorithms

We chose for our prediction model two different models: on the one hand side, we made use of the Adaboost algorithm, a boosting algorithm. On the other hand side, we decided to make use of the LogitRegression model. For smplicity issues, these algorithms will ne be explained in detail in this work.

It was considered to try out an ensemble approach by aggregating the final results

[5] https://www.linkedin.com/pub/nooshin-nabizadeh/40/8b5/828

[6] http://www.cs.cmu.edu/~avrim/ML10/lect0120.txt

of each model however this couldn't be pursued due to the limited time. This would be also implemented in future research.

Remark – TreeNet

TreeNet is a commercial Data Mining tool developed by Salford Systems (Jerome Friedman and Leo Breiman of Stanford University) Because it is a commercial software we explicitly decided not to use it as our main model predictor, also because it functions with a black−box principle. However, the motivation to make use of this software came because TreeNet was the winner of the Golden Prize of the KDD customers churn prediction challenge in 2003[7] and it contains the Adaboost algorithm in its library. Adaboost was implemented later in R but TreeNet allowed us to create an intuition in many cases.

	Adaboost Algorithm	LogitRegression Model
PCC	0.56 (average)	0.54 (average)
Class_Error (1−PCC)	0.44 (average)	0.51 (average)

Model Assesment

Unfortunately we couldn't implement a testing function on R so we had to undertake the testing in TreeNet which created a 35000 instance test sample from the test dataset. However the table above shows the final results. If LogitRegression did perform much better by predicting no churners (61%), in average was the performance lower as the one from the Adaboost algorithm which performed better with churners (churn = 1) (63%). Both models however did grade quite similarly.

You will find in the Appendix further details to the developed models. Figures 1 belongs

Difficulties encountered – Insights from this work

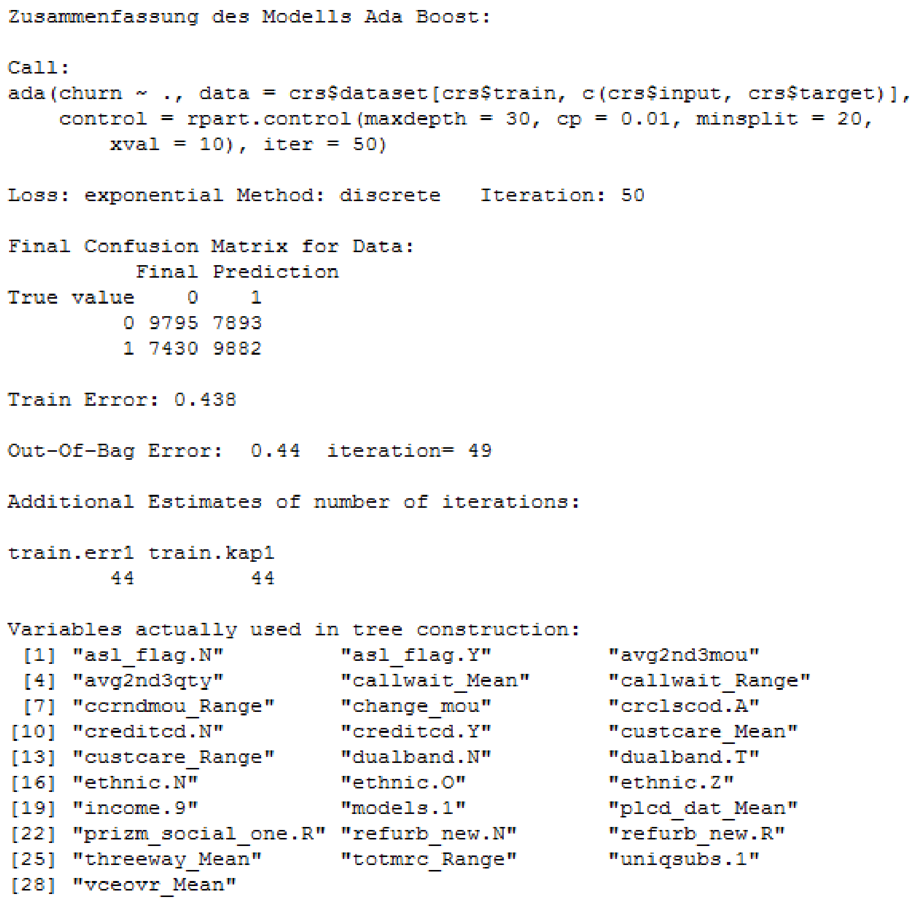
[7] http://link.springer.com/article/10.1007%2Fs10479-008-0400-8

1. When preparing the data it is rather better to make sure that both training and test datasets are formatted equally. In other words
2. Appropriated algorithms weren't easy to implements and were aborted after 12−30h. Thus, it wasn't possible to discretize the data and even after trying an easier algorithm to implement (the Winnograd logarithm) the data was discretized faster but the additional information (the added value of factorizing) was unattended and thus rather low compared to the raw/uncleaned data.

