Benchmarking of different classes of models used for credit scoring

We use this competition as an opportunity to compare the performance of different classes of predictive models. In particular we want to compare the industry standard method (logistic regression) with Boosting Trees, Random Forrests, MARSplines and neural network.

Knut Opdal
StatSoft Norway AS
Oslo, Norway

Rikard Bohm
StatSoft Norway AS
Oslo, Norway

***Abstract*— in this article we study the performance of different classes of score models.**

***Keywords*— unbalanced class problem, reject inference, out-of-time prediction, robustness, choice of class of models, AUC.**

I. INTRODUCTION

Logistic regression is the industry standard when it comes to credit scoring. A typical scorecard based on logistic regression is build by using 12-15 predictors – all categorical each with typically 4-5 categories, and only main effects. It puzzles us that this type of extremely simple score models are preferred when modern data mining techniques routinely outperform this kind of logistic regression.

The major advantage of using such simple models are that they are easy to understand and also easy to hardcode into the bank systems. The question is: are these advantages still relevant if the bank has access to modern data mining technology where it is easy to build more accurate models and these models easily can be deployed without hard coding?

A. Statistical data analysis vs. general predictive (pattern recognition) models

There are important differences between traditional statistical modeling methods (e. g. logistic regression) compared to the general predictive modeling techniques now widely adopted across many domains (Breiman, 2001; Nisbet, Elder, and Miner, 2009; Miner, Elder, Hill, Nisbet, Delen, and Fast, 2012).

Statistical analysis and modeling is based on testing of hypotheses and the estimation of population parameters from samples, and statistical inference. For example, in logistic regression the parameters of a linear model are estimated that predict some outcome variable y (e.g., credit default risk) as a linear function of the predictor variables.

In general (and without going into details about the theory of statistical inference), the approach for building such models is to estimate the parameters of the model from a subsample of cases, and then to perform statistical significance tests to decide if the model parameters and predictions from the model are more accurate than some baseline (usually random) expectation.

The general approach to modeling is to test a set of a-priori expectations regarding possible functional relationships between predictors and outcomes for statistical significance (i.e. an outcome that is highly unlikely to have been observed in the sample by chance alone). If found to be significant, it is concluded that the respective relationships thus hold in the population at large. So fundamentally, statistical modeling is about hypothesis testing, and the rejection or acceptance of a-priori hypotheses about the data, and how the important outcomes can be predicted from available inputs (predictor variables).

B. Pattern recognition: Data are the model

Modern data mining, machine learning, or predictive analytics approaches are different. In these approaches, no a-priori hypotheses are tested, but instead the goal of the analysis is to extract from the dataset repeated patterns and relationships that are useful for the accurate prediction of future outcomes.

II. THE PROBLEM

This paper will use the data and the problem definition formulated by the BRICS-CCI & CBIC 2013 Congress. BRICS is an acronym that refers to economic group of Countries that includes Brazil, Russia, India, China and South Africa, all of them at a similar stage of newly advanced economic development. The congress arranged a data mining competition, see below:

The BRICS-CCI & CBIC 2013 Congress is pleased to host the first data mining competition in Brazil, co-organized by NeuroTech S.A. within the scope of the event. This year's Competition, is on the well known application of credit scoring. However, this time it focuses on the effects of temporal degradation of performance and seasonality of payment delinquency. As an interactive environment there is a real-time LeaderBoard for stimulating the competitors' daily

participation and allow some parameter adjustment. The webpage offers a moderated forum for interaction about the competition. The competition is open for academia and industry and can be accessed either through the BRICS-CCI & CBIC 2013 Conference site (brics-cci.org) or directly to the competition server (brics-cci.neurotech.com.br).

Task 1: Robustness against performance degradation caused by market gradual changes along few years of business operation. This task will be evaluated based on the usual area under the ROC curve metrics (AUC_ROC; a Java routine for calculating it is available for download). Innovative ways of handling this matter can be found in PAKDD 2009 Competition whose focus was on this type of degradation and also in more recent concept drift approaches.

