Bank of Russia

THE FINER POINTS OF MODEL COMPARISON IN MACHINE LEARNING: FORECASTING BASED ON RUSSIAN BANKS' DATA

No. 43 / August 2019

WORKING PAPER SERIES

Denis Shibitov, Mariam Mamedli

Denis Shibitov

Bank of Russia. Email: ShibitovDS@cbr.ru

Mariam Mamedli

Bank of Russia. Email: MamedliMO@cbr.ru

The authors are grateful to Sergei Seleznev (Bank of Russia) and the participants of the Research and Forecasting Department Research Seminar for their comments and suggestions.

The Bank of Russia Working Paper Series is anonymously refereed by members of the Bank of Russia Research Advisory Board and external reviewers.

All rights reserved. The views expressed in this paper are solely those of the authors and do not necessarily reflect the official position of the Bank of Russia. The Bank of Russia assumes no responsibility for the contents of the paper. Any reproduction of these materials is permitted only with the express consent of the authors.

Cover image: Shutterstock.com

Address: 12 Neglinnaya street, Moscow, 107016
Tel.: +7 495 771-91-00, +7 495 621-64-65 (fax)
Website: www.cbr.ru

Contents

Abstract

We evaluate the forecasting ability of machine learning models to predict bank license withdrawal and the violation of statutory capital and liquidity requirements (capital adequacy ratio N1.0, common equity Tier 1 adequacy ratio N1.1, Tier 1 capital adequacy ratio N1.2, N2 instant and N3 current liquidity). On the basis of 35 series from the accounting reports of Russian banks, we form two data sets of 69 and 721 variables and use them to build random forest and gradient boosting models along with neural networks and a stacking model for different forecasting horizons (1, 2, 3, 6, 9 months). Based on the data from February 2014 to October 2018 we show that these models with fine-tuned architectures can successfully compete with logistic regression usually applied for this task. Stacking and random forest generally have the best forecasting performance comparing to the other models. We evaluate models with commonly used performance metrics (ROC-AUC and F1) and show that, depending on the task, F1-score could be better at defining the model's performance. Comparison of the results depending on the metrics applied and types of cross-validation used illustrate the importance of choosing the appropriate metric for performance evaluation and the cross-validation procedure, which accounts for the characteristics of the data set and the task under consideration. The developed approach shows the advantages of non-linear methods for bank regulation tasks and provides the guidelines for the application of machine learning algorithms to these tasks.

Keywords: machine learning, random forest, neural networks, gradient boosting, forecasting, bank supervision

JEL: C45, C53, C52, C5.

Introduction

Since 2013 the Bank of Russia has moved toward an active policy aimed at the enhancement of the financial stability of the banking sector. It has only increased the interest in forecasting the withdrawal of bank licenses and fuelled further analysis of this subject.

A wide range of research is devoted to the analysis of the Russian banking sector, namely, to the forecasting of the bank license withdrawal and to the determination of the key factors affecting a bank's efficiency and the possible interruption of its activity. The research on license withdrawal differs among others in the time period analysed, whether or not the distinction is made between the reasons of license withdrawal or the macroeconomic variables are included in the model or not. However, the analysis is usually based on logistic regression (logit-model) estimated on the quarterly data. This methodology is widely applied in the analysis of Russian bank defaults and license withdrawals (Styrin (2005), Lanine, Vennet (2006), Peresetsky et al. (2011)). Sinelnikova-Muryleva et al. (2018) is one of the first attempts to apply machine learning models to Russian banks' data on license withdrawal. They forecast bank defaults with logit-models and random forest with the predefined architecture based on quarterly data, underlining the potential of the random forest model to compete with the logit-model.

We contribute to the literature by applying several machine learning algorithms with the optimal architecture to the forecasting of bank license withdrawal and violation of regulatory banking requirements to Russian banks accounting data. We consider several machine learning models, namely, random forest, gradient boosting, neural networks and logit-model widely used in previous research on license withdrawal. We define the optimal model architectures to forecast bank license withdrawal (both the case of actual license withdrawal and bank liquidation) and the violation of the five requirements: capital adequacy ratio (N1.0)[1], common equity Tier 1 adequacy ratio (N1.1)[2], Tier 1 capital adequacy ratio (N1.2)[3], instant and current liquidity requirements (N2 and N3). In both tasks we additionally develop an ensemble based on the models with the best forecasting properties. We compare the results of machine learning models and the logit-model with the optimal architecture. The optimal architecture is chosen for the forecasting horizons of one, two, three months, half of

[1] Capital adequacy ratio (H1.0) represents ratio of own funds (capital) of credit institutions to total risk-weighted assets.

[2] Common equity Tier 1 adequacy is a ratio of common equity (capital) of credit institutions to total risk-weighted assets.

[3] Tier 1 capital adequacy (H1.2) represents ratio of core equity (capital) of credit institutions to total risk-weighted assets.

year and 9 months based on the monthly data from February 2014 to October 2018. We consider two datasets, containing 69 and 721 variables, based on 35 publicly available indicators from bank accounts.[4]

One of the key differences of this research from the others devoted to the forecasting of bank license withdrawal is that we apply several machine learning algorithms, allowing us to incorporate the nonlinear relationship between variables and to define the optimal architecture for the model of each type and each forecasting horizon. With the aim of illustrating the application of machine learning techniques to economic forecasting tasks, we compare the performance of these algorithms with the optimal set of hyperparameters to the logistic regression usually used in the forecasting of the license withdrawal. To the best of our knowledge this is the first comparison of such machine learning techniques as random forest, gradient boosting, neural networks and stacking and their application to the license withdrawal task and regulatory requirements violation on the Russian banks' microdata. Moreover, the forecasting of the violation of bank requirements has not been previously considered in the literature. In contrast to most of the research, estimation is based on the monthly data, with the optimal model architecture defined for horizons from 1 to 9 months. Particular attention is paid to the proper choice of the performance metric and to the impact of the data splitting type during cross-validation on the final results. Incorrect data splitting methods can lead to less-than-optimal choice of the model architecture for forecasting. During data splitting into training and testing samples the chronological order of observations is also important: random splitting can lead to factitiously high metric values.

In the case of license withdrawal all considered models have the comparable forecasting accuracy. At the same time, the application of machine learning techniques and the choice of optimal architecture allow us to obtain more accurate forecasts than those based on the logit-models used in previous research. In the forecasting of the violation of capital adequacy ratios several nonlinear models have higher forecasting performance than the logit-model. Namely, the random forest model has the highest accuracy for most forecasting horizons. In the forecasting of instant and current liquidity ratios all considered models have a comparable accuracy inside-the-margin-of-error. This accuracy is lower than the one in the forecasting of capital ratios.

Regarding the example of Tier 1 capital adequacy ratio we show how the procedure of data splitting on cross-validation affects the results: performance metrics on cross-validation,

[4] The accounting forms are available on the Bank of Russia website in the corresponding section.

and thus, the choice of the optimal model highly depends on the splitting methodology, which can significantly affect the forecasting performance of models. We also check the robustness of the results on the dataset of 721 variables formed on the basis of the previously used 69 variables. This data extension does not qualitatively increase the forecasting performance of the considered models.

This paper has the following structure. The next section gives a brief overview of the literature of the forecasting of license withdrawal based on the data of Russian banks and several papers on the US data related to current research methodologically. The following section provides details on the estimated models and key hyperparameters used in the search for the optimal model architecture in forecasting license withdrawal and violation of requirements. Section 3 covers the description of the data, forecasting performance metrics and the cross-validation. Section 4 provides the results of model estimation and the model choice in both forecasting tasks. The emphasis is made on the proper choice of performance metrics. The summary of the results is provided in the Conclusion.

1. Literature review

A variety of foreign and domestic research on banking data is devoted to the forecasting of license withdrawal and the reasons for license withdrawal. Karas et al. (2010) use Russian bank data to analyse the link between the bank's efficiency and the property type. It is followed by Belousova et al. (2018) who also consider how the property type influences the efficiency of Russian banks during 2004-2015 based on quarterly data. Claeys et al. (2005) and Claeys, Schoors (2007) define which factors affect the license withdrawal of Russian banks. Fungáčová, Solanko (2009) analyse the link between bank characteristics and the degree of risk-taking on the basis of quarterly data from 1999 to 2007.

Peresetsky et al. (2011) and Peresetsky (2007, 2012) apply the binary choice model to the forecasting of a bank default considered as a license withdrawal. Their research is followed by Peresetsky (2013), who makes the distinction between different reasons for license withdrawal from the second quarter of 2005 to the end of 2008. The probability of license withdrawal is forecasted one year in advance on the basis of macro and financial time-series. With a logit-model Peresetsky (2013) forecasts the probability of license withdrawal, license withdrawal due to money laundering, due to both money laundering and economic reasons and the withdrawal only due to economic reasons (economic inconsistency). For each of these reasons Peresetsky (2013) estimates three models on different datasets: the model with both macro and micro variables, and two models which include either micro or macro variables. He also estimates the model of multiple choice for different reasons of license withdrawal, showing that it does not provide higher forecast accuracy than the binary choice model. To deal with an unbalanced dataset Peresetsky (2013) uses the subsampling method similar to the one proposed by Peresetsky (2007).

Another approach to solve the problem of the unbalanced datasets is applied by Emelyanov, Briukhova (2013). The authors estimate the logit-model in order to forecast the probability of the default of Russian commercial banks on the basis of the monthly accounting data from 2010 to 2011. In the following research Emelyanov, Briukhova (2015) underline the importance of addressing the unbalanced dataset problem which occurs when dealing with banking data (the low share of license withdrawal) and the need to choose the optimal structure of the subsample. They show that when this problem is solved the model forecasting accuracy increases on the basis of 12 financial variables for the horizon from 1 to 8 months in advance. They solve it by a bootstrap method, including only part of the banks that did not lose their license and all the banks whose license was withdrawn in the dataset.[5] After that, they choose the share of licenced banks in the whole sample, considering, however, only the licenses withdrawn due to financial reasons.

On the basis of the data from 1998 to 2011, Karminsky, Kostrov (2013) illustrate the existence of a quadratic relationship between the probability of a bank default and its size, capital adequacy and profitability. Additionally, they show that accounting for macroeconomic variables and time factor helps to increase substantially the model's accuracy. Karminsky, Kostrov (2013) determine the negative relationship between the default probability and monopolistic power. This result coincides with Fungacova, Weill (2009) who analyse on the basis of the logit-model how the market power affects the license withdrawal of Russian banks based on the quarterly data from 2001 to 2007. Their result is robust to the use of different measures of the market power and the definition of bank failure. Fungacova, Weill (2009) also illustrate the negative impact of the bank size on the bank failure, which is statistically significant in most specifications confirming the results of Claeys et al. (2005) and Claeys, Schoors (2007). In their recent paper Karminsky, Kostrov (2017) estimate a logit-model to forecast bank defaults with negative capital based on the quarterly data from 2010 to the first half of 2015. In order to deal with the unbalanced sample, the authors modify the maximum likelihood function.

[5] Review of the solutions for unbalanced sample problem is provided in e.g. He, Garcia (2008), Ganganwar (2012), Sonak, Patankar (2015).

Bidzhoyan, Bodganova (2017) also use both macro- and financial annual time-series of banks' activity in the forecasting with the logit-model the probability of license withdrawal faced by Russian banks. They show that the inclusion of median, standard deviation and dispersion of macro variables increases the accuracy of the forecast compared to the model with an average exchange rate and other macro variables. Nevertheless, this improvement in the sensitivity is not significant, while the total accuracy rises by 0.77% (from 69.97% to 70.74%). Bidzhoyan (2018) also estimates the logit-model on 43 variables including the characteristics of macro variables volatility based on quarterly data of Russian banks for the period from 2012 to 2016. He uses the RIDGE logit-model as a way to deal with multicollinearity in economic and financial data. The total accuracy of the model is 77.8% while the correctly forecasted license withdrawals on the test set is only 66.2%, presumably because only economic reasons are included in the analysis.

One of the latest analyses of Russian banks data is Mäkinen, Solanko (2017) (Mäkinen, Solanko (2018)). They analyse the role of CAMEL variables in explaining the closure of Russian banks and estimate the logit-model, defining the key factors in license withdrawal of Russian banks based on the monthly data from July 2013 to July 2017. They show that the change in CAMEL variables (capital, asset quality, management, earnings and liquidity) are always significant in explaining the license withdrawal, and become less important in the longer lags.[6] At the same time the liquidity level is the only factor which remains important in the lags longer than one month.

The other domain of the research of the Russian banking sector is the modelling of bank credit ratings using ordered choice models (Soest et al. (2003), Peresetsky (2009), Peresetsky, Karminsky (2007, 2011), Vasiluk, Karminsky (2011)). One of the most recent papers is by Peresetsky, Zhivaikina (2017) who analyse the link between bank credit ratings and the possibility of license withdrawal of Russian banks. On the basis of quarterly data from 2012 to 2016 they define the agencies (S&P, Moody's, "Expert RA") whose ratings enable better forecast license withdrawal. They, however, show that models of binary choice estimated to project license withdrawal have the higher forecasting accuracy comparing to the models based on ratings.

It can be seen that logistic regression is the most widely used model in the research devoted to the forecasting of licence withdrawal. There are significantly fewer cases of the application of machine learning models to the forecasting tasks based on banking data.

[6] Abbreviation CAMEL stands for capital, asset quality, management, earnings and liquidity.

Sinelnikova-Muryleva et al. (2018) is one of the first attempts to apply machine learning models to Russian bank data on license withdrawal. They forecast bank defaults with logit-models and random forest with the predefined architecture based on the quarterly data from 2015 to the first quarter of 2017. According to their results, the random forest model provides an extremely low error (MAE level) outperforming the logit-model, with a small difference in models errors. However, these results can be explained by a very short test sample (1 quarter) with only four forecasted events occurring and the increase in the model accuracy compared to the logit-model coming entirely from better predictions of the major class (no default) at the expense of better default forecasting. Nevertheless, Sinelnikova-Muryleva et al. (2018) show the potential gains which can come from the use of machine learning techniques. The most relevant to the current research in the application of a wider range of machine learning techniques are Mai, Baek (2012) and Petropoulos et al. (2017) who forecast bank bankruptcy in the USA and Bagherpour (2017) who analyses mortgage loan defaults.

