Query-based versus tree-based classification: application to banking data

Alexey Masyutin and Yury Kashnitsky

National Research University Higher School of Economics
Moscow, Russia
alexey.masyutin@gmail.com,ykashnitsky@hse.ru

Abstract. The cornerstone of retail banking risk management is the estimation of the expected losses when granting a loan to the borrower. The key driver for loss estimation is probability of default (PD) of the borrower. Assessing PD lies in the area of classification problem. In this paper we apply FCA query-based classification techniques to Kaggle open credit scoring data. We argue that query based classification allows one to achieve higher classification accuracy as compared to applying classical banking models and still to retain interpretability of model results, whereas black-box methods grant better accuracy but diminish interpretability.
Keywords: PD, classification, Kaggle, FCA, credit scoring.

1 Introduction

From the 1960s, banks have started to adopt statistical scoring systems that were trained on datasets of applicants, consisting of their socio-demographic and loan specific features [3]. The aim of those systems was to support decision making whether to grant a loan for an applicant or not. As far as mathematical models are concerned, they were typically logistic regressions run on selected sets of attributes. The target variable was defined as a binary logical value, one if default occurs, zero otherwise. Typical scorecard is built in several steps. The first step is so-called WOE-transformation [1], which transforms all numerical and categorical variables into discrete numerical variables. For continuous variables the procedure, in effect, breaks the initial variable into several ranges, for categorical ones – the procedure regroups the initial categories. The second step is single factor analysis, when significant attributes are selected. The commonly used feature selection method is then based on either information value, or Gini coefficient calculation [1]. With the most predictive factors included into the model, they are further checked for pairwise correlations and multicollinearity. Features with high observed correlation are excluded. As soon as single-factor analysis is over, logistic regression is run taking the selected transformed attributes as input. The product of beta-coefficient and WOE value of the particular category produces the score for that particular category. The sum of variable scores produces the final score for the loan application. Finally, the cutoff score is selected based on the revenue and loss in the historical portfolio. When the scorecard is launched

into work, the loan application immediately receives its score which is compared to the cutoff point. In case the score is lower than cutoff value, the application is rejected, otherwise it is approved. It has to be mentioned that despite its simple mathematical approach scorecards were incredibly attractive for lending institutions for several reasons. First of all, new loan application received score for each of its attributes, which provided clarity: in case of rejection the reason, why the final score was lower than cutoff, can be retrieved. The discriminative power of the models, however, is still at the moderate level. The Gini coefficient for the application scorecards varies from 45% to 55%, and for the behavioral scorecards the range is from 60% to 70% [5]. Apparently, a considerable amount of research was done in the field of alternative machine learning techniques seeking the goal to improve the results of the wide-spread scorecards [7,8,9].

The methods of PD estimation can either produce so-called "black box" models with limited interpretability of model result, or, on the contrary, provide interpretable results and clear model structure. The key feature of risk management practice is that, regardless of the model accuracy, it must not be a black box. That is why methods such as neural networks and SVM classifiers did not earn much trust within banking community [16].

On the contrary, alternative methods such as associative rules and decision trees provide the user with easily interpretable rules which can be applied to the loan application. FCA-based algorithms also belong to the second group since they use concepts in order to classify objects. The intent of the concept can be interpreted as a set of rules that is supported by the extent of the concept. However, for non-binary context the computation of the concepts and their relations can be very time-consuming. In case of credit scoring we deal with numerical context, as soon as categorical variables can be transformed into set of dummy variables. Lazy classification [15] seems to be appropriate to use in this case since it provides the decision maker with the set of rules for the loan application and can be easily parallelized.

In this paper, we test query-based classification framework on Kaggle open data contest.[1] The contest was held in 2011 and provided credit scoring data to test different classification algorithms. We compare results of query-based classification with classical methods adopted in banks and black-box methods. We argue that query-based classification allows one to achieve higher accuracy than classical methods and still to retain interpretability of model results, whereas black-box methods grant better accuracy but diminish interpretability.

2 Main Definitions

First, we recall some standard definitions related to Formal Concept Analysis, see e.g. [10].