Task 2: Fitting of the estimated delinquency produced by the estimation model to that observed on the actual data for the applications approved by the model for Task 1. This very realistic task represents an innovation in data mining competitions worldwide by emphasizing the relevance of the quality of future delinquency estimation instead of the usual lowest future average delinquency. The distance D from the Chi-square statistics will be used for evaluating the quality of the monthly delinquency estimated, provided that the average delinquency is kept within half and double of the actual delinquency in the period.

I. DATA

Three datasets were downloaded from the site: http://brics-cci.neurotech.com.br/downloads/:

Datasets	Number of records	Cumulative (Count)	Percent
Modeling	762966	762966	60,18
Leaderboard	60000	822966	4,73
Prediction	444828	1267794	35,09

Table 1. Three datasets were downloaded

The modeling dataset is the only dataset that includes the response variable (TARGET_LABEL).

A. Data preparation

The modeling dataset was divided by us into three samples, training, testing and validation:

Modeling dataset	TARGET_LABEL		Records
	Good	Bad	
Test	112379	40213	152592
Train	337824	119956	457780
Validation	104360	48234	152594
All Grps	554563	208403	762966

Table 2. The modeling dataset was split into 3: Test, Train and out-of-time validation.

In addition to the original data downloaded from the site, we have derived a few variables from the original variables such as day of year and number of approved applications per week.

II. UNBALANCE

A. Unbalance due to rejected applications

When it comes to credit scoring, there are usually only data about the company's clients for modeling, not about the rejected applicants. These represent a sample of the potential clients (market) that is strongly biased given that a systematic procedure focused on the problem target (payment default) has been applied for their selection.

A common perception is that this problem can be addressed using a suitable method for *reject inference*. Our opinion is that you need additional information about the rejected applications for such methods to be successful since "you can't know what you don't know".

The most straight forward way to collect such information is (1) simply make sure that a (small) random sample of the rejected applications regularly is approved. Most banks are not willing to accept the extra risk by using this approach.

Another way is (2) to purchase external information of the future payment behavior of the rejected applicants. This is the reason FICO, Experian, Equifax, TransUnion all will advise you to use reject inference to avoid ending up with a very biased model. They want their customers to purchase the external information needed to handle the reject inference problem. In this competition neither (1) nor (2) is possible to use.

Another consideration is that a number of applications that a new model will reject were approved by the old model. Therefore, we actually do have information about applications that will be rejected by the new model. Consequently we think there is very little to gain using methods for reject inference in this competition, and we simply ignore this unbalance.

B. Out-of-time prediction

Credit score models are always based on historic data and applied to future applications. The performance of a model will decrease over time. The relationship between the predictors and the response will change over time, and this will cause the models to perform poorer and poorer as time goes by. We try to minimize this effect by using these two steps:

Step one: Split the dataset into three samples: Training, test and validation. We use the most recent 20% as the validation dataset, and the test and training is drawn by random from the oldest part of the available date.

We will measure the performance of the different model candidates on the validation dataset. This way we will try to find the models that will perform best in the future.

Step two: After deciding which model to use, we will recalibrate the model by randomly dividing the validation sample into test and training samples to make sure the most recent data is used to build the model you are going to use when predicting future payment behavior.

The table below shows how the training, testing and out-of-time validation data is distributed across time:

YearMonth	Test	Train	Out-of-time Validation
200901	4585	13757	0
200902	3685	11053	0
200903	4882	14648	0
200904	5072	15213	0
200905	6770	20311	0
200906	7250	21750	0
200907	6814	20445	0
200908	6984	20950	0
200909	6974	20923	0
200910	8220	24663	0
200911	9674	29018	0
200912	14590	43770	0
201001	5306	15918	0
201002	5092	15277	0
201003	7158	21471	0
201004	6774	20322	0
201005	8781	26346	0
201006	8617	25854	0
201007	8508	25521	0
201008	7878	23633	0
201009	7852	23559	0
201010	1126	3378	35568
201011	0	0	46530
201012	0	0	70496

Table 3. The modeling dataset was split into 3: Test, Train and out-of-time validation.

