NATIONAL RESEARCH UNIVERSITY HIGHER SCHOOL OF ECONOMICS

International College of Economics and Finance

Лукьянов Матвей Александрович

Lukianov Matvei Aleksandrovich

BACHELOR'S GRADUATE QUALIFICATION WORK

BACHELOR GRADUATION PAPER

МОДЕЛЬНЫЕ РИСКИ В ЗАДАЧАХ ОЦЕНКИ РЕЙТИНГА ВЕРОЯТНОСТИ ДЕФОЛТОВ

MODEL RISKS IN DEFAULT PROBABILITY ESTIMATION PROBLEMS

Field of study: 38.03.01 «Economics»

Degree programme: Double degree programme in Economics of the NRU HSE and the University of London

Supervisor
Associate Professor

_____
Lapshin V. A.

Moscow 2022

Content

Abstract

This paper aims to estimate the significance of preprocessing techniques for the classic classification models in the retail clients' default estimation problems. For this goal there was analyzed which significance test of comparison of model performance would be the most accurate. There were obtained some significant results. Firstly, there was established that "point632" error, which results from bootstrap procedure, leads to the most accurate test. Secondly, with the help of this error there was visualized and tested the performance of classification models with different preprocessing. It was obtained that KNN demands the highest degree of preprocessing, including scaling, One-Hot encoding, handling imbalanced data and transformation, but with the help of these steps it can show relatively good performance compared to other models. On the contrary, LDA show its best without any preprocessing steps. Logistic and gradient boosting models are moderately dependent from preprocessing, at the same time gradient boosting model showed significantly better performance compared to other classification models.

Аннотация

Целью данной работы является оценка статистической значимости методов предварительной обработки для классических моделей классификации в задачах оценки дефолта розничных клиентов. Для этой цели было проведено исследование, какой именно критерий оценки значимости различий в качестве моделей является наиболее точным. Были установлены следующие результаты: во-первых, было установлено, что ошибка “point632”, которая подсчитана методом бутстрэпа, является несмещенной оценкой генеральной ошибки и приводит к наиболее точным тестам статистической значимости; во-вторых, с помощью данной ошибки была протестирована и визуализирована работоспособность моделей с разной предварительной обработкой. Было выяснено, что KNN требует наибольшего числа методов предварительной обработки, включая “One-Hot” кодирование, масштабирование, обработку несбалансированных данных, преобразование независимых переменных. С помощью данных методов модель KNN продемонстрировала хорошее качество предсказаний относительно других моделей. Наоборот, LDA показывает себя лучше всего без каких-либо методов предварительной обработки. Логистические модели и модели градиентного бустинга умеренно зависят от предварительной обработки, в то же время модель градиентного бустинга показала значительно лучшее качество предсказаний по сравнению с другими моделями классификации.

Introduction

Model risks is one the most rapidly developing topics in today's risk management. These risks cannot be called traditional types of risks like market, credit or liquidity risk, what is more, its borders are vague and are still not well defined. This risk shows how certain model imperfections can cause credit organizations (mainly banks) lose some part of potential of profit. By imperfections of model I mean mainly conscious omissions of some steps of model pipeline by such reasons as a lack of time, lack of opportunities, or expectation about the inefficiency of certain step of model pipeline. Thus, since models are used in all actions of modern credit organizations, model risk is the integral part of any credit organization. Moreover, if this risk is omitted, it will be counted in other types of risk, so we will get biased results, which can be wrongly interpreted. Summarizing all above, it is undoubted that both academic and professional sphere have growing interest in this topic. Academically it is important to formally define this risk, summarize its components and analyze its behavior. While in professional sphere clear understanding of model risks will help to avoid possible mistakes in model setup and in estimating the model, avoid over calculating and unnecessary steps of the model, what will optimize the modeling costs.

The conceptual framework of my paper is banks or microfinance organizations making loans to mainly retail clients. So, the object of my paper is retail clients, taking loans from credit organizations. The subject of my paper is the model risk, which arises during the estimation of probability of default of these consumers.

The goal of my paper is to propose the ways of testing model performance (including ones, using bootstrap and cross-validation), compare them and find out which of them offer the most accurate testing results. As a result, it will be possible to fulfill the second goal of my paper: for each classification model to understand what steps of model pipeline can be indeed skipped because of inefficiency, or, in other words, how the model risk can be minimized in modern credit organizations in retail sector.

To achieve the goal several steps should be performed. First of all, there should be made research with aim to analyze and compare the methods of testing the performance of models, considered in the recent significant academic papers: cross-validation, nested cross-validation, bootstrap. As a result, there will be proposed a method with the most accurate testing, possibly offering a freedom of interpretation and visualization of its results. Thus, this research will have theoretical academic significance.

Another step is to gather the most popular datasets to estimate the model, understand, what preprocessing steps and classification models are used by researcher, get the information about the

metrics used and analyze the comparative performance of models (get baseline values of metrics) with the help of literature review of the corresponding papers. Namely, there will be a systemization of the main components of model risk in retail banking, existing in academic papers.