Let G be a set (of objects), let (D, $\sqcap$) be a meet-semi-lattice (of all possible object descriptions) and let δ: G $\rightarrow$ D be a mapping. Then (G, $\underline{D}$,δ), where

[1] `https://www.kaggle.com/c/GiveMeSomeCredit`

$\underline{D}$ =(D, $\sqcap$), is called a *pattern structure* [11], provided that the set
δ(G) := $\{\delta(g)|g \in G\}$ generates a complete subsemilattice $(D_\delta, \sqcap)$ of (D, $\sqcap$), i.e., every subset X of $\delta(G)$ has an infimum $\sqcap$X in (D, $\sqcap$). Elements of D are called *patterns* and are naturally ordered by *subsumption* relation $\sqsubseteq$:
given $c, d \in D$ one has $c \sqsubseteq d \leftrightarrow c \sqcap d = c$. Operation $\sqcap$ is also called a *similarity operation*. A pattern structure $(G,\ \underline{D},\ \delta)$ gives rise to the following *derivation operators* $(\cdot)^\diamond$:

$$A^\diamond = \prod_{g \in A} \delta(g) \qquad \text{for } A \in G,$$
$$d^\diamond = \{g \in G \mid d \sqsubseteq \delta(g)\} \qquad \text{for } d \in (D,\ \sqcap).$$

These operators form a Galois connection between the powerset of G and $(D, \sqcap)$. The pairs (A, d) satisfying A $\subseteq$ G, d $\in$ D, $A^\diamond$ = d, and A = $d^\diamond$ are called *pattern concepts* of (G,$\underline{D}$, δ), with *pattern extent* A and *pattern intent* d. Operator $(\cdot)^{\diamond\diamond}$ is an algebraical closure operator on patterns, since it is idempotent, extensive, and monotone [10].

The concept-based learning model for standard object-attribute representation (i.e., formal contexts) is naturally extended to pattern structures. Suppose we have a set of positive examples G_+ and a set of negative examples G_- w.r.t. a target attribute, $G_+ \cap G_- = \emptyset$, objects from
$G_\tau = G \setminus (G_+ \cup G_-)$ are called undetermined examples. A pattern $c \in D$ is an α - weak positive premise (classifier) iff:

$$\frac{|c^\diamond \cap G_-|}{|G_-|} \leq \alpha \ and \ \exists A \subseteq G_+ : c \sqsubseteq A^\diamond$$

A pattern $h \in D$ is an α - weak positive premise iff:

$$\frac{|h^\diamond \cap G_-|}{|G_-|} \leq \alpha \ and \ \exists A \subseteq G_+ : h = A^\diamond$$

In case of credit scoring we work with pattern structures on intervals as soon as a typical object-attribute data table is not binary, but has many-valued attributes. Instead of binarizing (scaling) data, one can directly work with many-valued attributes by applying interval pattern structure [18]. For two intervals $[a_1, b_1]$ and $[a_2, b_2]$, with $a_1, b_1, a_2, b_2 \in \mathbb{R}$ the *meet operation* is defined as [14]:
$[a_1, b_1] \sqcap [a_2, b_2] = [min(a_1, a_2), max(b_1, b_2)]$.

The original setting for lazy classification with pattern structures can be found in [12,13].

3 Loan Default Prediction in Banking: Scorecards

The event of default in retail banking is defined as more than 90 days of delinquency within the first 12 months after the loan origination. Defaults are

divided into fraudulent cases and ordinary defaults. The default is told to be a fraudulent case when delinquency starts at one of the three first months. It means that when submitting a credit application, the borrower did not even intend to pay back. Otherwise, the default is ordinary when the delinquency starts after the first three months on book. That is why scorecards are usually divided into fraud and application scorecards. In fact the only difference is the target variable definition, while the sets of predictors and the data mining techniques remain the same. The default cases are said to be "bad", and the non-default cases are said to be "good". Banks and credit organizations have been traditionally using scorecards to predict whether a loan applicant is going to be bad or good.