Mai, Baek (2012) forecast bankruptcy of US banks with quarterly data on financial accounts and economic variables from 2002 to 2011. They apply logit-model and support vector machines (SVM) with different kernel function specifications, using the cross-validation to choose the optimal set of features, included in both the logit-model and SVM model. As opposed to the logit-model with close levels of recall and precision on test and cross-validation, the SVM model with polynomial kernel has a higher forecasting performance on the test set with lower levels of recall and precision than those of the logit-model. Mai, Baek (2012) suggest that searching for the optimal set of features specifically for the SVM model can improve the results. They also analyse the errors which may occur in forecasting, defining the errors linked to forecasting of bank default "prematurely", when the model predicts bankruptcy one quarter earlier than it actually occurs. Mai, Baek (2012) explain it by the close values of accounting variables of the bank in bankruptcy and one quarter prior to the event. The forecasting of US bank defaults is also analysed by Petropoulos et al. (2017). They consider a wide range of models and compare their performance based on the data from 2008 to 2014. The analysis includes logit-model and linear discriminant analysis as well as such machine learning algorithms as SVM, neural networks and random forest. They show that random forest has the best forecasting performance among the considered models. According to their results CAMELS variables on earnings and capital are the most important in bankruptcy forecasting compared to the others.

Bagherpour (2017) also compares the performance of different machine learning algorithms (factorization machines, K-nearest neighbours, random forest and SVM) with the logit-model. He considers the other binary classification problem, the prediction of defaults on mortgage loans, based on three data periods: prior to the financial crisis (2000-2006), during the financial crisis (2007-2011) and afterwards (2012-2015). Bagherpour (2017) shows that all machine learning models have a higher forecasting accuracy than the logit-model: factorization machines have the best forecasting performance (88-91% ROC-AUC), followed by random forest and K-nearest neighbours (88% ROC-AUC on average for the whole data sample), outperforming logit-model (ROC-AUC – 85%).

2. Models

Here we consider two binary classification problems: the prediction of bank license withdrawal and the violation of requirements for capital and liquidity by Russian banks. The target variable equals 1 in the case of a license withdrawal (both the case of the actual license withdrawal and liquidation) or a requirement violation, and -1 otherwise. In both forecasting tasks we make predictions for 1, 2, 3, 6 and 9 months in advance. We consider the violation of five requirements: capital adequacy ratio (N1.0), common equity Tier 1 adequacy ratio (N1.1), Tier 1 common equity adequacy ratio (N1.2), instant (N2) and current (N3) liquidity. Below we present four separate models used in the forecasting, namely, logistic regression (Section 2.1), random forest (Section 2.2), gradient boosting (Section 2.3), neural networks (Subsection 2.4) and the ensemble of the best performing models, stacking (Section 2.5).

2.1. Logistic regression

Logistic regression is a linear model, widely used in classification tasks, including the forecasting of bank license withdrawal. One of the key advantages of the logit-model is its interpretability: it is usually possible to define how the change in the explaining variables affects the probability of a particular class. Moreover, taking into account the realization simplicity of this algorithm, the logit-model can be easily applied to data of a big size with modest requirements on computational capability. At the same time, one of the key drawbacks of this model is its low forecasting performance in case of a nonlinear relationship between the target variable and factors.

Here we use logistic regression with L_2 regularization with different regularization levels and solution methods (Table 1). For a binary classification the following minimization problem is solved:

$$L = \min_{\omega,c}\left(\frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}\log(\exp(-y_i(X_i^T\omega + c)) + 1)\right), \tag{1}$$

where ω are weights, C stands for an inverse of regularization strength, X^T are input data and y is a target variable which takes values of 1 or -1.[7]

Table 1. Values of the parameters used in the estimation of the logistic regression

Hyperparameters	Values
solver	newton-cg, lbfgs, *liblinear*, sag, saga
inverse of regularization strength	1.5, *1.0*, 0.5, 0.1, 0.05, 0.005
maximum number of iterations	50, *100*, 150, 200, 300, 350

Note: default values of hyperparameters are written in italics.

2.2. Random forest

Random forest is an ensemble model proposed by Breiman (2001). The random forest algorithm is known to often have higher accuracy than linear models and be less sensitive to the outliers in the data. As opposed to the logit-model it allows us to spot nonlinear relationships between the target and explanatory variables. At the same time, the results of this algorithm may be hard to interpret and it cannot be extrapolated to the new data. Moreover, the estimation of the random forest may require larger computational resources than the logit-model, which may limit the application of this algorithm to large-scale data.

The algorithm of random forest is based on decision trees. Each tree is a graph model which consists of a set of rules on explanatory variables to obtain the target variable. This model has a tree structure with nodes as decision points. The split occurs according to a certain criterion on one of the explanatory variables, while terminal nodes (leaves) contain the value of the target variable. The decision tree is built in a stepwise manner: first, the sample is split into two subsamples according to the specified criterion, then each of subsamples is consequently split further, until a certain stop criterion is not reached.[8]

[7] Logit-model, random forest and gradient boosting models are estimated with scikit-learn package (Python). We adjust the sample weights used in both these algorithms to deal with the unbalanced samples. Model estimation is conducted with a fixed random seed.
[8] Algorithm is presented in detail in Hastie et al. (2009), p. 588.

The random forest model can be expressed as follows. In a node m, in region R_m with N_m observations, the share of observations of class k in the node m is:

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x \in R_m}^{K} I(y_i = k). \tag{2}$$

Observations in the node m belong to the class with the highest number of observations, $k(m) = \arg\max_k \hat{p}_{mk}$. In order to evaluate the quality of a split we use the Gini criterion:[9]

$$\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk} (1 - \hat{p}_{mk}). \tag{3}$$

In the case of a binary classification it transforms into $2p(1-p)$, where p is a share of observations of the second class.

Via cross-validation we choose the architecture which ensures the best forecasting performance of random forest models. The considered values of hyperparameters are given in Table 2.

Table 2. Values of the parameters used in the estimation of random forest

Hyperparameters	Values
number of estimators	100, 500, 800
minimum number of samples in a leaf node	*1*, 2, 6
maximum depth of a tree	*None*, 5, 10, 25
minimum number of samples required to split a node	*2*, 6
maximum number of features	*auto*, log2

Note: default values of hyperparameters are written in italics.

2.3. Gradient boosting

Gradient boosting is another machine learning algorithm based on the combination of predictive models, decision trees in our case. In this form it was proposed by Friedman (2001). One of the advantages of a gradient boosting algorithm is its ability to generalize. It is often possible to build compositions, which outperform the basic algorithms. Gradient boosting can allows to identify outliers and to exclude them from the training set. However, it is known to have a tendency to overfit, while the stepwise approach of this algorithm can lead to a non-optimal set of weak learners. That is why it is highly important to choose optimally

[9] Other criteria to measure the quality of a split are presented in Hastie et al. (2009), p. 309. Within a robustness check we have considered entropy as criterion for the slit. The results show that it does not alter the values of the performance metrics: for most of the cases the difference in the values of F1-score and ROC-AUC are not statistically significant in the both cases (license withdrawal and requirements violation).

the combination of the number of estimators and learning rate, which can help to avoid overfitting.

Analytically gradient boosting can be expressed as an additive sum of more simple models:

$$F(x)=\sum_{m=1}^{M}\gamma_m h_m(x), \tag{4}$$

where $h_m(x)$ are basic functions, so-called weak learners (decision trees), and $\gamma_m(x)$ is a step length. The procedure of model estimation is the same in the case of classification and regression problem and differs only by type of loss function used. Gradient boosting is built in a stepwise manner in the following way:

$$F_m(x)=F_{m-1}(x)+\gamma_m h_m(x)\,. \tag{5}$$

On each step the decision tree $h_m(x)$ is chosen optimally from the minimization of the loss function L with a given F_{m-1} and $F_{m-1}(x_i)$:

$$h_m(x)=\arg\min_{h,\beta}\sum_{i=1}^{n}L\left(y_i,F_{m-1}(x_i)+\beta h(x_i)\right). \tag{6}$$

The minimization problem is solved via steepest descent, whose direction is defined as a negative gradient of the loss function evaluated at the current model F_{m-1}. The step length $\gamma_m(x)$ is chosen according to the equation (7):

$$\gamma_m=\arg\min_{\gamma}\sum_{i=1}^{n}L\left(y_i,F_{m-1}(x_i)-\gamma h_m(x)\right). \tag{7}$$

Within the search for the optimal architecture we consider different model characteristics, including the specification of the loss function, the maximum depth of a tree and the number of trees. The complete set of hyperparameters is presented in Table 3.

Table 3. Values of the parameters used in the estimation of gradient boosting

Hyperparameters	**Values**
loss function	*deviance*, exponential
learning rate	*0.1*, 0.05
number of estimators	75, *100*, 150
minimum number of observations in a leaf	*1*, 3
maximum depth	2, *3*, 4
maximum number of features	*None*, log2
subsample	*1.0*, 0.9

Note: default values of hyperparameters are written in italics.

2.4. Neural network

Here we consider a feed-forward neural network with three hidden layers. To avoid overfitting, at each hidden layer we use the dropout procedure where the neuron stays in the network with a fixed probability. We define the initial weights by Xavier initialization.[10]

A three layer neural network can be represented by equations (8)-(11):

$$h_1 = f\left(b_1 + \sum_{i=1}^{N}\left(w_{1i}x_i\right)\right), \tag{8}$$

$$h_2 = f\left(b_2 + \sum_{i=1}^{N_1}\left(w_{2i}h_{1i}\right)\right), \tag{9}$$

$$h_3 = f\left(b_3 + \sum_{i=1}^{N_2}\left(w_{3i}h_{2i}\right)\right), \tag{10}$$

$$y^{'} = g\left(b_y + w_y h_3\right), \tag{11}$$

where $x_1,...,x_N$ are input data with N features, w_j are weights of the hidden layers $j=1,...,3$ with N_1, N_2, N_3 being the number of neurons on the first, second and third layers and w_y is the vector of size $N_3 \times 1$ of weights on the output layer, b_j and b_y are biases, $f(\cdot)$ and $g(\cdot)$ are the activation functions (*ReLU* and *tanh*, correspondingly), h_j is the output of the hidden layer j, $y^{'}$ is the output of the neural network.[11]

In both tasks, whether it is the forecasting of license withdrawal or the prediction of requirements violation, the data are highly unbalanced. As the result, during the model training aggregate errors of the classification of the minor class are less significant compared to the aggregate errors of the major class classification. Therefore, the trained network would most likely attribute observations to the major class during the forecasting. There exist three common methods to deal with unbalanced datasets (see, e.g., He, Garcia (2008); Ganganwar, (2012); Sonak, Patankar (2015)). The first method consists of the exclusion the observations of the major class from the training set until the dataset becomes balanced. The second method is based on the inclusion of duplicates of minor class observations until the dataset becomes balanced. The third method prescribes an increase in the weights of errors of the minor class in the loss function. The choice of a particular method depends highly on the task. In our case the first method cannot be applied because the exclusion of major class

[10] For greater detail see, e.g., Glorot, Bengio (2010).

[11] Estimations are conducted with the use of TensorFlow framework (Abadi et al. (2016)) with Adam algorithm (Kingma, Ba (2014)) as a method for the cost-function optimization.

observations would lead to the exclusion of most of the observations from the training set, and the remaining data would not be sufficient to train the classifier. The application of the second method would lead to a more complicated cross-validation procedure and possible errors in it. Therefore, here we apply the third method: we use the custom loss function, with a higher weight of the minor class. It is represented by equation (12):

$$L=\sum_{i}^{N}\max\left\{0;0.01-(y\circ y')_i\right\},\text{where}$$

$$y=\begin{cases} weight*1, if \;\; y>0 \\ -1 \end{cases}, \;\; weight=\frac{N^{train}(y=-1)}{N^{train}(y=1)}, \qquad (12)$$

and y' is the estimate of an attribute to the particular class.[12] Initially y takes values of 1 or -1 depending on whether the bank license was withdrawn (or the requirement was violated) in this month or not.

We chose the optimal hyperparameters of the neural network via cross-validation. The considered hyperparameters are summarized in Table 4.

Table 4. Values of the parameters used in the estimation of the neural network

Hyperparameters	Values
type of regularization	L1, L2
size of regularization	0, 0.1
learning rate[13]	0.001
number of neurons at 1st, 2rd and 3rd layer	100, 400
drop-out at 1st and 3rd layer	0.25
drop-out at 2rd layer	0.25, 0.75
batch size[14]	200, 3000
epochs	10, 40

2.5. Model stacking

The stacking of models is widely used in machine learning in order to increase the accuracy and the stability of estimation. Stacking is a method of combining two or more different models to form the final result. This final model often has a higher forecasting performance compared to the individual models included in the stacking.

In the binary classification there exist different methods to form a model ensemble.[15] Here we apply stacking on the basis of a logistic-model. As features in the model we use the

[12] The window size of 0.01 was chosen empirically during preliminary estimates. In this case the classification with a right size by small in absolute value is penalized less.

[13] Learning rate is linearly decreasing to 1e-7 depending on the epoch.

[14] Batch size sets the number of observations used at each step of the gradient boosting during training.

[15] Clarke, Clarke (2018). Ensemble Methods, Predictive Statistics: Analysis and Inference beyond Models.

class predictions obtained in the first step from the models described above: random forest, gradient boosting and logit-model ("first level" models). In the second step, in order to train the stacking model and test its forecasting performance, we need two sets of data: training and test datasets. They are based on the data from the first step. In the second step, we obtain the training set in the following way. First, we choose the optimal hyperparameters for each "first level" model via cross-validation on the training set of the initial data. Next, this dataset is split into six equal subsets, where for each observation we form a prediction concerning the affiliation to a particular class on the basis of "first level" models trained at the remaining five datasets. After applying this procedure to each subset we obtain the predictions for each observation from the training set. The estimates on the control set are obtained from the training of "first level" models. They form the final estimate on the "first level" control set.

The choice of the optimal hyperparameters of the stacking model (based on the logit-model) is conducted via cross-validation on the training set of the "second level". Chosen hyperparameters are used in logit-model estimated on the whole "second level" training set. Next the final forecast is built on the control set. Hyperparameter values considered during cross-validation are presented in Table 5.[16]

Table 5. Values of the parameters used in the estimation of stacking via logistic regression

Hyperparameters	Values
solver	newton-cg, lbfgs, *liblinear*, sag, saga
inverse of regularization strength	1.5, *1.0*, 0.5, 0.1, 0.05, 0.005
maximum number of iterations	50, 75, *100*, 125, 150, 200

Note: default values of hyperparameters are written in italics.