III. External information

There are a large number of records available for building score models in this competition, but not so many predictors. In real life there usually is a lot more information available for each applicant. In this respect the analytic problem for this competition is a bit different form real life situations.

It would therefore probably be possible to improve the model performance significantly if we could add geo- and demographic information about the geographical districts of Brazil. There appears to be a lot of highly relevant information on a very detailed level at this site [5], but we have not been able to merge any of this information to the dataset on Neighborhood\Urban sub district of residence level. Again, in real life, we would probably be able to merge this kind of information into the dataset, but in this case we have too little information to be able to perform such a merge.

We did however manage to merge unemployment rates per state and month. This variable ended up to play a significant role in our final model, as shown in table 3 below.

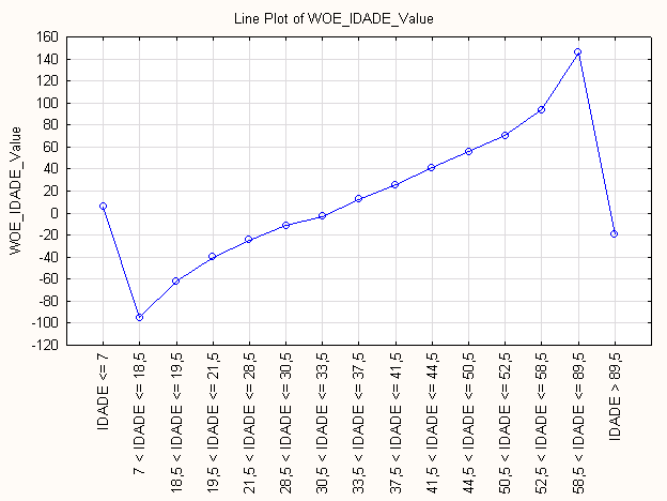
IV. Data cleaning

Except for replacing missing values with the constant -99 we have not applied any detailed data cleaning. In a real project we would have performed thorough descriptive analyses of each variable, to correct errors. Variable *Age* for instance contains some values below 10 years and above 150 years. In real projects we would look into these extreme values and replace them with the correct numbers; for this competition we cannot do this. However, the logistic regression model will probably not suffer too much because of this kind of data problem since we will build the model based on WOE (*Weight-of-Evidence*) transformed (binned) predictor variables. Extreme values and missing data will be placed in separate categories and handled separately. Modern predictive modeling techniques like Boosting Trees are quite robust to such data quality problems, and will also handle missing and extreme values adequately. So our approach is to simply leave the raw data as it is, and use WOE transformations or use variable selection techniques to decide whether a predictor should be included to the model or not.

V. Task 1: Modeling

First we developed a "baseline" model using logistic regression. The logistic regression was using pre-build WOE transformations as continues predictors. We allowed the WOE transformations to have one minimum and one maximum. An example of such a transformation is age, which also was one of the most important variables, see Fig. 2 below:

Fig. 1. WOE transformation of the Age variable.



Similar transformations were applied to all predictor candidates. This was done using a special tool designed for this purpose that automatically identified the most suitable recoding for all predictor variables in a single analysis, see [4] and [6].

We then used the "Best subset" function in the statistical software we used (see [4]) to find the best set of predictors. The predictors we used are listed in table 3.