Mathematical architecture of scorecards is based on a logistic regression, which takes the transformed variables as an input. The transformation of the initial variables is known as WOE-transformation [1]. It is wide-spread in credit scoring to apply such a transformation to the input variables as soon as it accounts for non-linear dependencies and provides certain robustness coping with potential outliers. The aim of the transformation is to divide each variable into no more than k categories. At step 0, all the continuous variables are binned into 20 quantiles, the nominal and ordinal variables are either left untouched or are one-hot encoded. Now, when all the variables are categorized, the odds ratio is computed for each category.

$$odds_{ij} = \frac{\%goods_{ij}}{\%bads_{ij}}$$

Then for each predictor variable X_i $(i = 1...n)$ non-significant categories are merged. Significance is measured by standard chi-square test for differences in odds with p-value threshold up to 10%. So, for each feature the following steps are done:

1. If X_i has 1 category only, stop and set the adjusted p-value to be 1.
2. If X_i has k categories, go to step 7.
3. Else, find the allowable pair of categories of (an allowable pair of categories for ordinal predictor is two adjacent categories, and for nominal predictor is any two categories) that is least significantly different (i.e. most similar) in terms of odds. The most similar pair is the pair whose test statistic gives the largest p-value with respect to the dependent variable Y.
4. For the pair having the largest p-value, check if its p-value is larger than a user-specified alpha-level merge. If it does, this pair is merged into a single compound category. Then a new set of categories of is formed. If it does not, then if the number of categories is less or equal to user-specified minimum segment size, go to step 6, else merge two categories with highest p-value.
5. Go to step 2.
6. (Optional) Any category having too few observations (as compared with a user-specified minimum segment size) is merged with the most similar other category as measured by the largest of the p-values.
7. The adjusted p-value is computed for the merged categories by applying Bonferroni adjustments [2]. Having accomplished the merging steps, we acquire categorized variables instead of the continuous ones.

When each variable X_i $(i = 1...n)$ is binned into a certain number of categories (k_i), one is able to calculate the odds for each category j $(j = 1...k_i)$, the weight of evidence for each category.

$$WOE_{ij} = ln(odds_{ij})$$

The role of the WOE-transformation is that, instead of initial variables, logistic regression receives WOE features as input. So, each input variable is a discrete transformed variable, which takes values of WOE. When estimating the logistic regression, the usual maximum likelihood is applied.

4 Query-Based Classification Algorithm

Query-based classification is in effect an approach proposed in [17] with certain voting scheme applied to predict the test object class (positive or negative). The idea behind the algorithm is to check whether it is positive or negative context that test object is more similar to. The similarity is defined as a total support of α - weak positive (negative) premises that contain the description of test object. The algorithm uses three parameters: subsample size, number of iterations and alpha-threshold. The first parameter is expressed as percentage of the observations in the context. At each step the subsample is extracted and the descriptions of the objects in subsample are intersected with the description of test object. As subsample size grows, the resulting intersection $\delta(g_1)\sqcap...\sqcap\delta(g_k)\sqcap\delta(g)$ becomes more generic and it is more frequently falsified by the objects from the opposite context. We randomly take the chosen number of objects from positive (negative) context as candidates for intersection with the test object. The number of times (i.e. number of iterations) we randomly extract a subsample from the context is the second parameter of the algorithm, which is also tuned through grid search. Intuition says, the higher the value of the parameter the more premises should be mined from the data. However, the obvious penalty for increasing the value of this parameter is time required for computing intersections. As we mentioned, the greater the subsample size, the more it is likely that $(\delta(g_1) \sqcap \ldots \sqcap \delta(g_k) \sqcap \delta(g))^{\diamond}$ contains the object of the opposite class. In order to control this issue, we add third parameter which is alpha-threshold. If the percentage of objects from the positive (negative) context that falsify the premise $\delta(g_1) \sqcap \ldots \sqcap \delta(g_k) \sqcap \delta(g)$ is greater than alpha-threshold of this context than the premise will be considered as falsified, otherwise the premise will be α-weak and, thereafter, used in classification of the test object. These steps are performed for each test object for positive and negative contexts seprately, producing a set of positive and negative α-weak premises. The final output for the test object we used was a difference between the total number of objects from positive context supporting the set of positive premises and the total number of objects from negative context supporting the set of negative premises.