3. Data and model estimation

3.1. Initial data and sample splitting

The models are estimated based on the bank accounting data published by the Bank of Russia. We use 35 series from 101, 102, 135 forms from February 2014 to October 2018 (Appendix, Table A1). Since the 102 form has quarterly frequency its figures are transformed into the monthly data. We split it into training and test sets: the data before June 2017 are included in the training set, while the remaining data are used to evaluate the forecasting

[16] Initial set of regularization values was adjusted after the preliminary estimates. Values differ from those in Table 1 due to the smaller number of features included in the model: 6 instead of 69 or 721, correspondingly.

performance of the models. As a part of data pre-processing we exclude the evident outliers and we cut the 99th percentile. We also exclude observations with the omitted values on a particular date, which do not allow us to build models with the considered number of lags or with the constructed variables from banking data.

On the basis of the initial 35 series we form two datasets, of 69 and 721 variables correspondingly, to construct predictions for 1, 2, 3 months, half a year and 9 months in advance. The first dataset additionally includes 34 variables formed on the basis of the initial data (Appendix, Table A2). The second dataset is obtained from the first one in the following way. For each variable we add its previous values and the differences with previous values with lags of 1, 2, 3 and 6 months.[17] Additionally, we add moving averages for 1, 2, 3 and 6 months and their pairwise differences for the considered bank ratios. As the result, the second dataset consists of 721 variables. We construct models for bank license withdrawal and the following statutory requirements: capital adequacy ratio, common equity Tier 1 ratio, Tier 1 capital ratio, instant and current liquidity ratios. The number of observations, left after pre-processing, in the training and test set for each target variable and each forecasting horizon is presented in Table A9 in the Appendix.

3.2. Performance metrics

There exist different metrics to evaluate the model performance in the binary classification problem. The choice of a particular metrics depends on the task at hand and on the data used. In the case of balanced data, when the number of observations of both classes is comparable, the use of a simple *accuracy* measure, calculated as a ratio of correctly classified observations to the total number of observations, is acceptable. However, in the prediction of bank license withdrawal and bank requirements violation the data are highly unbalanced. Both the number of license withdrawals and the cases of requirements violation is small comparing to the total number of observations. In this case the high value of the accuracy measure does not indicate the good forecasting performance of the model because this measure is based on the total number of correctly classified observations, without any separation by class. The accuracy measure would have a high value even if the model correctly classified most of the observations of the major class and was mistaken in all minor class observations. This measure is not applicable to the tasks we consider here as

[17] Mäkinen, Solanko (2017) show that the differences of CAMEL variables as opposed to their absolute values have the higher impact on the probability of default.

it does not allow us to correctly evaluate the forecasting performance of the model. That is why here we apply a set of different metrics to compare models by their performance.

Let us consider in greater detail the binary classification problem: the prediction of an observation class based on the estimation of its affiliation to a particular class. If this estimate exceeds a threshold level (cut-off threshold), the observation is referred to the one class, if the estimate is below the threshold – to the other class. The adjustment of the threshold level can lead to an increase or a decrease in the frequency of a particular class. In the case of binary classification ("positive" or "negative") the observation can belong to either of the following types depending on the probability of the model prediction:

- *TP (true positive)* – if the observation is correctly classified as "positive".
- *FP (false positive)* – if the observation is incorrectly classified as "positive".
- *TN (true negative)* – if the observation is correctly classified as "negative".
- *FN (false negative)* – if the observation is incorrectly classified as "negative".

Based on the number of observations in each of these classification types one can calculate the following measures:

1. **Sensitivity:**

$$TPR = \frac{TP}{TP + FN};$$

2. **Specificity:**

$$TNR = \frac{TN}{TN + FP};$$

3. **F1-score:**

$$F1 = \frac{2TP}{2TP + FP + FN}.$$

The other metric widely used along with these measures is a *ROC-AUC score*. It is defined as an area underneath the ROC curve, built in (TPR, 1-TNR) axes with the threshold changing from 0 to 1. Therefore, a ROC-AUC score provides a general characteristic of the classifier, which does not depend on the threshold level.

Sensitivity and specificity, as opposed to the accuracy measure, allow us to evaluate how well the classifier identifies the items of each class (correctly classified “positive” or “negative”). These two measures are related by the construction: in case of an imperfect classifier the model tuning aimed at the improvement of one metric will generally lead to the decrease in the other.

As opposed to sensitivity and specificity, the F1-score takes into account the information concerning the predictions of both classes. This allows us to evaluate the quality of the classification model with the use of one measure. That is why we apply the F1-score as a more appropriate quality metric for our tasks. In the case of bank license withdrawal or bank requirements violation we treat both errors as equivalent. So we use the F1-score which includes classification errors of both classes with equal weights. In the other tasks, where classification errors of one class are more important than the others, it is possible to use the F1-score with the different weights.

3.3. Cross-validation

Tuning of hyperparameters is an important stage in the training of a machine learning model. Due to the limited number of observations, here we use a cross-validation procedure to choose the optimal hyperparameters. The training dataset is split into three subsets, and each is used as a test set. The rest of the data is used in the training of the model with the given set of hyperparameters. After the training we select the threshold level from the maximization of F1-score on each test subset. For each of the subsets we calculate the actual F1-score for an average of the threshold levels obtained at each subset. The final F1-score is an average value of actual F1-scores on each subset. Afterwards we choose the set of hyperparameters with the highest F1-score. For the forecasting we use the model with the chosen set of hyperparameters and the threshold level. On its basis we build the forecast for the rest of the data.

In cross-validation splitting, which takes into account the data used, is crucial. The use of random splitting in the prediction problem can lead to ambiguous results. In the case of long-term forecasting the training and test sets can include “close” observations for a particular bank which can affect the performance metrics of the model. For example, in the forecasting of a bank license withdrawal in the next 6 months for the bank whose license was truly withdrawn in the data we would have 6 observations with the same value of the target variable (“license withdrawn”). At the same time, their explanatory variables can have close values. During the splitting of these data into training and test datasets there can be an information leak about the observation class label from training to test dataset. It can lead to overestimation of the score on the test set and interfere with the choice of the optimal hyperparameters of the model.

In order to avoid this leak during cross-validation, we relocate all observations for each bank to one of the three subsamples (later on, "controlled cross-validation"). The number of observations of bank with a withdrawn license is equally allocated to all three subsamples. Unallocated observations of banks who kept their license are dispersed among the subsets so to maintain the equality of the final subsample sizes.

4. Results

This section summarizes the results of the model estimation. First, we consider the forecasting of bank license withdrawal, comparing the performance of different machine learning algorithms (Section 3.1). We show how the model priority based on the forecast accuracy depends on the chosen performance metrics and how it can change when moving from cross-validation to the test set. Section 3.2 provides the results on the forecasting of violation of two requirements (Tier 1 capital adequacy ratio and current liquidity) as the most representative ones for the results on the capital and liquidity requirements. The results of the prediction of the violation of the other requirements are provided in Appendix. In Section 3.3 we check the robustness of the results, first, by considering an alternative type of data splitting on cross-validation (Subsection 3.3.1) and, second, by considering the larger data sample of explanatory variables (Subsection 3.3.2). We show that in the former case the results do change with the different, more favourable type of data split, resulting in higher levels of performance metrics due to the information leak from train to test subset during cross-validation. In the latter case, on the contrary, the larger data sample does not affect the results in one particular direction. The results vary depending on the forecasting horizon and the type of the considered model.

4.1. Forecasting license withdrawal

First, let us consider the performance of machine learning models in the task of forecasting *bank license withdrawal*. Figure 1 presents the results of the model estimation: F1-score and ROC-AUC on the cross-validation and test set. The results are provided for the estimation on the dataset of 69 variables. The optimal set of parameters chosen when using cross-validation for each forecasting horizon is proved in the Appendix (Table A3).

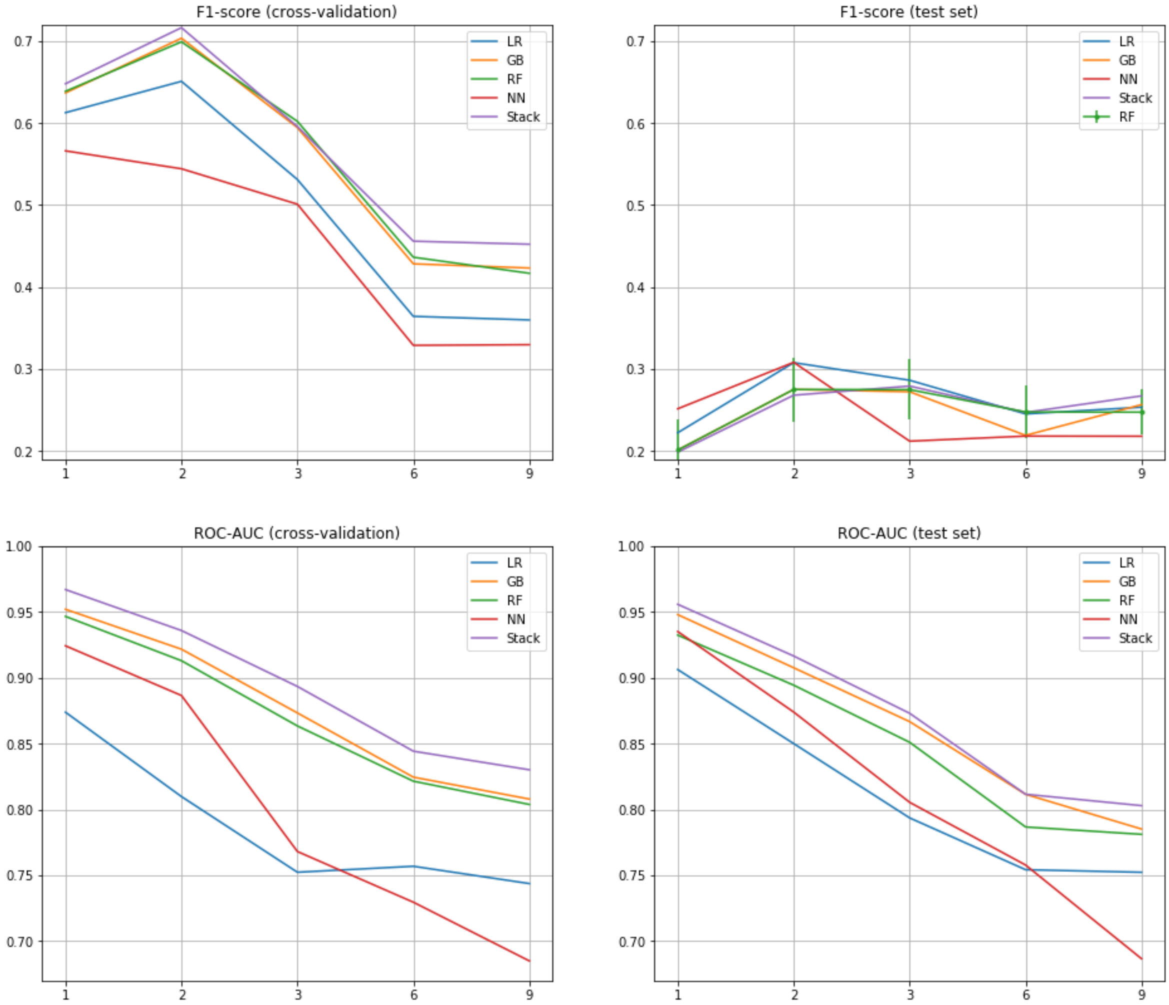

Figure 1. F1-score and ROC-AUC in the forecasting of bank license withdrawal
on the cross-validation set (on the left) and on the test set (on the right)

Let us consider first the performance of models on the cross-validation set. Gradient boosting, random forest and stacking have the best accuracy on all forecasting horizons compared to the logit-model and neural networks. The forecast based on stacking is the most accurate according to both measures (ROC-AUC and F1-score).[18]

The comparison with the estimation results on the test set shows that ROC-AUC measure and F1-score change in a different way, advocating for different models. ROC-AUC measures of gradient boosting, random forest and stacking slightly decrease on the test set but remain close to their values on cross-validation set. The F1-score, on the contrary, falls significantly on the test set compared to its values on cross-validation. The gap between the F1-score on the cross-validation set and the test set with almost unchanged values of ROC-AUC measures can be a sign of a significant difference in the data in these datasets, as the

[18] For the three months in advance the prediction based on the stacking model the cross-validation set has the comparable accuracy with the gradient boosting model.

test set includes data from June 2017. It can probably be explained by the active policy of the Bank of Russia's active policy during 2014-2016 aimed at the reorganization of the banking sector. That is why the test set may include banks, whose license may be withdrawn due to the other reasons compared to those included in the data during cross-validation.

To identify whether the difference in F1-scores of the considered models is significant on the test set we apply the bootstrapping technique, generating 1000 subsamples drawn from the test set. The obtained distribution, cut at the 5th and 95th percentiles, is presented in Figure 1. We can see that there is a significant difference only with the neural network models for the horizons of one and three months. In the other cases the difference in the performance is not significant, suggesting that machine learning models can indeed compete with logit-model in the forecasting of license withdrawal.

Despite the structural changes occurring during the considered time period the models presented here allow us to obtain more accurate forecasts than those previously obtained in the literature (on the basis of the ROC-AUC measure). The forecasting results presented, for instance, are more accurate than those of Emelyanov, Briukhova (2015), for the comparable forecasting horizons (on the basis of the area under ROC-curve). However, Emelyanov, Briukhova (2015) focus only on the cases of license withdrawal linked to financial stability of the bank, considering 30 banks in the train and 10 banks in the test set. Thus the comparison with their results is not entirely correct due to the different time period and the limited sample of banks considered. It is possible that this difference in the forecast accuracy would be even bigger if the machine learning approach would be applied to the data on preselected banks, limiting the scope of the analysis to the financial reasons of license withdrawal only.

4.2. Forecasting of the requirements violation

Within the task of forecasting bank requirements violation we have considered three capital requirements (capital adequacy ratio, common equity Tier 1 adequacy ratio, Tier 1 capital adequacy ratio) and two liquidity requirements (instant and current liquidity). Due to the similarity of the estimation results for all three capital ratios and two liquidity requirements we provide here only the results for Tier 1 capital ratio and the current liquidity ratio. The results for the other requirements are provided in the Appendix (Figures A2-A6). Figure 2 summarizes the results for the violation of Tier 1 capital ratio. The optimal set of parameters chosen on cross-validation for each requirement violation and each forecasting horizon is provided in the Appendix (Table A4-A8).