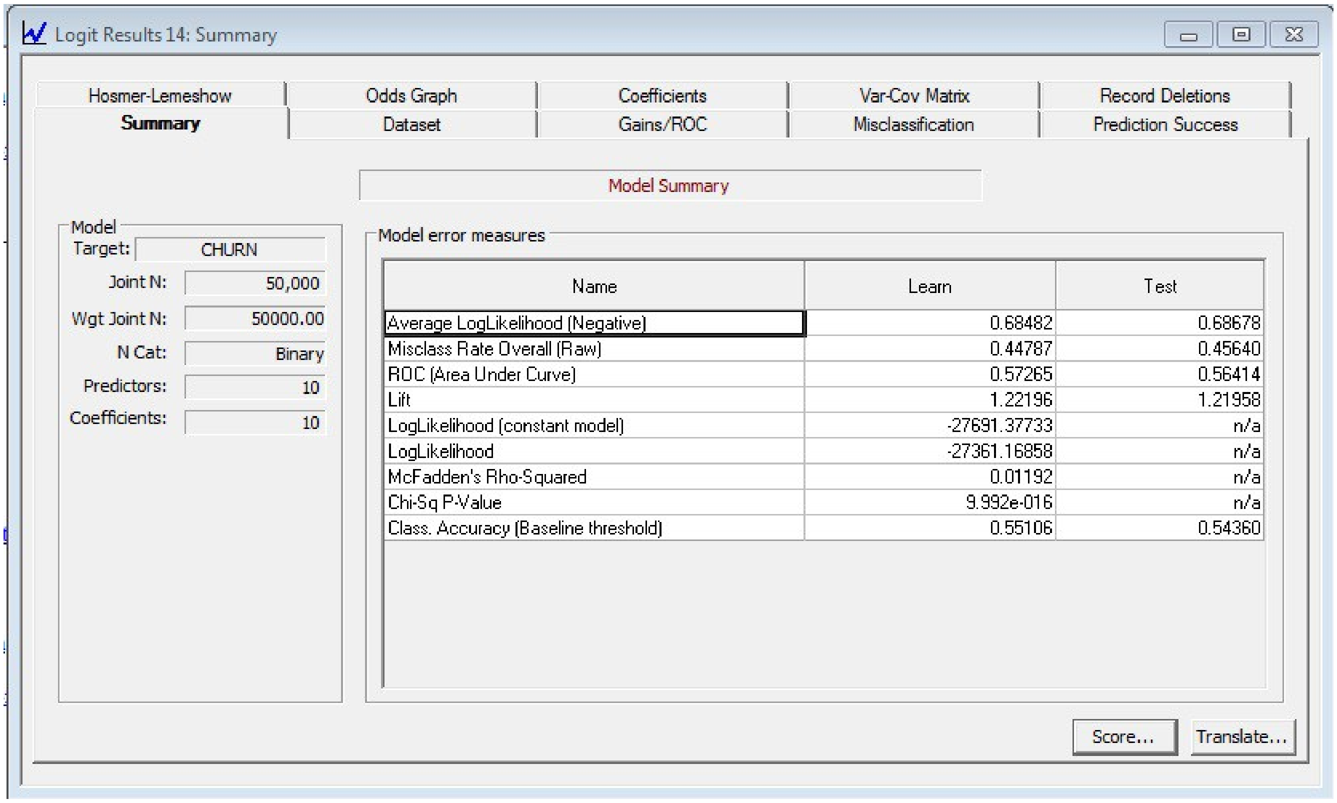
Conclusions

The task of building a prediction model of churn probability of customers was not only informative and exciting but difficult in many ways. As mentioned before, the core part of our work consisted of data cleaning, data preprocessing and feature selection. Altough the focus should have set on building a prediction model, our focus was set on handling and cleaning the data. Many models and methods can be used to extract information and predict churn probabilities but all will fail when the used data is too noisy. When we started to work on our assignment, we were eager to apply the algorithms and techniques, learned in our class. The interdependence between features, algorithms and models was a difficulty too. Many algorithms, like chi2−discretization or PCA−algorithms needed too much time to adjust some parameters. Many methods assume normal distribution on continuous data, but a normal distribution was not always easy to achieve. For further research, it could be useful to imply data from different sources then customer data, like macro− and microeconomical data.