5 Data and Experiments

We decided to retrieve open dataset devoted to the credit scoring. We considered the "Give Me Some Credit" contest held in 2012[2]. The data has a binary target variable (class label) whether the borrower defaulted or not. However, it is not specified whether the default event was ordinary or fraudulent. We develop a scorecard and examine its accuracy via out-of-sample validation with provided target variable. The validation process requires calculation of performance metrics (ROC AUC and Gini coefficient) of the model based on the data sample that was retrieved from the same distribution but was not used to develop the model itself. This approach allows the user to check for accuracy and stability of the model. In order to train the models we extracted 1000 good loans and 1000 bad loans. The size of the validation set was 300 observations. All these observations were randomly extracted from the contest dataset. Our aim was to compare classical scorecard versus black-box models such as boosting versus query-based classification approach based on interval patterns. We implemented the query-based classification algorithm using R, which is a flexible tool for statistical analysis. The R language is becoming more recognizable in the banking sphere as well. The features for loan default prediction are presented in Table 1:

Table 1. Kaggle Data Description

Variable Name	**Description**	**Type**
SeriousDlqin2yrs	Person experienced 90 days past due delinquency or worse	Y/N
RevolvingUtilizationOfUnsecuredLines	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits	percentage
age	Age of borrower in years	integer
NumberOfTime30-59DaysPastDueNotWorse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years.	integer
DebtRatio	Monthly debt payments, alimony, living costs divided by monthly gross income	percentage
MonthlyIncome	Monthly income	real
NumberOfOpenCreditLinesAndLoans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)	integer
NumberOfTimes90DaysLate	Number of times borrower has been 90 days or more past due.	integer
NumberRealEstateLoansOrLines	Number of mortgage and real estate loans including home equity lines of credit	integer
NumberOfTime60-89DaysPastDueNotWorse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years.	integer
NumberOfDependents	Number of dependents in family excluding themselves (spouse, children etc.)	integer

First, we concluded that the variable distributions might be not very appropriate for applying trees-like transformations. The values of features are evenly distributed across wide ranges both for good and bad loans, therefore applying cutpoint does not perform well to distinguish among loan applicants. Examples of such distributions are presented below:

[2] https://www.kaggle.com/c/GiveMeSomeCredit

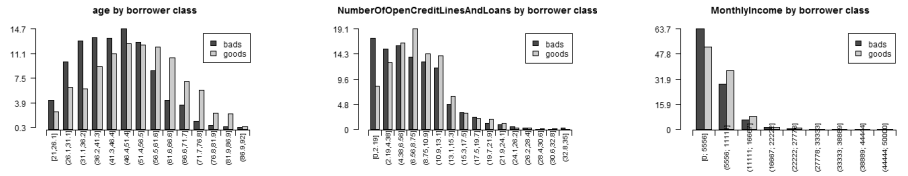

Fig. 1. Age distribution by goods and bads (left), number of open credit lines and loans by goods and bads (middle), and monthly applicant income by goods and bads (right)

In order to build scorecard we applied WOE-transformation to the variables (using rpart and smbinning packages in R) on training sample. The WOE-transformation was controlled for maximum number of observations in the final nodes of one-factor trees in order to escape overfitting at the starting point. Therefore, variables were binned into two to four categories. The examples of variable binning are provided in Fig. 2:

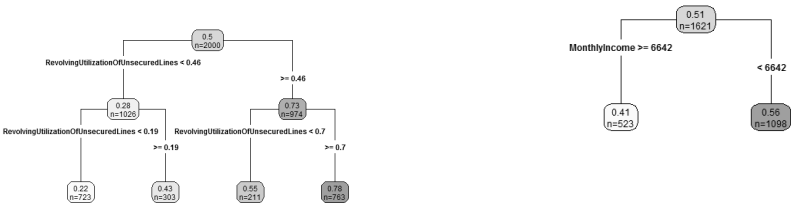

Fig. 2. One-Factor Trees for WOE-transformation of Revolving Utilization of Unsecured Lines (left) and Monthly Income (right)

As soon as we have transformed the factors, the individual Gini coefficients were calculated to assess the predictive power of the coefficients. We excluded variables that have shown dramatic drop in Gini on validation sample. The rest were fed to logistic regression and the final model included the features presented in Table 2.