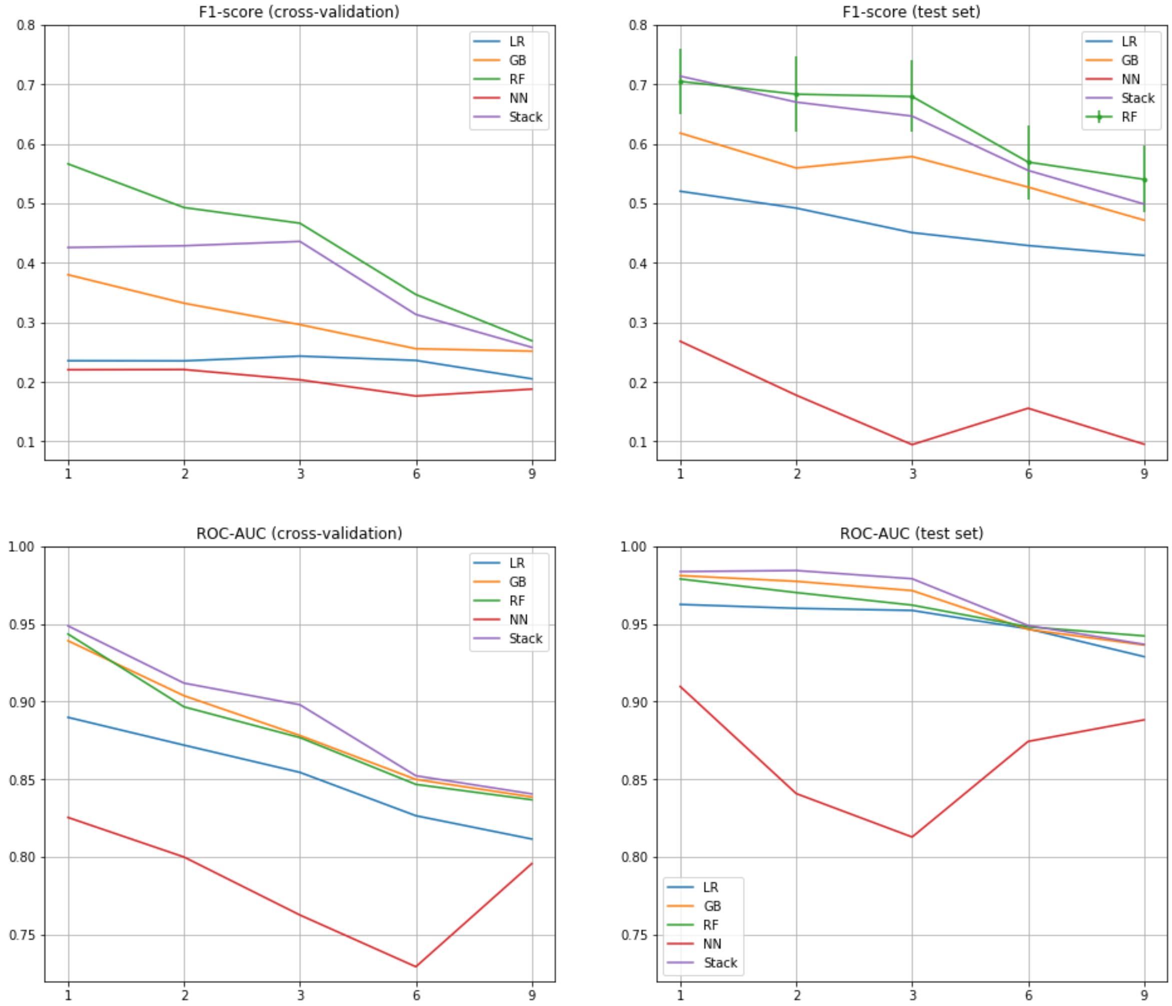

Figure 2. F1-score and ROC-AUC in the forecasting of Tier 1 capital ratio
on the cross-validation set (on the left) and on the test set (on the right)

In the case of requirements violation as opposed to license withdrawal the preferable models on the cross-validation set differ depending on the considered performance metric. Random forest models on the cross-validation set have the highest level of the F1-score at all forecasting horizons. At the same time the stacking models have the highest level of ROC-AUC, outperforming random forest and gradient boosting. As opposed to the license withdrawal task, the logit-model has a lower forecasting performance according to both performance metrics, compared to the random forest and gradient boosting. As in the case of license withdrawal, neural networks continue to have the lowest scores.

According to both metrics on the test set for most of the horizons the priority of models by their accuracy remains consistent. This underlines the need to choose the target performance metric as early as at the cross-validation stage. Despite the higher ROC-AUC levels of gradient boosting models they have the lower values of F1-score compared to random forest on both cross-validation and the test sets. Moreover, the difference in the F1-

score between models on the test set is significant for all forecasting horizons except the horizon of 6 months. The F1-score is the most appropriate metric to evaluate the forecasting performance of the presented models as it equally accounts for both binary classification errors for a given threshold level. At the same time, ROC-AUC is a general classification metric which does not depend on the threshold level. Its higher level for one algorithm does not imply the existence of the threshold level for which the F1-score will also be higher than for the other algorithm.

Now let us move to the case of current liquidity. Figure 3 shows the results for performance metrics for the forecasting of the violation of current liquidity on the basis of 69 variables. The results for instant liquidity are similar and are presented in Appendix (Figure A6).

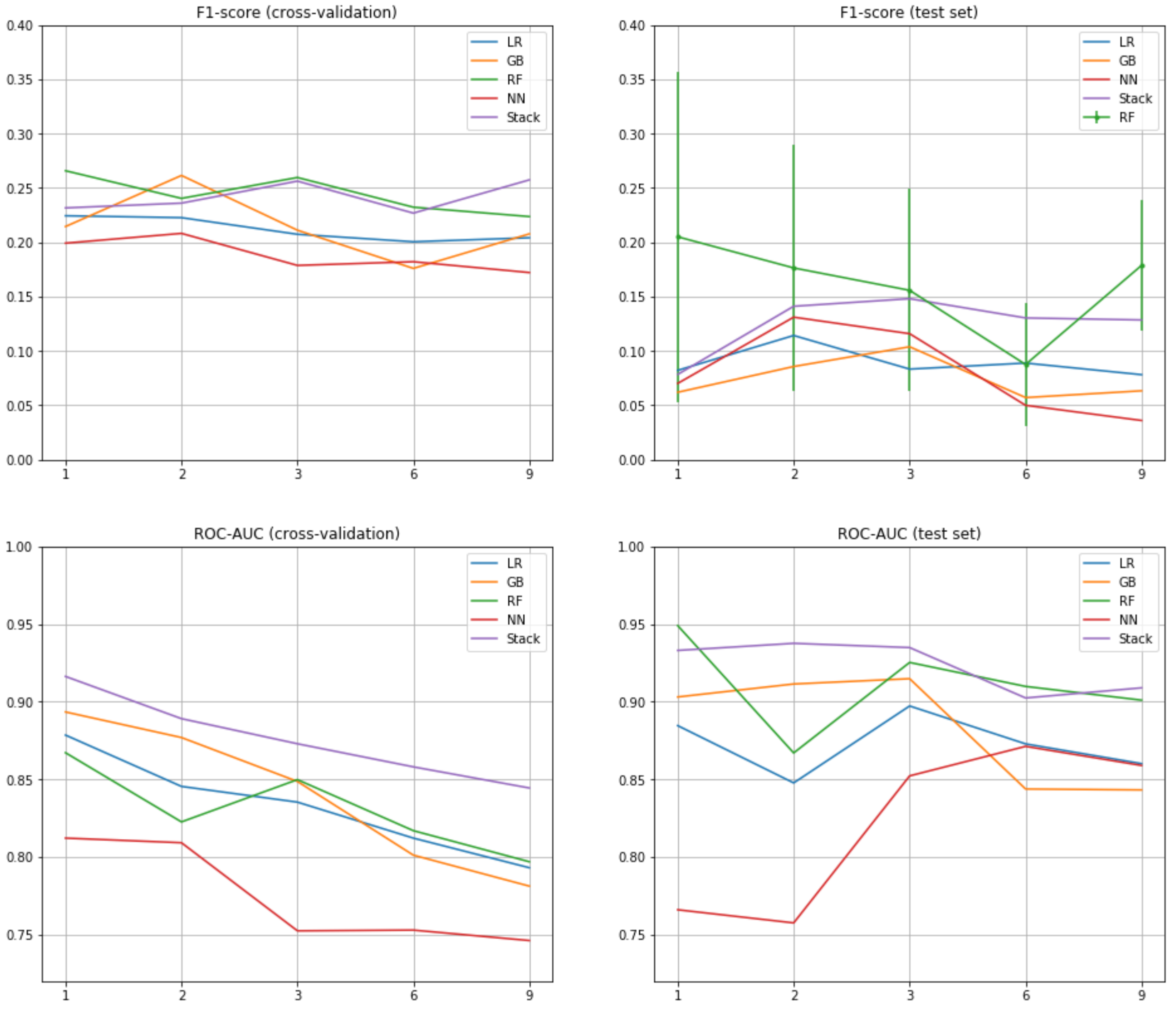

Figure 3. F1-score and ROC-AUC in the forecasting the violation of current liquidity ratio
on the cross-validation set (on the left) and on the test set (on the right)

Overall, the priority of the models remains consistent when moving from cross-validation to the test set with stacking, gradient boosting or random forest, depending on the forecasting horizon, outperforming the logit-model. Specifically, on the cross-validation set random forest and stacking models have the highest F1-scores compared to the other models for most forecasting horizons except the period of 2 months, where gradient boosting has the higher score. Neural networks have the lowest F1-score for all forecasting horizons except the period of 6 months, where gradient boosting has the worst performance. Neural networks and gradient boosting models have one of the lowest ROC-AUC levels at 6 and 9 months horizons, while stacking has the highest ROC-AUC value. On the test set the stacking (for 2, 3 and 9 months) and random forest models (for 1 and 6 months) have the highest ROC-AUC levels. In most cases the ROC-AUC values for stacking and random forest models are rather close. For most horizons random forest models have the highest levels of F1-score (except the 6 months horizon). However, the estimated confidence bands do not suggest that the difference between models at short horizons is the significant one (Figure 3).

On the cross-validation and test sets the forecasting accuracy of the models built for the forecasting of the violation of liquidity requirements is lower than the one of the models aimed at forecasting the violation of capital requirements (capital adequacy ratio, common equity Tier 1 ratio, Tier 1 capital ratio). It can be explained by the high volatility of the liquidity ratios, where the previous dynamics of ratios and explanatory variables has little information for the forecasting of their future dynamics.

4.3. Robustness of the results

4.3.1. The types of data splitting to subsamples

Figure 2 clearly illustrates the gap between the levels of performance metrics on cross-validation and test sets. Values on the test set are higher for all horizons and all models except the neural networks. This gap may be due to the chosen cross-validation methodology (controlled splitting), which underestimates the metric values on cross-validation. In order to verify this hypothesis we consider another type of data splitting, random splitting: splitting where the observations of each bank are allocated randomly across subsamples. The number of observations of both classes in the subsamples is kept equal as far as possible.

Let us compare the levels of performance metrics for random forest and logit-model, estimated for two different types of data splitting during cross-validation. It is clear from Figure 4 that the type of splitting does indeed significantly affect the size of the gap between metrics levels on cross-validation and test sets. As expected, the metrics level under random splitting is generally higher compared to the controlled splitting. The reason for the higher metric levels on the cross-validation set is twofold. The first effect appears because in the case of controlled splitting in the training set there are no observations of the banks which were included in the test set. Thus, the models are trained on the data on another banks. It is a positive consequence of this type of splitting as in this case the classifier draws patterns for the banks with the common features. However, despite the close values of the explanatory variables, the level of ratios may vary, and thus, the violation of the requirements may occur or not depending on the bank under consideration. This may bring about an error in the forecasting of requirements violation. At the same time, in the case of random splitting, the models get more information on the bank included in the test set, as the observations on a particular bank are likely to be included in the training set. This effect makes the cross-validation with random splitting close to real forecasting.

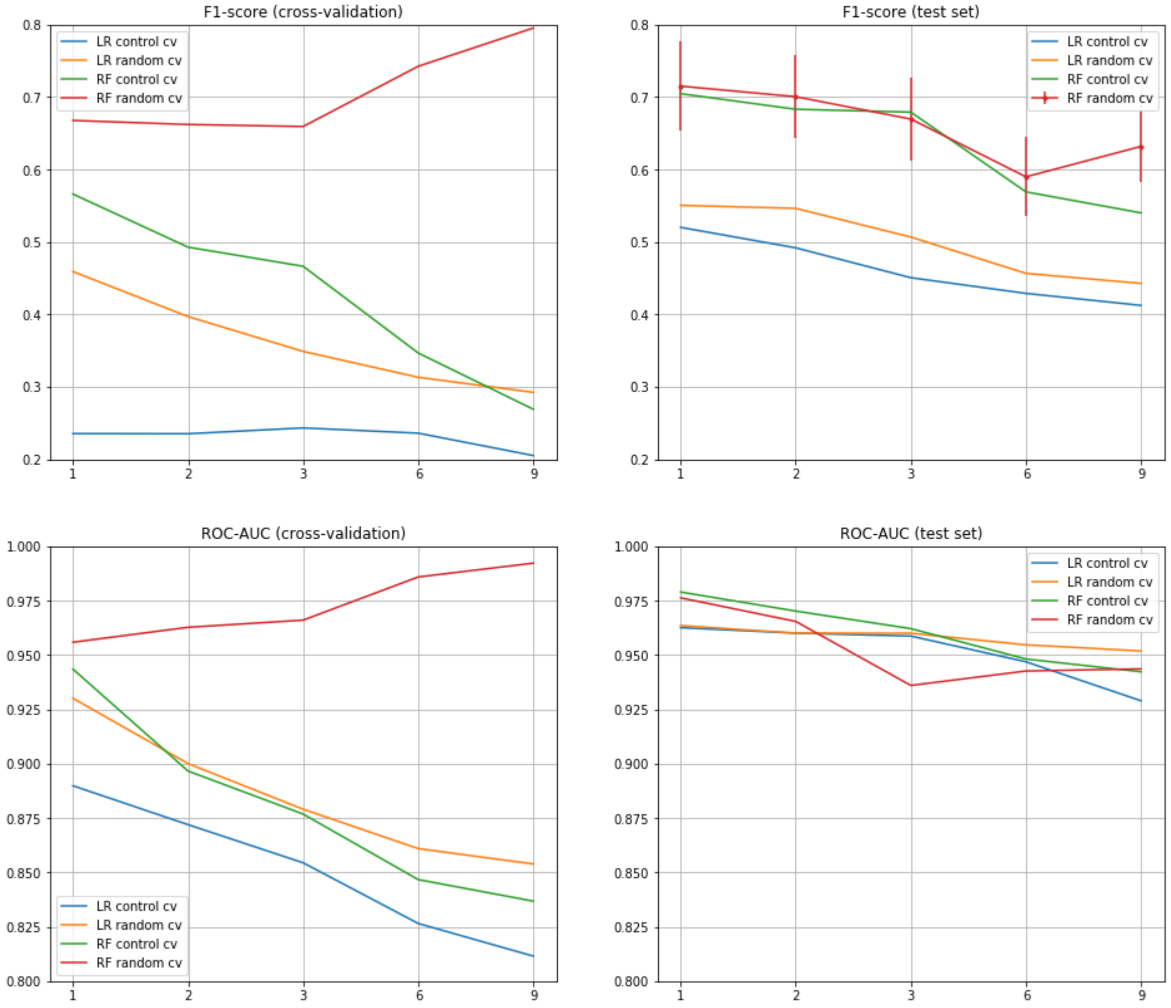

Figure 4. Comparison of performance metrics under two types of cross-validation
(forecasting of the violation of Tier 1 capital adequacy ratio)

The second effect consists in the plausible inclusion of the similar observations for a particular bank in both the training and test subsets on cross-validation. If we take, for example, the forecasting of the June violation of the requirement for three months in advance we would have three observations with the “violation” label (for March, April and May). The values of bank accounting data for three consecutive months can be close to each other, so when the data are split in the training and test sets on cross-validation there will be an information leak about the observation class. In this case, there will be an overestimation of the metric levels on cross-validation compared to the real levels.