Appendix



```
Zusammenfassung des Modells Ada Boost:

Call:
ada(churn ~ ., data = crs$dataset[crs$train, c(crs$input, crs$target)],
    control = rpart.control(maxdepth = 30, cp = 0.01, minsplit = 20,
        xval = 10), iter = 50)

Loss: exponential Method: discrete   Iteration: 50

Final Confusion Matrix for Data:
          Final Prediction
True value    0    1
         0 9795 7893
         1 7430 9882

Train Error: 0.438

Out-Of-Bag Error:  0.44  iteration= 49

Additional Estimates of number of iterations:

train.err1 train.kap1
        44         44

Variables actually used in tree construction:
 [1] "asl_flag.N"          "asl_flag.Y"          "avg2nd3mou"
 [4] "avg2nd3qty"          "callwait_Mean"       "callwait_Range"
 [7] "ccrndmou_Range"      "change_mou"          "crclscod.A"
[10] "creditcd.N"          "creditcd.Y"          "custcare_Mean"
[13] "custcare_Range"      "dualband.N"          "dualband.T"
[16] "ethnic.N"            "ethnic.O"            "ethnic.Z"
[19] "income.9"            "models.1"            "plcd_dat_Mean"
[22] "prizm_social_one.R" "refurb_new.N"        "refurb_new.R"
[25] "threeway_Mean"       "totmrc_Range"        "uniqsubs.1"
[28] "vceovr_Mean"
```