Table 2. Logistic Regression Output

Feature	Estimate	Std. Error	t-stat	P-value
(Intercept)	-0.56881	0.05002	-11.371	<2e-16 ***
trscr_RevolvingUtilizationOfUnsecuredLines	0.73361	0.04317	16.992	<2e-16 ***
trscr_age	0.39750	0.08257	4.814	1.59e-06 ***
trscr_NumberOfTime3059DaysPastDueNotWorse	0.55770	0.05593	9.971	<2e-16 ***
trscr_NumberOfTime6089DaysPastDueNotWorse	0.44882	0.06373	7.043	2.58e-12 ***

After training the scorecard we applied query-based classification to the validation set. The algorithm QBCA (for "Query-Based Classification Algorithm") defines number of iterations, *alpha*-level and *subsample-size* parameters upon algorithm tuning. Finally, the algorithms were compared on a validation set by plotting ROC curves and calculating Gini coefficients achieved.

Table 3. Experimental results: cross-validation and validation Gini coefficients for 3 models. "Scorecard" stands for logistic regression with WOE-transformed features, and "QBCA" designates the query-based classification algorithm

metric \algo	**Scorecard**	**QBCA**	**Xgboost**
Valid. Gini	0.5806	0.6624	0.708

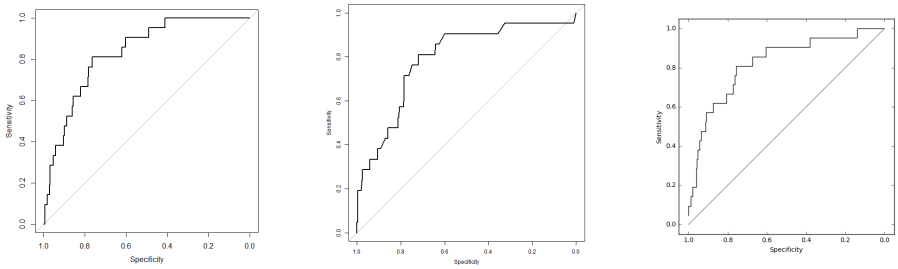

Fig. 3. ROC curves for QBCA (left), Scorecard (middle) and Xgboost (right)

Finally, we applied the Xgboost[3] gradient boosting algorithm to the same data to estimate the classification accuracy achievable with the "black-box" model. The parameters were tuned via 5-fold stratified cross-validation. The results (cross-validation and validation Gini) for 3 tested algorithms are given in Table 3. The ROC curves for validation set are presented in Fig. 3. As we can see, Xgboost performs best in terms of Gini. However, its results are not

[3] https://github.com/dmlc/xgboost

interpretable, and the best explanation for classification that we one can extract from the trained Xgboost model is the estimated feature importance, based on the number of times splits in trees were done with each feature.

On the contrary, it is interesting to realize that certain patterns can be extracted from the QBCA model. We can observe rules such as if a loan applicant's age is greater than 50 and there was no delinquency in the past and the overall revolving utilization of unsecured lines was less than 11%, then the probability of default is almost 4 times lower than average. On the other side applicants younger than 30 and having revolving utilization of unsecured lines greater than 72% will default 1.5 times more frequent than on average. This is where we enjoy the advantage of interval pattern structures: they represent the rules that can be easily interpreted, and at the same time they make prediction for each new object in validation dataset individually, which allows to improve classification accuracy over the default scorecard model.

6 Conclusion

We considered three approaches to modeling probability of default in the problem of credit scoring. All approaches were tested on the random sample from Kaggle dataset. The first was testing classical methods of scorecard, which is easily interpretable but provides limited predictive accuracy. The second, was query-based classification algorithm on interval pattern structures, which provides higher predictive performance, and still keeps the interpretability clear. The third, was a black-box algorithm represented by Xgboost, which showed best predictive ability but nevertheless did not allow one to extract interesting client insights from the data. Therefore, we argue that FCA based classification algorithms can compete with ordinary statistical instruments adopted in banks and still provide the sets of rules which were relevant for particular loan applicant.