This effect is well illustrated by the measure levels for the random forest. Instead of the expected fall in the measure values with the extension of the forecasting period we, in contrast, notice a sizeable increase in the F1-score starting from the three-month horizon. For ROC-AUC this effect appears from the 2-month horizon. At the same time, there is no evidence of this effect for the logit-model. This can be explained by the complexity of the random forest algorithm compared to the logit-model. Being a nonlinear algorithm, random forest can be trained to better capture complex relationships. That is why it can better capture the mentioned information leak from the training to the test subset on cross-validation. This can be implicitly confirmed by the fact that under random splitting the optimal model architecture is more complex than the one under controlled splitting.

The difference in the levels of performance metrics is clear from their comparison on cross-validation: under controlled splitting the levels are lower than under random splitting. The same is true for the logit-models on the test set. For this class of models, therefore, it is preferable to apply random splitting on cross-validation. For random forest models the preferable type of data splitting on cross-validation is less obvious. Under random splitting we obtain more complex random forest models and, thus, probably less stable. Therefore, we can assume that in the case of random forest it is better to apply controlled data splitting on cross-validation.

4.3.2. Number of variables and the forecast accuracy

In this subsection we provide an estimation of the models described above on two datasets of different size: the first dataset consists of 69 variables, the second one – of 721 variables and is based on the first dataset. We construct the second dataset to verify whether we can improve the forecasting performance by training models on a more detailed dataset, which includes additional information on the previous values of the variables. Figure 5 shows the performance metrics of the logit and random forest models for forecasting the violation of the Tier 1 capital adequacy ratio.

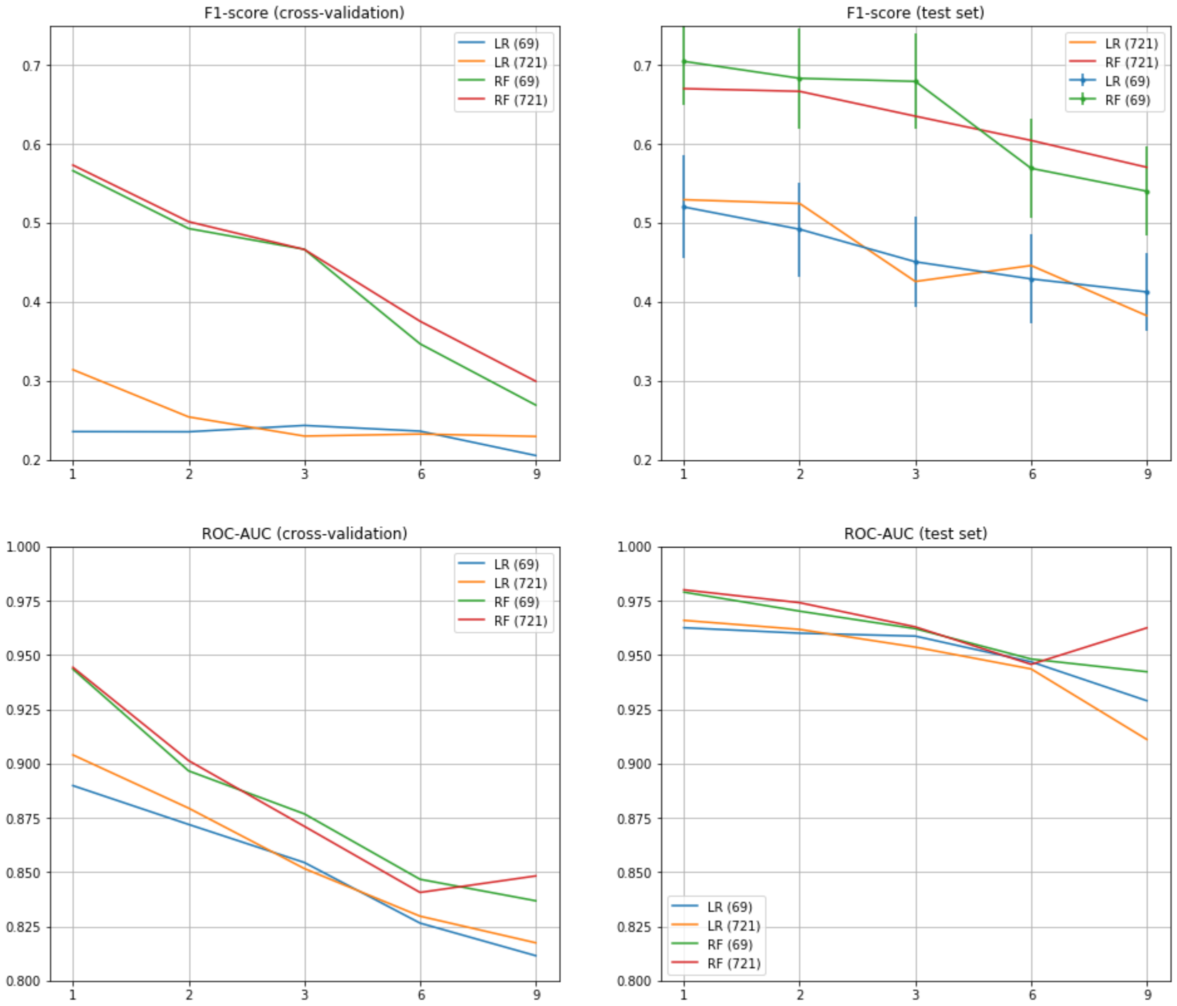

Figure 5. Comparison of performance metrics in the forecasting of Tier 1 capital ratio violation depending on the number of explanatory variables
on cross-validation set (on the left) and on the test set (on the right)

We can see that on cross-validation set random forest models estimated on the bigger data sample have higher F1-scores compared to the models based on the smaller sample. Their ROC-AUC measures are also higher for the horizons of 1, 2 and 9 months, although the difference in measures is not significant. Logit-models estimated on 721 variables have higher F1-scores for 1, 2 and 9 months and ROC-AUC measures at all horizons, except 3 months.

On the test set according to both metrics the model priority remains unchanged for logit-models for 1 and 2 months and for random forest at the horizon of 9 months. Confidence bands for the F1-score suggest that the difference between the performance of models built on a large and small dataset is not significant. As for license withdrawal (Appendix, A7) and the violation of the other requirements the difference in forecasting performance (F1-score) of the models estimated on 69 and 721 is also insignificant in most of the cases.[19] The obtained results suggest that the use of additional data constructed from the initial data sample does not allow us to significantly improve the forecasting performance of the considered machine learning models.

Conclusion

We apply machine learning techniques to choose the optimal model architectures for the forecasting of bank license withdrawal and bank requirements violations (capital adequacy ratio, common equity Tier 1 adequacy ratio, Tier 1 capital adequacy ratio, instant and current liquidity requirements). We estimate random forest, gradient boosting and neural network models along with stacking and compare the results with the logit-model forecasts for the horizons of one, two, three months, half of year and 9 months. We consider two datasets, containing 69 and 721 variables, based on 35 monthly indicators from Russian banks' accounting from February 2014 to October 2018. We compare the results depending on the performance metric used (F1-score or ROC-AUC), showing the importance of the choice of the correct performance metric for the task under consideration.

We show that the logit-model, widely used in the related literature, may not be the model with the highest forecasting accuracy. In the case of license withdrawal prediction all considered models with the optimal architecture have a comparable forecasting performance on the test set. However, on cross-validation set the gradient boosting and random forest models have considerably higher F1-scores compared to the logit-models. The performance metrics of these models on cross-validation set are also higher than those on the test set. This can be explained by the structural change in the data on cross-validation set and in the test set and shows the potential of such machine learning models as gradient boosting and random forest to provide more accurate forecasts of license withdrawals than the commonly used methods.

In the case of forecasting requirements violation the gradient boosting and random forest models can also compete with the logit-model in terms of forecast accuracy. In particular, random forest models have higher accuracy for most horizons in the forecasting of capital adequacy ratios violation. In the case of the violation of liquidity ratios the levels of

[19] Figures for the requirements violations are omitted for the sake of brevity and can be provided upon the request.

performance metrics of different models are comparable and lower than those in the forecasting of capital ratios. This may be explained by the higher volatility of liquidity ratios compared to the capital ones, which complicate the short-term forecasting of these requirements.

Particular attention is paid to the methodology of data splitting when using cross-validation and its effect on the forecasting performance of the models. In the example of forecasting the violation of Tier 1 capital ratio with the random forest and logit-model, we show that the levels of performance metrics may significantly depend on the type of data splitting. This result underlines the need to choose the appropriate type of data splitting, taking into account the forecasting target and the characteristics of the data. We also check the robustness of the results on the dataset of 721 variables formed from previously used 69 variables, showing that this data extension does not allow us to qualitatively increase the forecasting performance of the considered models.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. Tensorflow: A system for large-scale machine learning, OSDI 16, pp. 265-283.
2. Bagherpour, A. (2017). Predicting Mortgage Loan Default with Machine Learning Methods. *University of California/Riverside*.
3. Belousova, V., Karminsky, A., Kozyr, I. (2018). Bank ownership and profit efficiency of Russian banks. *BOFIT Discussion Papers 5.*
4. Beutel, J., List, S., von Schweinitz, G. (2018). An evaluation of early warning models for systemic banking crises: Does machine learning improve predictions?.
5. Bidzhoyan D. S. (2018). Model for assessing the probability of revocation of a license from the Russian bank. *Finance: Theory and Practice*, 22(2):26-37. DOI: 10.26794/2587-5671-2018-22-2-26-37 (in Russian).
6. Bidzhoyan D. S., T. K. Bogdanova (2017). The concept of modeling and forecasting the probability of revoking a license of Russian banks, *Economics and contemporary Russia*, (4), 88-102. (in Russian).
7. Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.
8. Claeys, S., Lanine, G., Schoors, K. J. (2005). Bank supervision Russian style: Rules versus enforcement and tacit objectives. *BOFIT Discussion Papers 10*.
9. Claeys, S., Schoors, K. (2007). Bank supervision Russian style: Evidence of conflicts between micro-and macro-prudential concerns. *Journal of Comparative Economics*, *35*(3), 630-657.
10. Clarke, B., Clarke, J. (2018). Ensemble Methods. In Predictive Statistics: Analysis and Inference beyond Models (Cambridge Series in Statistical and Probabilistic Mathematics, pp. 449-523). Cambridge: Cambridge University Press. doi:10.1017/9781139236003.012
11. Emelyanov A. M., O. O. Briukhova (2013). The estimation of the probability of bank default, *Finance and credit* (27 (555)). (in Russian).
12. Emelyanov A. M., O. O. Briukhova, Drivers of Banks License Withdrawal: the after Crisis (2010–2011) Study, *Economics and mathematical methods*, *51*(3), pp. 41-53. (in Russian).
13. Friedman, J., Hastie, T., Tibshirani, R. (2009). The elements of statistical learning: Data Mining, Inference, and Prediction, Second Edition, Springer series in statistics.

14. Fungáčová, Z., Solanko, L. (2009). Risk-taking by Russian banks: Do location, ownership and size matter?. *The HSE Economic Journal, 13*(1).
15. Fungacova, Z., Weill, L. (2009). How market power influences bank failures: Evidence from Russia. *BOFIT Discussion Papers 12*.
16. Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, *2*(4), 42-47.
17. Glorot, X., Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249-256).
18. He, H., Garcia, E. A. (2008). Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, (9), 1263-1284.
19. Karas, A., Schoors, K., Weill, L. (2010). Are private banks more efficient than public banks? Evidence from Russia. *Economics of Transition*, *18*(1), 209-244.
20. Karminsky A. M., Kostrov A. V. (2013). Probability of default of Russian banks modelling: new possibilities, *Journal of new economic association*. Т. 17. № 1. pp. 64-86. (in Russian).
21. Karminsky A. M., Peresetsky A. A., (2007). International agencies' ratings models. *Applied Econometrics*, №1, 2007, стр. 3-19. (in Russian).
22. Karminsky, A., Kostrov, A. (2017). The back side of banking in Russia: Forecasting bank failures with negative capital. *International Journal of Computational Economics and Econometrics*, *7*(1-2), 170-209.
23. Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
24. Lanine, G., Vander Vennet, R. (2006). Failure prediction in the Russian bank sector with logit and trait recognition models. *Expert Systems with Applications*, *30*(3), 463-478.
25. Mai, C., & Baek, S. (2012) Predicting Bank Default from the Quarterly Report. Stanford University.
26. Mäkinen, M., Solanko, L. (2017). Determinants of bank closures: Do changes of CAMEL variables matter?. BOFIT Discussion Papers 16.
27. Makinen, M., Solanko, L. (2018). Determinants of Bank Closures: Do Levels or Changes of CAMEL Variables Matter? *Russian Journal of Money and Finance*, v. 77(2), pp. 3-21. doi: 10.31477/rjmf.201802.03. (in Russian).