Significant Predictors sorted by the Wald observatory	Wald
WOE_IDADE	8629,4
WOE_ZipCode_2	5077,6
WOE_SEXO	2153,7
WOE_RENDA_LIQUIDA	1522,9
WOE_DIA_VENCIMENTO	1307,2
WOE_CODIGO_PROFISSAO	866,4
WOE_ESTADO_CIVIL	826,9
WOE_FLAG_COMPROVANTE_RESIDENCIA	782,8
WOE_LOJA	719,2
WOE_FLAG_TELEFONE_RECADO	661,4
WOE_DayOfYear_2	555,0
WOE_FLAG_EXPERIENCIA_CREDITO	458,8
WOE_FLAG_NOME_PAI	410,0
WOE_TEMPO_EMPREGO	394,4
WOE_NUM_CONTAS_BANCO	314,7
WOE_COMPROVANTE_RENDA	269,5
WOE_ NumberOfLoansLast7days vsYearWeekMean	183,3
WOE_NUMERO_DE_DEPENDENTES	148,7
WOE_DDD_RESIDENCIAL	148,1
WOE_FLAG_TEM_OUTRO_CARTAO	104,3
WOE_FLAG_COMPRA_IMEDIATA	73,7
WOE_ NumberOfLoansLast7daysvstheweekbefore	66,6
WOE_TIPO_RESIDENCIA	45,8
WOE_FLAG_MESMA_UF_RESIDENCIA_COMERCIAL	45,1
WOE_FLAG_MESMA_CIDADE_RESIDENCIA_COMERCIAL	27,0
WOE_UNEMPLOYMENTINDEXPERSTATPERMONTH	25,4
WOE_FLAG_TEL_RESIDENCIAL	23,7
WOE_TEMPO_RESIDENCIA	19,5
WOE_FLAG_ENDERECO_CORRESP_IGUAL_RESIDENCIAL	16,1
WOE_FLAG_NOME_MAE	14,9

Table 3. The list of predictors used by the baseline model.

We used the baseline model to score the "Leaderboard" dataset (see table 1), and uploaded to the site: http://brics-cci.neurotech.com.br/registration-and-submissions/. It received the Gini coefficient 0.70, see table 4.

Rank	Results from submissions to the Leaderboard			
	Taks 1 Metric Value	***Taks 2 Metric Value***	***Method***	***Comment***
1	0,7186	1,54	MARSplines	MARSpline using only the 4 scores from Boosting Trees, MARSplines, Random Forrests and Logistic regression as predictors.
2	0,7168	1	Boosting Trees	Boosting trees using only the 4 scores from Boosting Trees, MARSplines, Random Forrests and Logistic regression as predictors.
3	0,7119	1,22	Marsplines	MARSpline using the best logistic regression score as predictor in addition to the rest of the selected predictors.
4	0,71	1,79	Boosting Trees	Boosting trees including the best logistic regression score as predictor in addition to the rest of the selected predictors.
5	0,7078	6,29	Boosting Trees	Boosting Trees on raw date
6	0,7067	1,21	Logistic regression	Logistic regression based on WOE transformations where the first part of 2009 was omitted. The first part of 2009 seemed to have higher average default rate than the rest of the modeling periode.
7	0,7008	***	Logistic regression	Logistic regression based on WOE transformations, using the WOE values as continues predictors. WOE transformation allowed one minimum and one maximum. It was required that each category would include at least 250 bads and at least 500 records all together. A "Best subset" procedure was used to select the set of predictors.

Table 4. Shows the results for our submissions to the leaderboard.

We do not expect a logistic regression model to be able to win this contest, so we tried out several modern predictive (pattern recognition) models.

Five types of classification algorithms were used to build model candidates: Boosted Trees (stochastic gradient boosting trees), Random Forest, MARSplines (Multivariate Adaptive Regression Splines), see [1,2,3], logistic regression and Neural Networks.

Our approach is the following:

Step 1: Build different score model candidates using different classes of methods. Use only the training subset to estimate the model parameters. For each of the classes of models we tried different parameter settings. Our experience is that the parameter settings are worth "tweaking", since they could have a great impact on the performance of the model. However, since we plan to use an ensemble of models we anticipate that the robustness of the final model will be less dependent of the parameter settings in each of the model candidates.

Step 2: Add the scores for the different candidates to the modeling dataset. Build a new model where the scores for the different candidates are considered as potential predictors. The

test sample is used for training and vice versa in step 2. The validation dataset is used to choose what predictors to chose.

Step 3: Recalibrate the final model after adding the validation sample to the training sample in order to use the most recent and relevant data for building the final model.