Model Summary

Model

Target: CHURN

Joint N: 50,000

Wgt Joint N: 50000.00

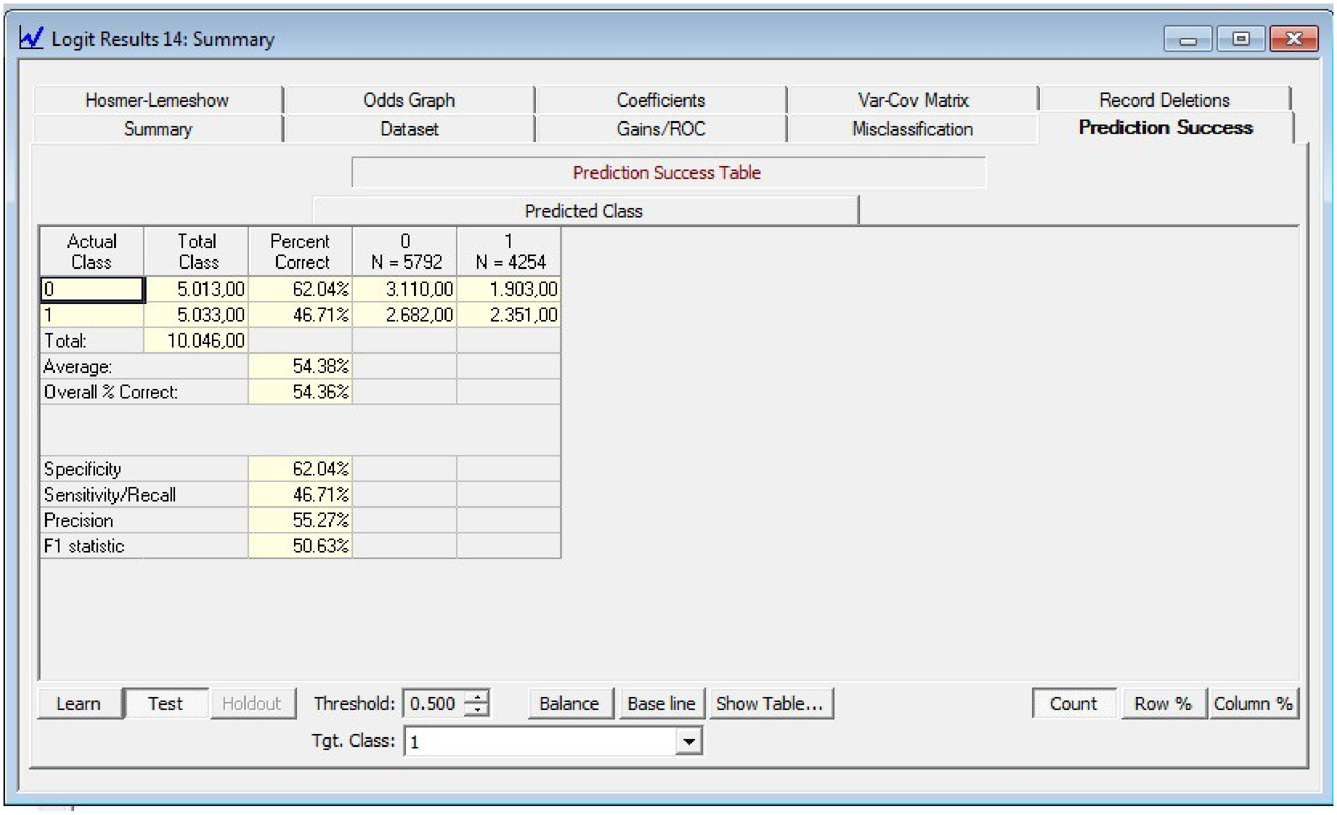
N Cat: Binary

Predictors: 10

Coefficients: 10

Model error measures

Name	Learn	Test
Average LogLikelihood (Negative)	0.68482	0.68678
Misclass Rate Overall (Raw)	0.44787	0.45640
ROC (Area Under Curve)	0.57265	0.56414
Lift	1.22196	1.21958
LogLikelihood (constant model)	-27691.37733	n/a
LogLikelihood	-27361.16858	n/a
McFadden's Rho-Squared	0.01192	n/a
Chi-Sq P-Value	9.992e-016	n/a
Class. Accuracy (Baseline threshold)	0.55106	0.54360



Prediction Success Table

Predicted Class

Actual Class	Total Class	Percent Correct	0 N = 5792	1 N = 4254
0	5.013,00	62.04%	3.110,00	1.903,00
1	5.033,00	46.71%	2.682,00	2.351,00
Total:	10.046,00			
Average:		54.38%		
Overall % Correct:		54.36%		
Specificity		62.04%		
Sensitivity/Recall		46.71%		
Precision		55.27%		
F1 statistic		50.63%		