28. Peresetsky A. A. (2012). Economic approach to off-site analysis of Russian banks. NRU HSE (in Russian).
29. Peresetsky A. A. (2013). Modeling reasons for Russian bank license withdrawal: Unaccounted factors. *Applied Econometrics*, 30 (2), 49–64. (in Russian).
30. Peresetsky A. (2009). Measuring external support factor of Moody's bank ratings. *Applied Econometrics*, №2, pp. 3–23. (in Russian).
31. Peresetsky A. A. (2007). Banks' probability of default models. *Economics and Mathematical Methods*, v.43, n.3, pp. 37–62 (in Russian).
32. Peresetsky, A. A., Karminsky, A. A., Golovan, S. V. (2011). Probability of default models of Russian banks. *Economic Change and Restructuring*, *44*(4), 297-334.
33. Petropoulos A., Siakoulis V., Stavroulakis E., Vlachogiannakis N., (2017) Predicting bank insolvencies using machine learning techniques, 2017 EBA Policy Research Workshop, London.
34. Sinelnikova-Muryleva E. V., Gorshkova T.G., Makeeva N.V. (2018). Default Forecasting in the Russian Banking Sector. *Economic Policy* 13(2):8-27. DOI: 10.18288/1994-5124-2018-2-01.
35. Sonak, A., Patankar, R. A. (2015). A survey on methods to handle imbalance dataset. *Int. J. Comput. Sci. Mobile Comput*, *4*(11), 338-343.
36. Styrin, K. (2005). X-inefficiency, moral hazard, and bank failures. *Economic Education and Research Consortium, Russia and CIS, Final Report, Moscow*.
37. Van Soest, A. H. O., Peresetsky, A., Karminsky, A. M. (2003). An Analysis of Ratings of Russian Banks. *CentER Discussion Paper*, *2003*.
38. Vasiluk A. A., Karminsky A. M. (2011). Credit risk of Russian banks modelling based on Russian financial report standards, *Financial risk management*. № 3. P. pp. 194-205. (in Russian).
39. Zhivaikina A. D., Peresetsky A. A. (2017). Russian bank credit ratings and bank license withdrawal 2012–2016. *Journal of the New Economic Association*, 4 (36), pp. 49–80. (in Russian).

Appendix

Tables

Table A1. Variables based on the bank accounting

Variable	Notation
Form 101	
Balance assets	2071
Form 102	
Total income	10000
Interest income	11000
Total expenses	20100
Total in section "Interest paid"	21000
After-tax profit	31001
Losses after tax	31002
Form 135	
Assets with zero risk coefficient	arisk0
Highly liquid assets	lam
Total of large credit risks	kskr
Liquid assets	lat
Bank liabilities on loans and deposits and active bank bonds with maturity date over a year	od
Demand liabilities	ovm
Demand liabilities and for a term of up to 30 days	ovt
Internal funds (capital)	kap0
Common Equity Tier 1	kap1
Capital assets	kap2
Loans issued by a bank, deposits including those in precious metals with the remaining maturity exceeding a year	krd
Risk-weighted assets (capital adequacy ratio)	ar.0
Assets included in the first group not weighted by the risk (capital adequacy ratio)	ar1.0
Assets included in the second group (capital adequacy ratio)	ar2.0
Assets included in the fourth group (capital adequacy ratio)	ar4.0
Risk-weighted assets (common equity Tier 1 ratio)	ar.1
Assets included in the first group not weighted by the risk (common equity Tier 1 ratio)	ar1.1
Assets included in the second group (common equity Tier 1 ratio)	ar2.1
Assets included in the fourth group (common equity Tier 1 ratio)	ar4.1
Risk-weighted assets (Tier 1 capital ratio)	ar.2
Assets included in the first group not weighted by the risk (Tier 1 capital ratio)	ar1.2
Assets included in the second group (Tier 1 capital ratio)	ar2.2
Assets included in the fourth group (Tier 1 capital ratio)	ar4.2
Capital adequacy ratio	n1.0
Common equity Tier 1 adequacy ratio	n1.1
Tier 1 capital adequacy ratio	n1.2
Instant liquidity ratio	n2
Current liquidity ratio	n3

Table A2. Variables formed on the bank accounting

Internal notation	Formula
log(2071)	log(2071)
lat_A	lat/2071*100
31001_A	31001/2071*100
31002_A	31002/2071*100
log(31001)	log(31001)
kskr_A	kskr/2071*100
krd_A	krd/2071*100
ovm_A	ovm/2071*100
od_A	od/2071*100
ovt_A	ovt/2071*100
31001/ovt	31001/ovt*100
31001/krd	31001/krd*100
10000_A	10000/2071*100
20100_A	20100/2071*100
11000_A	11000/2071*100
21000_A	21000/2071*100
31001/kap0	31001/kap0*100
kap0_A	kap0/2071*100
log(kap0)	log(kap0)
(11000+21000)_A	(11000 + 21000)/2071*100
log(arisk0)	log(arisk0)
arisk0_A	arisk0/2071*100
lam_A	lam/2071*100
log(lat)	log(lat)
log(lam)	log(lam)
kap0/ar.0	kap0/ar.0*100
kap1/ar.1	kap1/ar.1*100
kap2/ar.2	kap2/ar.2*100
sum_ar_0	ar1.0 + ar2.0 + ar4.0
sum_ar_1	ar1.1 + ar2.1 + ar4.1
sum_ar_2	ar1.2 + ar2.2 + ar4.2
kap0/sum_ar_0	kap0/sum_ar_0*100
kap1/sum_ar_1	kap1/sum_ar_1*100
kap2/sum_ar_2	kap2/sum_ar_2*100

Table A3. Hyperparameters chosen on cross-validation for each forecasting period

(license withdrawal)

			logistic regression		
	1 month	*2 months*	*3 months*	*6 months*	*9 months*
C	0.1	0.1	0.05	1.5	0.5
solver	saga	saga	liblinear	newton-cg	liblinear
			random forest		
n_estimators	500	800	800	500	800
max_depth	10	10	5	10	10
min_samples_split	2	2	6	2	6
min_samples_leaf	6	6	1	6	2
max_features	log2	log2	auto	auto	auto
			gradient boosting		
n_estimators	100	100	100	150	100
learning_rate	0.05	0.05	0.1	0.1	0.1
loss	deviance	exponential	deviance	deviance	exponential
subsample	1	1	1	1	1
min_samples_leaf	1	1	1	1	1
max_depth	2	3	2	2	3
max_features	log2	log2	log2	log2	log2
			neural network		
epochs	10	10	40	40	40
batch_size	200	200	200	3000	200
learning_rate	0.001	0.001	0.001	0.001	0.001
drop_out_1	0.25	0.25	0.25	0.25	0.25
drop_out_2	0.75	0.75	0.75	0.25	0.25
drop_out_3	0.25	0.25	0.25	0.25	0.25
n_count_1	400	400	400	400	400
n_count_2	400	400	400	400	400
n_count_3	100	100	400	400	100
reg	l1	l2	l2	l2	l1
scale	0.1	0	0	0	0
			stacking		
C	0.05	0.1	0.005	0.005	0.005
solver	liblinear	saga	liblinear	saga	newton-cg

Table A4. Hyperparameters chosen on cross-validation for each forecasting period

(forecasting of violation of capital adequacy ratio)

			logistic regression		
	1 month	*2 months*	*3 months*	*6 months*	*9 months*
C	1.5	0.5	1.5	1.5	1
solver	lbfgs	sag	lbfgs	lbfgs	saga
			random forest		
n_estimators	800	500	100	100	500
max_depth	None	25	5	None	None
min_samples_split	6	2	6	6	2
min_samples_leaf	6	6	6	6	1
max_features	log2	auto	log2	log2	auto
			gradient boosting		
n_estimators	75	150	150	100	100
learning_rate	0.1	0.05	0.05	0.05	0.05
loss	exponential	exponential	exponential	exponential	exponential
subsample	1.0	0.9	1.0	1.0	1.0
min_samples_leaf	1	1	1	1	3
max_depth	3	2	3	2	2
max_features	log2	log2	log2	None	None
			neural network		
epochs	40	40	40	40	10
batch_size	3000	3000	3000	3000	3000
learning_rate	0.001	0.001	0.001	0.001	0.001
drop_out_1	0.25	0.25	0.25	0.25	0.25
drop_out_2	0.25	0.25	0.25	0.25	0.75
drop_out_3	0.25	0.25	0.25	0.25	0.25
n_count_1	400	400	400	400	400
n_count_2	100	100	400	400	400
n_count_3	400	100	400	400	100
reg	l1	l1	l2	l1	l1
scale	0	0	0.1	0	0
			stacking		
C	0.1	0.1	0.05	0.5	1
solver	saga	liblinear	liblinear	newton-cg	sag

Table A5. Hyperparameters chosen on cross-validation for each forecasting period

(forecasting of violation of common equity Tier 1 ratio)

			logistic regression		
	1 month	*2 months*	*3 months*	*6 months*	*9 months*
C	1.5	1.5	1.5	1.5	1.5
solver	lbfgs	sag	lbfgs	lbfgs	newton-cg
			random forest		
n_estimators	100	100	100	500	500
max_depth	5	10	10	5	5
min_samples_split	6	6	6	6	2
min_samples_leaf	2	1	6	1	1
max_features	log2	log2	auto	log2	log2
			gradient boosting		
n_estimators	150	150	100	150	150
learning_rate	0.05	0.05	0.1	0.05	0.1
loss	exponential	exponential	exponential	exponential	exponential
subsample	1	0.9	1	1	1
min_samples_leaf	1	3	1	1	3
max_depth	2	2	2	2	2
max_features	log2	log2	log2	log2	log2
			neural network		
epochs	40	40	40	40	40
batch_size	3000	3000	3000	3000	3000
learning_rate	0.001	0.001	0.001	0.001	0.001
drop_out_1	0.25	0.25	0.25	0.25	0.25
drop_out_2	0.25	0.25	0.25	0.25	0.25
drop_out_3	0.25	0.25	0.25	0.25	0.25
n_count_1	400	400	400	400	400
n_count_2	100	100	100	100	400
n_count_3	100	100	400	400	100
reg	l2	l2	l2	l2	l2
scale	0.1	0.1	0.1	0.1	0.1
			stacking		
C	0.5	0.05	0.5	0.5	0.5
solver	sag	liblinear	newton-cg	newton-cg	newton-cg

Table A6. Hyperparameters chosen on cross-validation for each forecasting period

(forecasting of violation of Tier 1 capital ratio)

	logistic regression				
	1 month	*2 months*	*3 months*	*6 months*	*9 months*
C	1.5	1.5	1.5	0.5	0.1
solver	saga	sag	lbfgs	newton-cg	liblinear
	random forest				
n_estimators	800	100	500	100	100
max_depth	10	10	10	5	5
min_samples_split	2	2	2	2	6
min_samples_leaf	6	2	1	6	6
max_features	auto	log2	log2	auto	auto
	gradient boosting				
n_estimators	75	75	100	150	100
learning_rate	0.1	0.05	0.05	0.1	0.05
loss	exponential	exponential	exponential	exponential	exponential
subsample	1	0.9	1	1	0.9
min_samples_leaf	1	1	1	1	1
max_depth	2	3	2	2	2
max_features	log2	None	None	log2	log2
	neural network				
epochs	40	40	40	40	10
batch_size	3000	3000	3000	3000	3000
learning_rate	0.001	0.001	0.001	0.001	0.001
drop_out_1	0.25	0.25	0.25	0.25	0.25
drop_out_2	0.25	0.25	0.25	0.25	0.25
drop_out_3	0.25	0.25	0.25	0.25	0.25
n_count_1	400	400	400	400	400
n_count_2	100	100	100	100	100
n_count_3	400	400	400	400	100
reg	l1	l2	l1	l2	l1
scale	0.1	0.1	0.1	0	0.1
	stacking				
C	1	0.1	0.1	0.05	0.005
solver	newton-cg	lbfgs	liblinear	liblinear	sag

Table A7. Hyperparameters chosen on cross-validation for each forecasting period

(forecasting of violation of instant liquidity ratio)

			logistic regression		
	1 month	*2 months*	*3 months*	*6 months*	*9 months*
C	1.5	1	1.5	1.5	0.05
solver	sag	sag	sag	lbfgs	lbfgs
			random forest		
n_estimators	500	100	100	100	100
max_depth	10	None	None	10	10
min_samples_split	2	2	6	6	6
min_samples_leaf	2	1	2	2	6
max_features	auto	auto	log2	auto	log2
			gradient boosting		
n_estimators	150	100	150	150	75
learning_rate	0.05	0.05	0.1	0.1	0.1
loss	exponential	exponential	exponential	deviance	exponential
subsample	1	0.9	1	1	1
min_samples_leaf	1	1	1	1	1
max_depth	3	2	2	2	2
max_features	log2	None	log2	log2	None
			neural network		
epochs	10	40	40	40	40
batch_size	3000	3000	3000	3000	3000
learning_rate	0.001	0.001	0.001	0.001	0.001
drop_out_1	0.25	0.25	0.25	0.25	0.25
drop_out_2	0.75	0.25	0.25	0.25	0.75
drop_out_3	0.25	0.25	0.25	0.25	0.25
n_count_1	400	400	400	400	400
n_count_2	400	400	400	400	400
n_count_3	400	400	100	400	400
reg	l1	l2	l2	l1	l1
scale	0.1	0.1	0.1	0	0.1
			stacking		
C	1	1	0.5	0.5	0.5
solver	liblinear	newton-cg	liblinear	liblinear	sag

Table A8. Hyperparameters chosen on cross-validation for each forecasting period

(forecasting of violation of current liquidity ratio)

	logistic regression				
	1 month	*2 months*	*3 months*	*6 months*	*9 months*
C	1.5	1.5	1.5	1.5	1.5
solver	saga	newton-cg	lbfgs	lbfgs	saga
	random forest				
n_estimators	800	100	100	100	500
max_depth	25	None	5	5	10
min_samples_split	6	2	2	6	2
min_samples_leaf	2	2	6	1	1
max_features	auto	log2	log2	log2	log2
	gradient boosting				
n_estimators	100	150	150	150	100
learning_rate	0.05	0.1	0.05	0.05	0.1
loss	deviance	exponential	exponential	exponential	deviance
subsample	1	0.9	1	1	0.9
min_samples_leaf	1	3	1	1	3
max_depth	3	2	2	2	2
max_features	log2	log2	log2	None	None
	neural network				
epochs	40	40	40	40	10
batch_size	3000	3000	3000	3000	3000
learning_rate	0.001	0.001	0.001	0.001	0.001
drop_out_1	0.25	0.25	0.25	0.25	0.25
drop_out_2	0.25	0.25	0.25	0.75	0.75
drop_out_3	0.25	0.25	0.25	0.25	0.25
n_count_1	400	400	400	400	400
n_count_2	400	400	400	400	400
n_count_3	400	100	100	100	100
reg	l2	l1	l2	l1	l1
scale	0.1	0.1	0	0.1	0
	stacking				
C	0.05	0.05	0.5	1	0.05
solver	liblinear	liblinear	saga	newton-cg	liblinear

Table A9. The structure of the dataset for license withdrawal and requirements violations for the horizon of 1, 2, 3, 6 and 9 months

	1 month				**2 months**				**3 months**			
	train		**test**		**train**		**test**		**train**		**test**	
	neg.	pos.	neg.	pos.	neg.	pos.	neg.	pos.	neg.	pos.	neg.	pos.
license withdrawal	26644	284	7098	70	26357	571	7034	134	26071	857	6972	196
N1.0	26739	189	7114	54	26664	264	7106	62	26596	332	7099	69
N1.1	26750	178	7098	70	26684	244	7086	82	26620	308	7077	91
N1.2	26696	232	7077	91	26620	308	7064	104	26546	382	7052	116
N2	26852	76	7163	5	26798	130	7159	9	26746	182	7155	13
N3	26795	133	7160	8	26716	212	7154	14	26640	288	7150	18

	6 months				**9 months**			
	train		**test**		**train**		**test**	
	neg.	pos.	neg.	pos.	neg.	pos.	neg.	pos.
license withdrawal	25225	1703	6800	368	24387	2541	6666	502
N1.0	26392	536	7080	88	26197	731	7067	101
N1.1	26428	500	7050	118	26251	677	7030	138
N1.2	26328	600	7021	147	26119	809	7005	163
N2	26601	327	7145	23	26471	457	7135	33
N3	26424	504	7138	30	26229	699	7127	41

Note: neg. and pos. signify whether or not the event of license withdrawal or the violation of the requirement occurs.