When using these kinds of models you have to decide on what predictors to use and each of the methods have separate parameter settings which can impact the performance significantly. We tried different sets of predictors, and ended up with this list of predictors:

Predictors	
Continues	Categorical
SEXO	ESTADO_CIVIL
IDADE	SEXO
DIA_VENCIMENTO	FLAG_TELEFONE_RECADO
TEMPO_EMPREGO	
RENDA_LIQUIDA	
DayOfYear	
UNEMPLOYMENTINDEX PERSTATPERMONTH	
NumberOfLoansLast7days	
WOE_ZipCode_2	
WOE_CODIGO_PROFISSAO_2	
WOE_DDD_RESIDENCIAL_2	
WOE_LOJA_2	

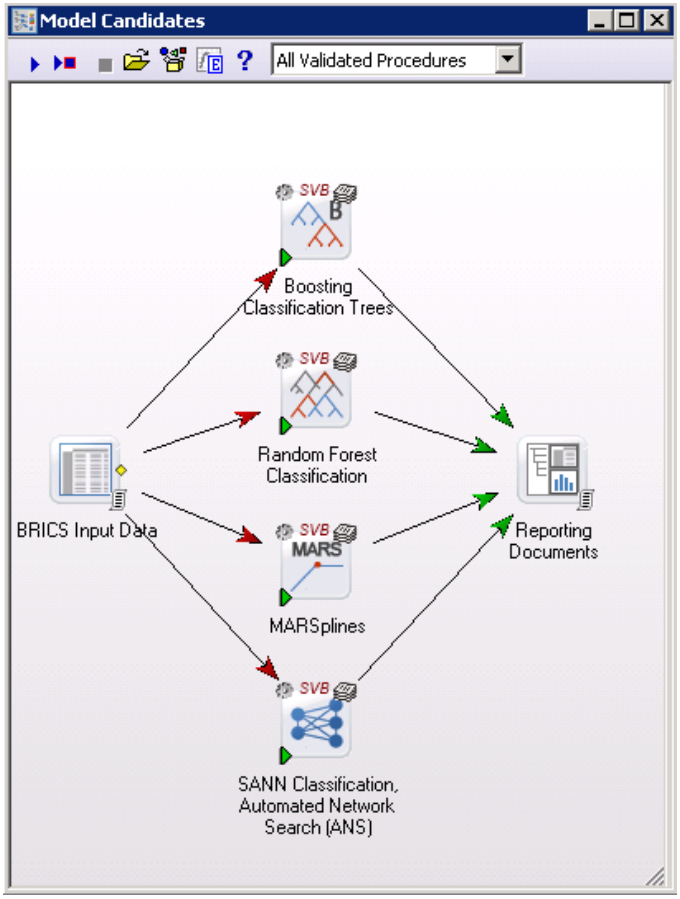
Table 5. Set of predictors used as input for Boostng trees, MARSplines, Random Forsets and Neural networks.

Those lists of predictors were used in all of the four model candidates (except for logistic regression which used the list in table 3). Unlike logistic regression, modern data mining performs usually better when the raw data is used, at least when it comes to natural continuous variables. Categorical variables with a lot of categories, like for instance postcodes, will usually work better when pre-processed. That's why we have used the WOE transformation for categorical variables with a lot of categories. Categorical variables with few categories usually work best in their original form. That's why we use variables like marital status and sex in their original form.

When it comes to parameter settings we did perform some trial and error experimentation and found suitable parameter settings for each of the 4 model candidate models).

After this was completed, it is easy enough to build the models. We used *STATISTICA* to perform the analysis, see [4], and this was done with a few clicks of the mouse.

Fig. 2 User interface of STATISTICA Data Miner.