Threshold: 0.500

Tgt. Class: 1

```
#author=Frederik Hegemann, Laura Gral Gomez
#Loading used libraries
library(FSelector)
library(FactoMineR)
library(caret)
library(corrplot)
library("psych")
library("MASS")
library(reshape)
library(ade4)
library(discretization)
library("Rcmdr")
library(RcmdrPlugin.FactoMineR)
library(Factoshiny)

##Prehandling
##1. Create subset of numerical data
##(2.delete correlated features by spearman-correlation)
trainingset=data.frame(read.csv("C:\\Users\\Boon\\Dropbox\\BAPM_WiSe14\\Data\\BAPM_Trainingset.csv", header=TRUE))
testset=data.frame(read.csv("C:\\Users\\Boon\\Dropbox\\BAPM_WiSe14\\Data\\BAPM_Testset.csv", header=TRUE))
# delete categorical variables as written in the report, also delete customer_id.

subset1=subset(trainingset, select=-c(adults,age1,age2,car_buy,cartype,children,crtcount,Customer_ID,div_type,hnd_webcap,dwllsize,dwlltype,educ1,HHstatin,infobase,kid0_2,kid3_5,kid6_10,kid11_15,kid16_17,lor,mailflag,mailordr,mailresp,marital,mtrcycle,numbcars,occu1,ownrent,pcowner,pre_hnd_price,proptype,REF_QTY,solflag,tot_acpt,tot_ret,wrkwoman,new_cell))

#delete continous variable as written in report due to missing and aggregated features
subset1=subset(subset1, select=-c(retdays,rmcalls,rmmou,rmrev,attempt_Mean,attempt_Range,complete_Mean,complete_Range,drop_blk_Mean,drop_blk_Range,ovrrev_Mean,ovrrev_Range))
#Factorization
factors=c("actvsubs","area","asl_flag","churn","crclscod","creditcd","csa","dualband","ethnic","forgntvl","hnd_price","income","last_swap","models","phones","prizm_social_one","refurb_new","rv","truck","uniqsubs")
nobinfacts=c("csa","last_swap","churn")
subset1[factors]=lapply(subset1[factors],as.factor)
#delete cats for binary-encoding:
factorset=subset1[,(names(subset1) %in% factors) & (!names(subset1) %in% nobinfacts)]
nummset=subset1[,!(names(subset1) %in% factors)]


#splitting AVG6 month in second3-month because data correlates with first 3month
nummset$avg6mou=nummset$avg6mou-nummset$avg3mou
#print(nummclass$avg6mou[1])
nummset$avg6qty=nummset$avg6qty-nummset$avg3qty
nummset$avg6rev=nummset$avg6rev-nummset$avg3rev
names(nummset)[names(nummset)=="avg6mou"]="avg2nd3mou"
names(nummset)[names(nummset)=="avg6qty"]="avg2nd3qty"
names(nummset)[names(nummset)=="avg6rev"]="avg2nd3rev"
#print(nummset$avg2nd3qty[2])
#eventually delete na's
#na.omit(nummset)
numms=c(names(nummset))


#Reduction of continuous features by Correlation-Based Approach
#Select either pearson, spearman or kendall
```

```
corrtype="pearson"
correlationmatrix=cor(nummset, y=NULL, use='pairwise', method=c(corrtype))
#Find high correlated feature; change cutoff for better result
cutoff=0.50
highlycorrelated=findCorrelation(correlationmatrix, cutoff=cutoff)
print(highlycorrelated)
corrplot(correlationmatrix,order="hclust")
nummset=nummset[,-highlycorrelated]
corrplot(correlationmatrix[-highlycorrelated,-highlycorrelated], order="hclust")


## Test-set-Preparation
subtest1=subset(testset, select=-c(adults,age1,age2,car_buy,cartype,children,crtcount,Customer_ID,div_type,hnd_webcap,dwllsize,dwlltype,educ1,HHstatin,infobase,kid0_2,kid3_5,kid6_10,kid11_15,kid16_17,lor,mailflag,mailordr,mailresp,marital,mtrcycle,numbcars,occu1,ownrent,pcowner,pre_hnd_price,proptype,REF_QTY,solflag,tot_acpt,tot_ret,wrkwoman,new_cell))
#delete continous variable as written in report due to missing and aggregated features
subtest1=subset(subtest1, select=-c(retdays,rmcalls,rmmou,rmrev,attempt_Mean,attempt_Range,complete_Mean,complete_Range,drop_blk_Mean,drop_blk_Range,ovrrev_Mean,ovrrev_Range))
subtest1=data.frame("churn"=subset1$churn,subtest1)
#Factorization
factors=c("actvsubs","area","asl_flag","churn","crclscod","creditcd","csa","dualband","ethnic","forgntvl","hnd_price","income","last_swap","models","phones","prizm_social_one","refurb_new","rv","truck","uniqsubs")
#excluded factors
nobinfacts=c("csa","last_swap","churn")
subtest1[factors]=lapply(subtest1[factors],as.factor)
#delete cats for binary-encoding:
factorsettest=subtest1[,(names(subtest1) %in% factors) & (!names(subtest1) %in% nobinfacts)]
nummtest=subtest1[,!(names(subtest1) %in% factors)]
#splitting AVG6rev into second3month-Range
nummtest$avg6rev=nummtest$avg6rev-nummtest$avg3rev
names(nummtest)[names(nummtest)=="avg6rev"]="avg2nd3rev"