Figures

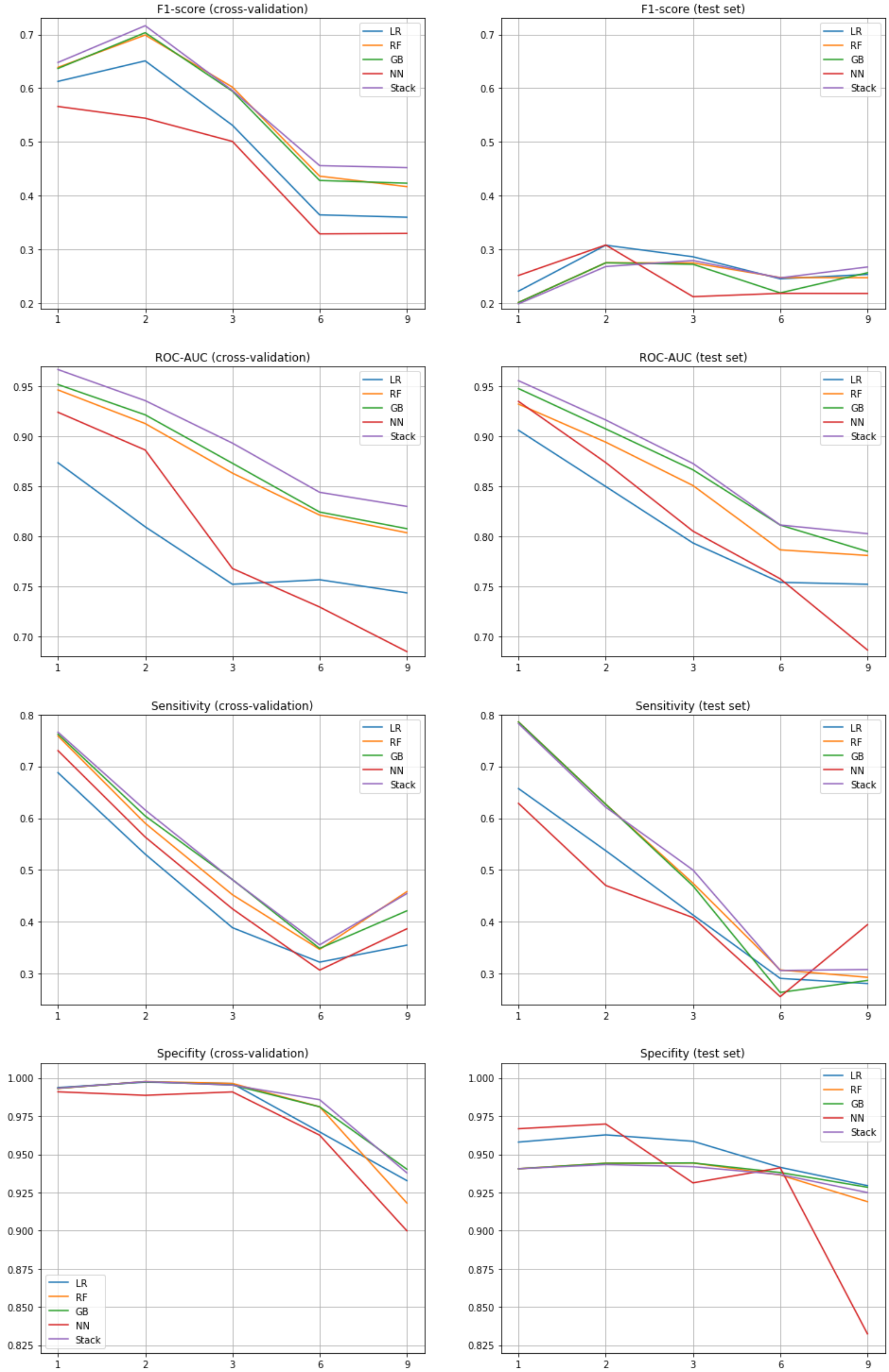

Figure A1. Metrics on cross-validation and test sets in the forecasting of bank license withdrawal

on cross-validation set (on the left) and on the test set (on the right)

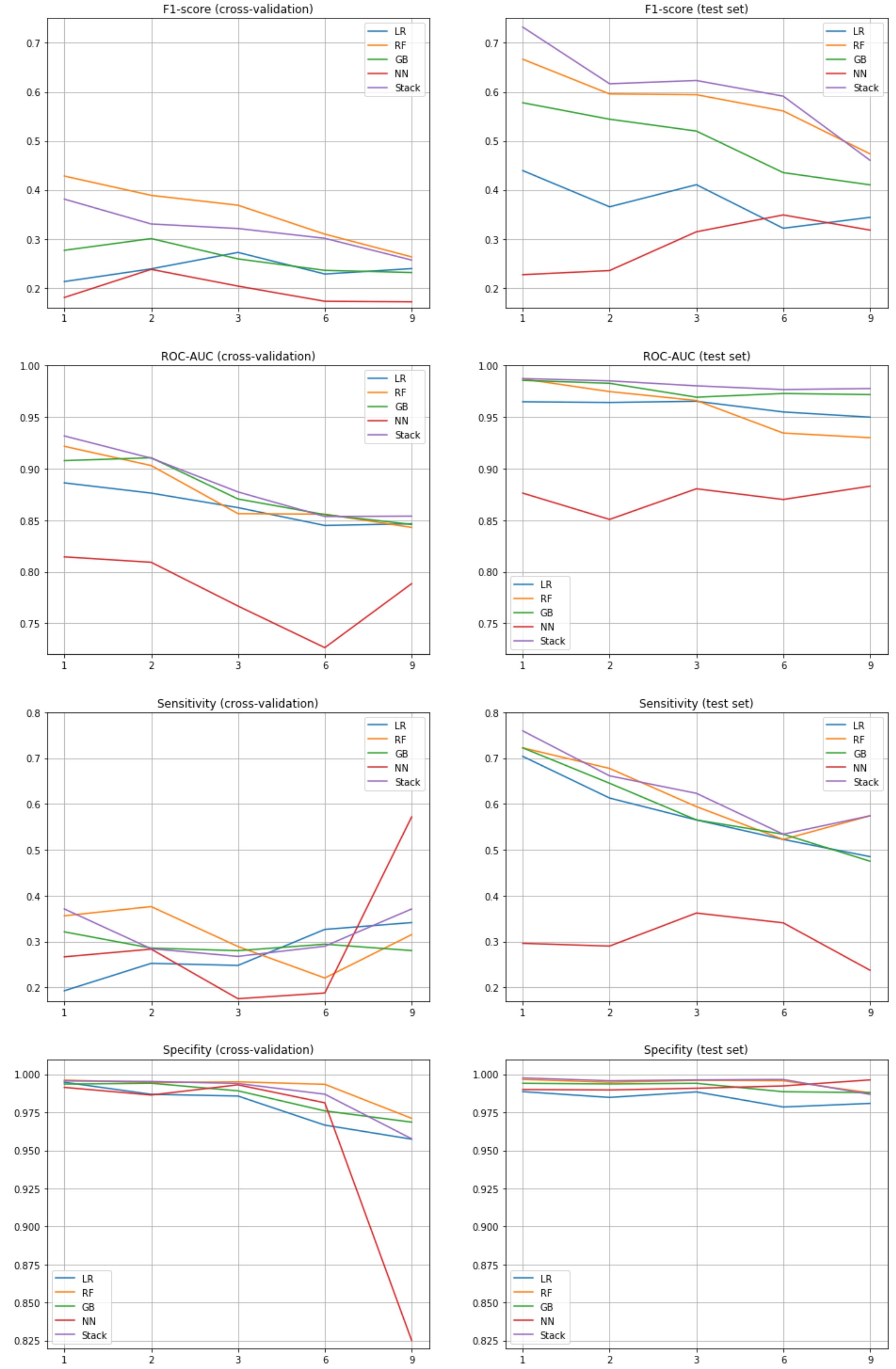

Figure A2. Metrics on cross-validation and test sets in the forecasting of the violation of capital adequacy ratio

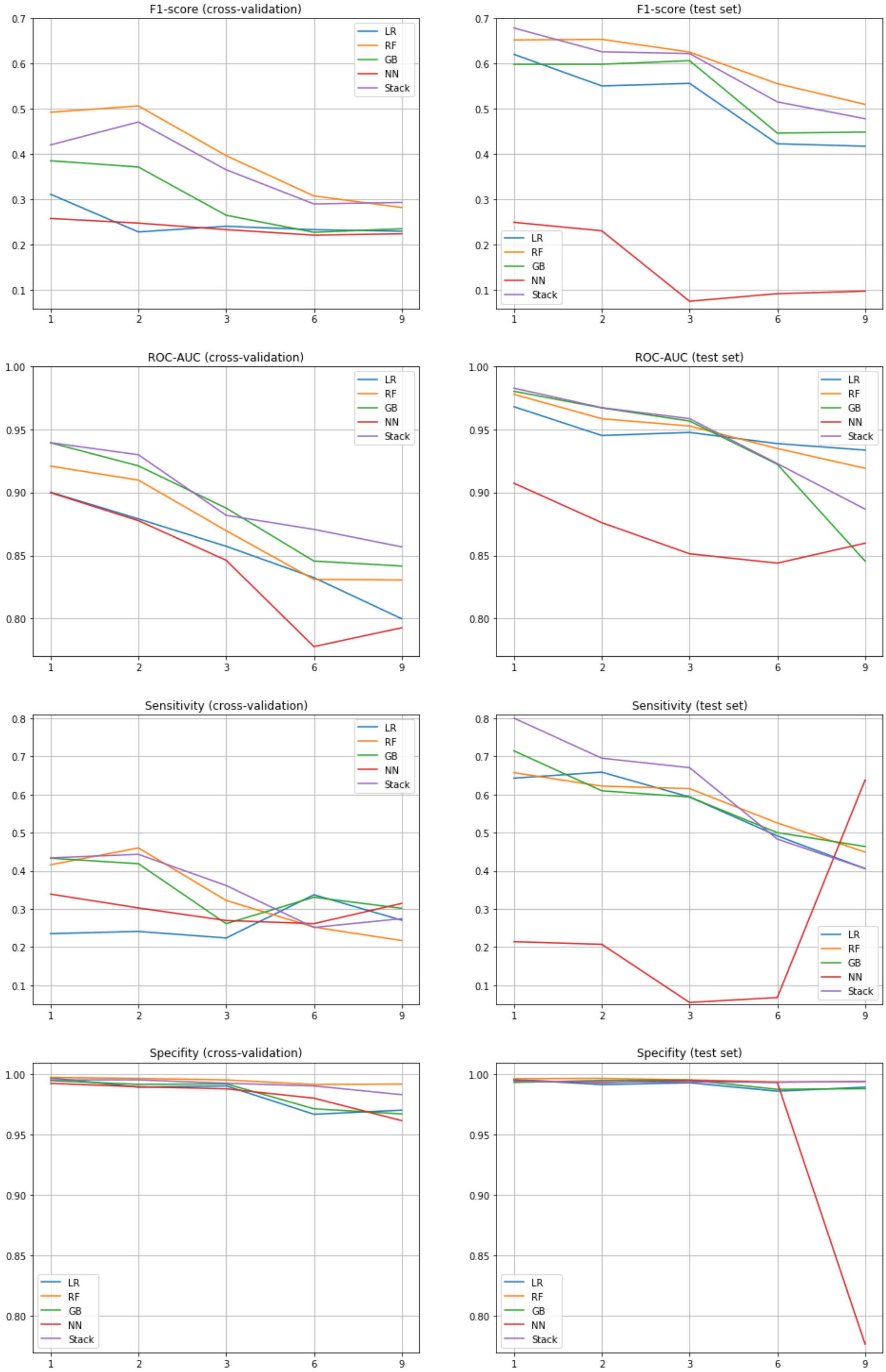

Figure A3. Metrics on cross-validation and test sets in the forecasting of the violation of common equity Tier 1 ratio