Acknowledgments

The paper was prepared within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE) and supported within the framework of a subsidy by the Russian Academic Excellence Project '5-100'.

References

1. Bigss, D., Ville, B., and Suen, E. A Method of Choosing Multiway Partitions for Classification and Decision Trees. Journal of Applied Statistics. Vol. 18, No. 1, pp. 49-62, 1991
2. Bonferroni, C. E. Teoria statistica delle classi e calcolo delle probabilità. Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze. Vol. 8, pp. 3-62, 1936
3. Naeem S. Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring, SAS Publishing, 2005
4. Thomas, L. C., Edelman, D. B., and Crook, J. N. Credit Scoring and Its Applications. Pliladelphia.: SIAM, 2002
5. Baesens B., Gestel T.V., Viaene S., Stepanova M., and Suykens J. Benchmarking state-of-the-art classification algorithms for credit scoring, Journal of the Operational Research Society, Vol. 54 (6), pp. 627-635, 2003
6. Silvia, F., and Cribari-Neto, F. Beta Regression for Modelling Rates and Proportions. Journal of Applied Statistics. Vol. 31, No. 7, pp. 799–815, 2004
7. Yu, L., Wang, S. and Lai, K. K. An intelligent agent-based fuzzy group decision making model for financial multicriteria decision support: the case of credit scoring. European journal of operational research. Vol. 195, pp. 942-959, 2009
8. Gestel, T. V., Baesens, B., Suykens, J. A., Van den Poel, D., Baestaens, D. E., and Willekens, B. Bayesian kernel based classification for financial distress detection. European journal of operational research. Vol. 172, pp. 979-1003, 2006
9. Kumar P. R., and Ravi V. Bankruptcy Prediction in Banks and Firms via Statistical and Intelligent Techniques – A Review. European Journal of Operational Research, Vol. 180, No. 1, pp. 1-28, 2007
10. Ganter, B. and Wille, R. Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc, 1997
11. Ganter, B., and Kuznetsov, S. O., Pattern Structures and Their Projections. In: G. Stumme and H. Delugach, Eds., 9th International Conference on Conceptual Structures, LNAI (Springer), Vol. 2120, pp. 129-142, 2001.
12. Kuznetsov, S. O. Scalable Knowledge Discovery in Complex Data with Pattern Structures. In: Proc. 5th International Conference Pattern Recognition and Machine Intelligence (PReMI'2013), Lecture Notes in Computer Science (Springer), Vol. 8251, pp. 30-41, 2013
13. Kuznetsov, S. O. Fitting Pattern Structures to Knowledge Discovery in Big Data. Proc. of the 11th International Conference on Formal Concept Analysis (ICFCA 2013). Lecture Notes in Computer Science, Springer. Vol. 7880. p 254-266, 2013
14. Kaytoue, M., Duplessis, S., Kuznetsov, S. O, and Napoli, A.. Mining Gene Expression Data with Pattern Structures in Formal Concept Analysis. Information Sciences. Spec.Iss.: Lattices, 2011
15. Aha. D.W. (Ed.). Lazy Learning. Kluwer Academic Publishers, 1997
16. Li, X. and Zhong Y.. An Overview of Personal Credit Scoring: Techniques and Future Work. Intl Journal of Intelligence Science. Vol. 2, No. 4A. pp. 182–189, 2012
17. Masyutin, A., Kashnitsky, Y., and Kuznetsov S. O. Lazy Classification with Interval Pattern Structures: Application to Credit Scoring. Proceedings of the International Workshop "What can FCA do for Artificial Intelligence?" (FCA4AI at IJCAI 2015). Ed.: Sergei O. Kuznetsov, A. Napoli, S. Rudolph. Buenos Aires. pp. 43-54, 2015
18. Kaytoue, M., Kuznetsov S. O., and Napoli, A. Revisiting numerical pattern mining with formal concept analysis. IJCAI 2011, pp. 1342-1347, 2011