#Delete highly-correlated variables from testset.
nummtest=nummtest[,-highlycorrelated]
numms=c(names(nummtest))



## Dummy-Encoding for test
##
binnumstest=nummtest
for (i in 1:ncol(binnumstest)){
  binnumstest[,i][binnumstest[,i]>median(nummtest[,i],na.rm=TRUE)]=1
  binnumstest[,i][binnumstest[,i]<=median(nummtest[,i],na.rm=TRUE)]=0
}

##Dummy-Encoding for train
binnumset=nummset
for (i in 1:ncol(binnumset)){
  binnumset[,i][binnumset[,i]>median(nummset[,i],na.rm=TRUE)]=1
  binnumset[,i][binnumset[,i]<=median(nummset[,i],na.rm=TRUE)]=0
}


#Categorical-Handling - Not working properlywith testset
#Level-Reduction
#Factor-Handling
# sets=c(factorset,factorsettest)
# appset=sets
# #Combining levels of actvsubs by 0,1,2,3
# levels(appset$actvsubs)=list("0"=c(1),"1"=c(2),"2"=c(3),">=3"=c(4:12))
# #Combining Levels of Creditcard with F including all others after E
# levels(appset$crclscod)=list(A=c(1:4),B=c(5:7),C=c(8:13),D=(14:18),E=c(19:25),F=c(26:54))
# #Combining levels of hnd_price by bis 50, 50-100,100-150,150-200,>200
# levels(appset$hnd_price)=list("0-50"=c(1:3),"50-100"=c(4:6),"100-150"=c(7:9),"150-
200"=c(10:12),">200"=c(13:17))
# #Combining levels of models by 1,2,3,>3
# levels(appset$models)=list("1"=c(1),"2"=c(2),"3"=c(3),">3"=c(4:25))
# #combining levels of phones by 1,2,3,>3
# levels(appset$phones)=list("1"=c(1),"2"=c(2),"3"=c(3),">3"=c(4:13))
# #combining levels of uniqsubs by 1,2,3,>3
# levels(appset$uniqsubs)=list("1"=c(1),"2"=c(2),"3"=c(3),">3"=c(4:15))
#
#
# appset=se
# #Combining levels of actvsubs by 0,1,2,3
# levels(appset$actvsubs)=list("0"=c(1),"1"=c(2),"2"=c(3),">=3"=c(4:12))
# #Combining Levels of Creditcard with F including all others after E
# levels(appset$crclscod)=list(A=c(1:4),B=c(5:7),C=c(8:13),D=(14:18),E=c(19:25),F=c(26:54))
# #Combining levels of hnd_price by bis 50, 50-100,100-150,150-200,>200
# levels(appset$hnd_price)=list("0-50"=c(1:3),"50-100"=c(4:6),"100-150"=c(7:9),"150-
200"=c(10:12),">200"=c(13:17))
# #Combining levels of models by 1,2,3,>3
# levels(appset$models)=list("1"=c(1),"2"=c(2),"3"=c(3),">3"=c(4:25))
# #combining levels of phones by 1,2,3,>3
# levels(appset$phones)=list("1"=c(1),"2"=c(2),"3"=c(3),">3"=c(4:13))
# #combining levels of uniqsubs by 1,2,3,>3
# levels(appset$uniqsubs)=list("1"=c(1),"2"=c(2),"3"=c(3),">3"=c(4:15))

#Binary-Encoding for categoricals train
binfacts=acm.disjonctif(factorset)
binfactschurn=data.frame("churn"=subset1$churn,binfacts)
binset=data.frame(binfacts,binnumset)
binsetchurn=data.frame("churn"=subset1$churn,binnumset,binfacts)
#Binary-Encoding for categoricals test
binfactstest=acm.disjonctif(factorsettest)
binfactstestchurn=data.frame("churn"=subtest1$churn,binfactstest)
binsettest=data.frame(binfacts,binnumstest)
binsettestchurn=data.frame("churn"=subtest1$churn,binnumstest,binfactstest)



complete_pearson=data.frame("churn"=subset1$churn,binset,factorset)
complete_pearsontest=data.frame("churn"=subtest1$churn,binnumstest,factorsettest)
#write complete set with pearson-reduced continuous and normal categoricals
write.table(complete_pearson,file="complete_pearson.csv",append=FALSE, quote=FALSE,sep=",", eol="\r",
na="NA",dec=".",row.names=FALSE,col.names=TRUE)
#write new test-csv
write.table(complete_pearson,file="complete_pearson_test.csv",append=FALSE, quote=FALSE,sep=",", eol="\r",
na="NA",dec=".",row.names=FALSE,col.names=TRUE)

#Write binset
write.table(binset,file="binset.csv",append=FALSE, quote=FALSE,sep=",", eol="\r",
na="NA",dec=".",row.names=FALSE,col.names=TRUE)
#write binsetchurn
write.table(binsetchurn,file="binsetchurn.csv",append=FALSE, quote=FALSE,sep=",", eol="\r",
na="NA",dec=".",row.names=FALSE,col.names=TRUE)
```

```
#write binfacts
write.table(binfactschurn,file="binfactschurn.csv",append=FALSE, quote=FALSE,sep=",", eol="\r",
na="NA",dec=".",row.names=FALSE,col.names=TRUE)

#write test-table with churn
write.table(binsettestchurn,file="binfactschurntest.csv",append=FALSE, quote=FALSE,sep=",", eol="\r",
na="NA",dec=".",row.names=FALSE,col.names=TRUE)



# #PCA-Approach
# res.pca=PCA(nummclass,scale.unit=TRUE,ncp=5,graph=TRUE,axes=c(1,2))
# summary(res.pca)
# plot(res.pca,choix="ind")
# dimdesc(res.pca,axes=1:2)
#
#
# #chi2(discretization,failed)
# chi2(nummclass,0.5,0.05)$cutp
# chi2(subset1,norm,0.5,0.05)$Disc.data
```