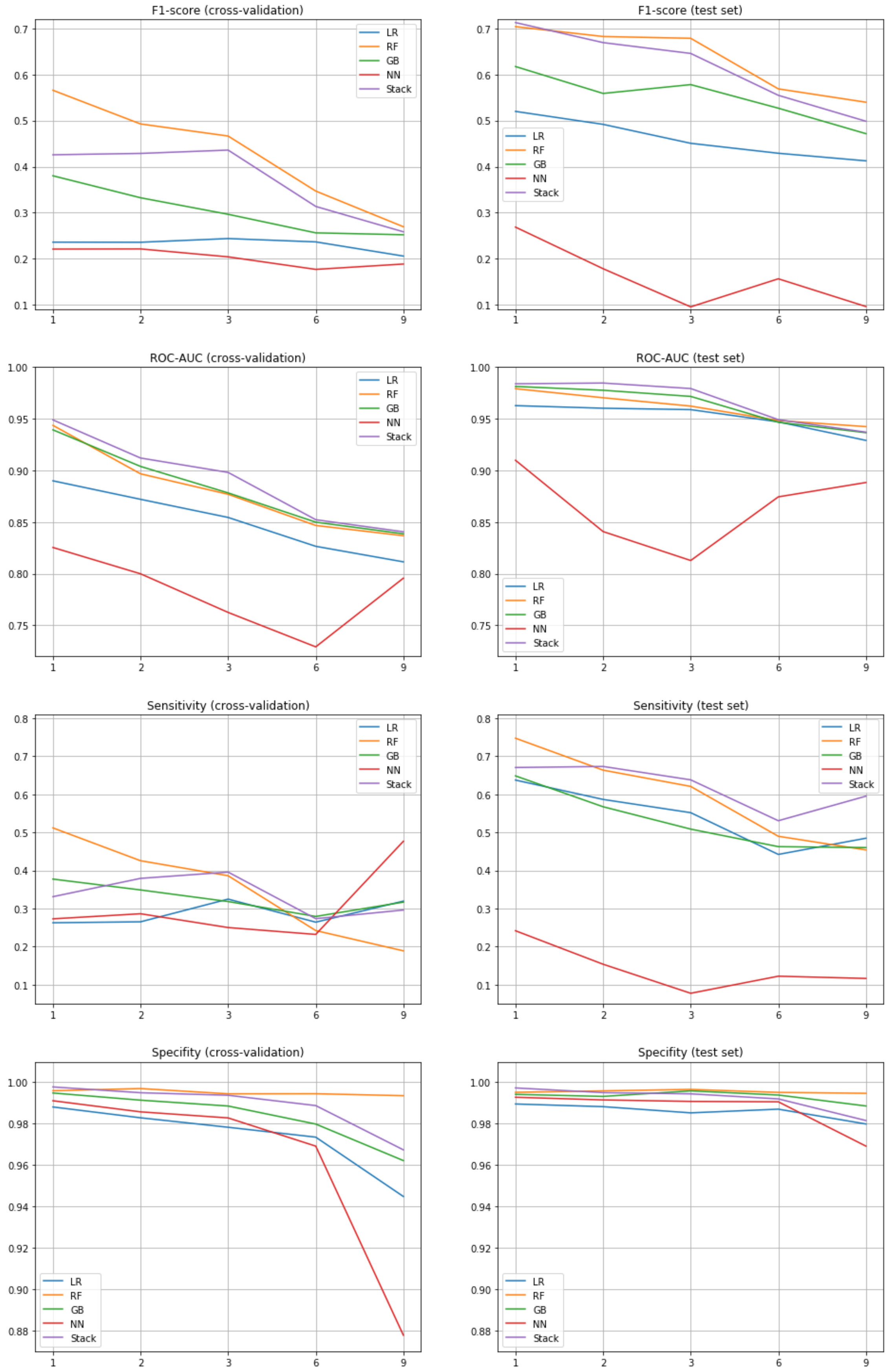

Figure A4. Metrics on cross-validation and test sets in the forecasting of the violation of Tier 1 capital ratio

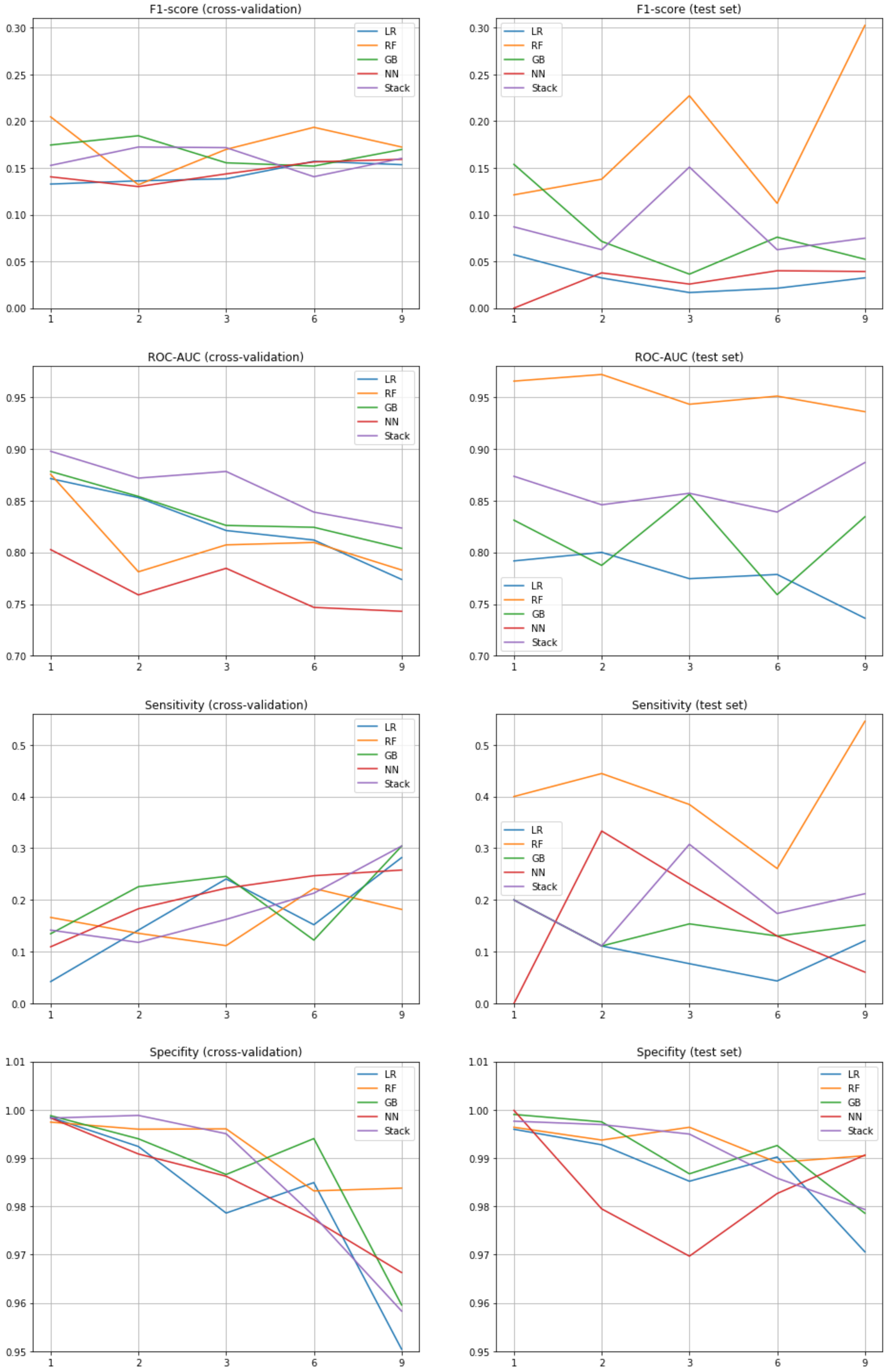

Figure A5. Metrics on cross-validation and test sets in the forecasting of the violation of the instant liquidity ratio

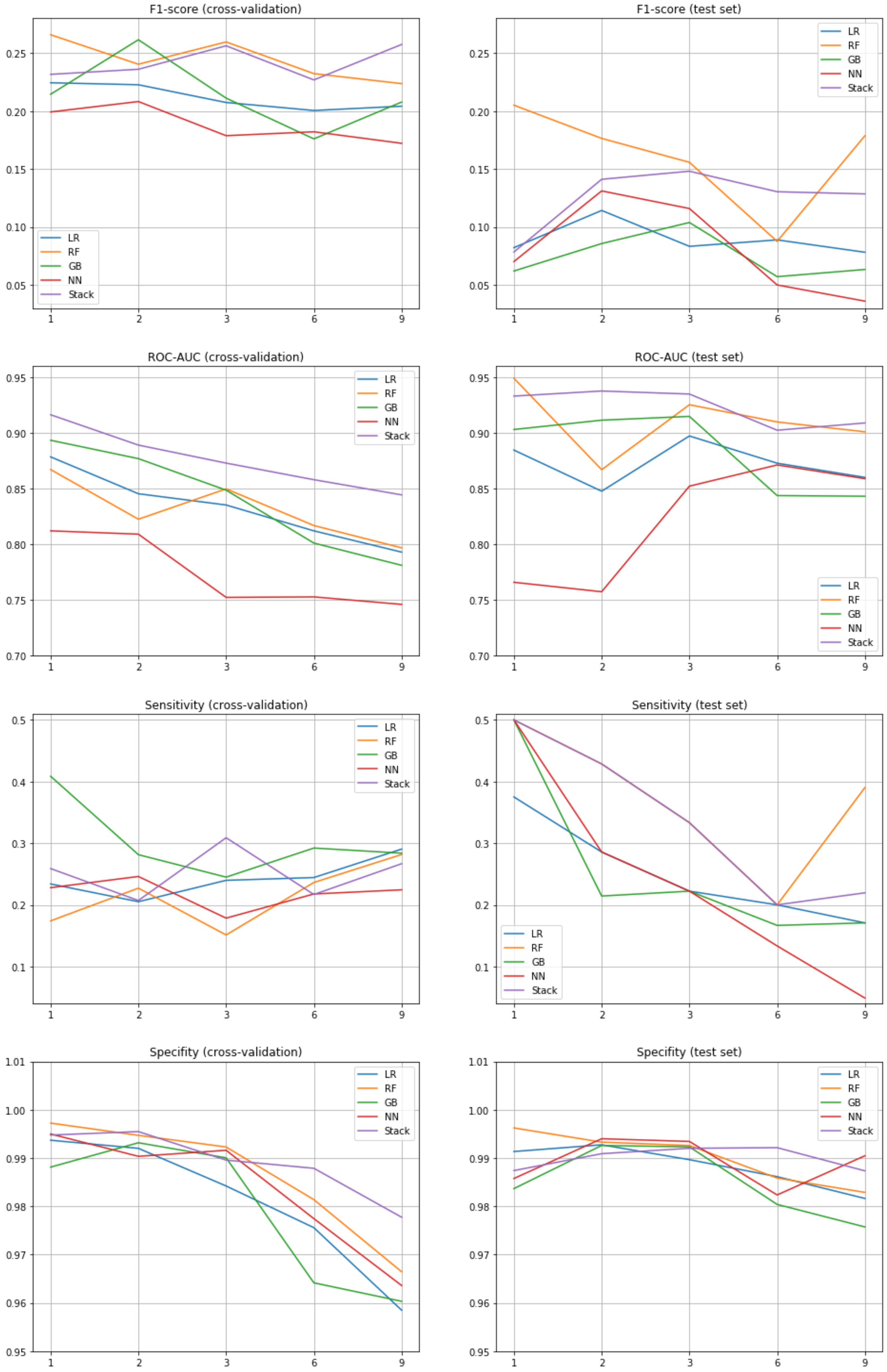

Figure A6. Metrics on cross-validation and test sets in the forecasting of the violation of the current liquidity ratio

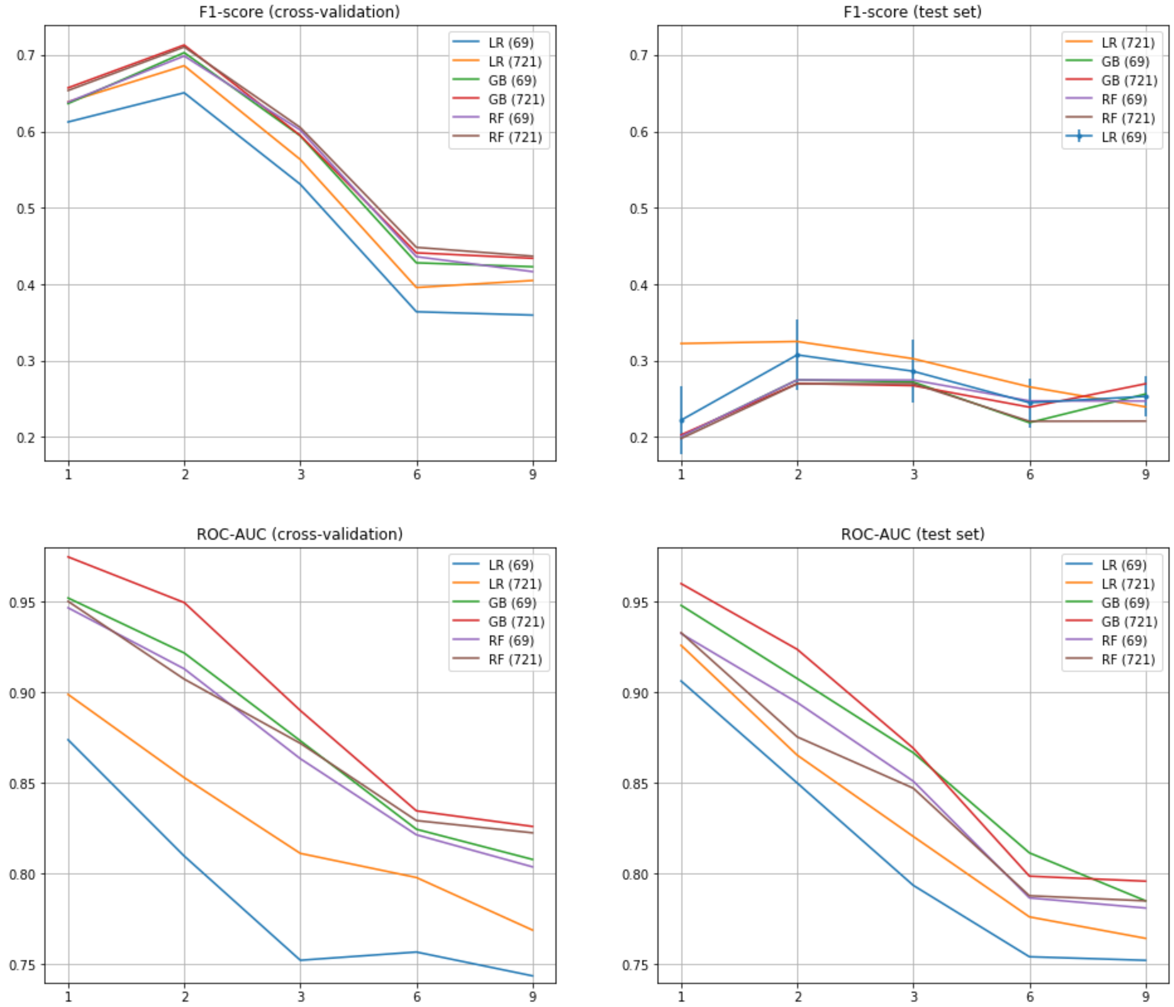

Figure A7. Comparison of metric values depending on the number of features in the forecasting of license withdrawal