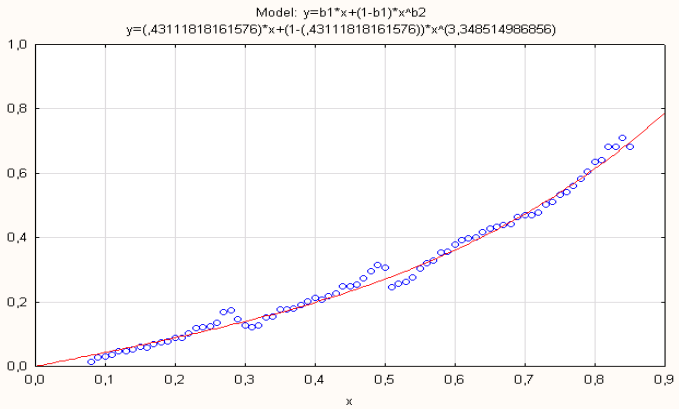
VI. TASK 2: MODELING

The task in 2 was to predict the monthly delinquency for the loans our model would suggest to approve. We solved this task in two steps:

Step 1: Transform the scores produced by our final model to probability scale. This was done using the Nonlinear Estimation procedure. Nonlinear Estimation uses one very efficient general algorithm (quasi-Newton) that approximates the second-order derivatives of the loss function to guide the search for the minimum (i.e., for the best parameter estimates, given the respective loss function). We used this loss function:

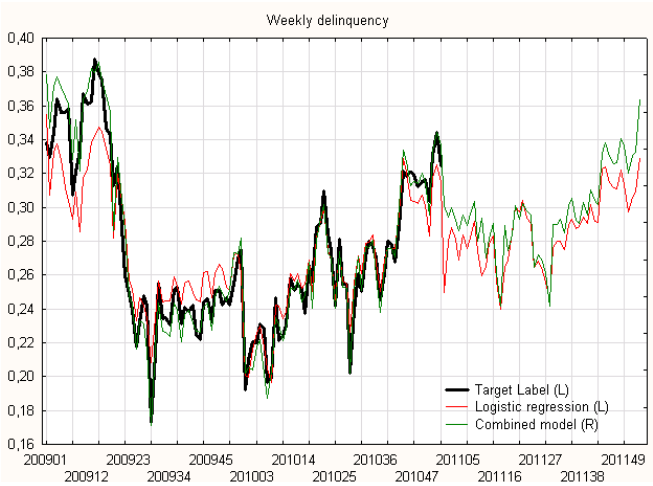
*L=N*abs(obs-pred)*

Fig. 3 Transformation of scores produces by the final model.



After transforming the scores to probability scale, it is easy enough to calculate the mean score value for the applications that are predicted "good" for the prediction dataset. This strategy seemed to work well (since we succeeded to top the leaderboard for task 2).

VII. FINDINGS



It is possible to build fairly good score models using Logistic regression if we allow the WOE transformations to have many categories and do not require monotone transformation. The effect is that the logistic regression model is more complicated (to understand, to communicate and to hard code into other systems), but the performance also improves significantly.

The best results were achieved when we used the 3 steps described above. We used scores from the following models as predictors in Boosting trees: Boosting Trees, MARSplines, Logistic regression (on WOE transformed predictors) and Neural Network. In the second step we used the training sample for test and the test sample for training. When calibrating the final model we added the validation sample (the last three months of 2010) to the training sample.

Our conclusion is that if you have a lot of data, that is, either a lot of records and/or a lot of variables, then you will be able to build score models that perform significantly better using modern data mining techniques such as Boosting Trees, MARSplines, Neural Network than traditional scorecards.

REFERENCES

[1] Breiman, L.: Random Forests, Machine Learning, vol. 45 (1):pp. 5–32, (2001)

[2] Friedman, J. H.: Stochastic gradient boosting, Comput. Stat. Data Anal. 38, 367–378, (2002)

[3] Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning, Springer, (2008)

[4] StatSoft, Inc. (2013). STATISTICA (data analysis software system), version 12.0. www.statsoft.com.

[5] http://www.sidra.ibge.gov.br

[6] http://documentation.statsoft.com/STATISTICAHelp.aspx?path=WeightofEvidence/WeightofEvidenceWoEIntroductoryOverview

[7] Miner, G.; Elder, J., Hill, T., Nisbet, R., Delen, D., Fast, A. (2012) Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications. NY: Elsevier.