Further, there should be made a preparation of practical part with description of methods of research used. This part of research will have an aim to classify existing models, estimating the probability of default, understand possible advantages and disadvantages of analyzed models. There should be made a full description of perfect model pipeline - description of the main steps of model pipeline and what is their aim in the model. Main blocks of pipeline are preprocessing (dealing with missing values, scaling, encoding of categorical variables, feature transformation, feature selection, handling imbalanced data), tuning hyperparameters and model training, and, finally, evaluating the model's performance (choice of metric, confusion matrix).

The last step is to carry out the practical steps of research: searching, collecting, preparing and making documentation of datasets with worldwide retail clients. Further, there should be made a stepwise omission of steps of the model to find what are the main factors of model risk so that to understand which modules of pipeline are indeed significant and which can be skipped. Here formal test and visual analysis of distribution of metrics is needed to draw certain conclusions. Consequently, there will be a summary of the paper results, proposing the ways, how this information can be used in modern credit organizations to manage model risk more successfully. Thus, this paper will have practical significance as well.

My main hypothesis is that indeed some computationally and time-consuming steps of the model pipeline can be skipped to optimize our final model design, which can be proved using appropriate significance test. This inefficiency can take place, because some steps can duplicate the effect of each other and hence their contribution to model risk is insignificant.

Literature review

In academic sphere there is a constantly growing interest towards machine learning technics in credit risk management. It led to the fact there was written dozens of papers each of which used different sets of datasets for training, different data preparation technics, different models and different metrics to estimate the model results. As a result, systematic review of these papers was needed, since in industry of risk management it was ambiguous what exactly models to use.

One of such systematic reviews was proposed by (Lousada, 2016), where credit scoring papers from 1991-2015 were analyzed. There was summarized main trends of machine learning technics towards risk management.

Speaking about the data ingestion there was obtained that the most popular datasets are Australian and German datasets, which present the anonymous credit data about retail clients and the information, whether the clients default on the credit or not. Nearly 40 % of considered papers used these datasets. Thus, it makes sense to use at least one of these datasets to understand the relative performance of the model pipeline, compared to other models, fitted by other researchers.

Speaking about data preparation, the researcher gets that the most popular technic in this sphere is feature selection (more than 50% of the papers), while missing values imputation was relatively unpopular (nearly 10% of papers).

Speaking about models used the researcher found out that although neural networks are extremely popular in recent papers (used in 21%), classic models like logistic, linear regressions, discriminant analysis are still popular and effective and occur in nearly 50% of the papers, so these classics technics should be considered as well in my analysis.

Speaking about model evaluation, researcher get that K-fold cross is the most popular method of validation methods (nearly 45% of papers) and the most popular metrics are metrics in confusion tables as accuracy, recall and precision; and ROC Curve metrics (45% and 30% respectively).

The further extension of the analysis was made in the paper (Markov, Seleznoyva, 2021), where the results were supported by the modern papers for 2015-2021 period. Among the recent papers that were considered and which offered some specific model pipeline extensions are (Carta, 2020), (Tripathi, 2020) and (Ashoften, 2021).

Salvatore Carta in his work considered entropy-based techniques, which allows the model to divide the training sample on reliable and non-reliable observations, what helps the model to ignore the non-reliable sample. This approach allows the researcher to get relatively high point estimate of Gini of 0.78 on German dataset and of 0.82 on Australian dataset.

Diwakar Tripathi in his research analyze performance of neural networks on credit scoring datasets. Specifically, he proposed an activation function, which allows to introduce evolutionary approach in neural networks behavior. This approach allows the researcher to get 0.7 point estimate of Gini on German dataset and 0.9 point estimate of Gini on Australian dataset.

Finally, Afshin Ashoften in his research analyze conservative approach towards machine learning technique, when model is formed using Kruskai-Wallis non-parametric static and compare the performance of classic models, like logistics regression, and neural network models in this conservative paradigm. This approach allows him to get 0.85 Gini point estimate for both logistic and neural networks in the German dataset, what can be considered as outstanding result, especially in case with logistic regression.

However, all described papers, described above, and the majority of papers summarized by Lousada, present only the point estimate of performance of their model – usually it is the mean result of Cross-validation criterion on 10 folds. So, no tests on significance difference of performance of their models were presented, what can be considered as significant drawback of recent papers.

The problem of the test of significance difference of model performance is well-known in the machine-learning community and there were published a sufficient number of papers, analyzing this problem. The main problem here is the problem of independence of resulting performance metrics of models. One of the popular validation methods is k-Fold Cross-validation technique – the method of model validation, when all data is divided in k folds, model is trained k times on $k-1$ folds and is estimated on the remaining fold, every time a different fold is treated a test set, what helps to get rid of overfitting problems, when errors of the models are underestimated because of non-generalization of the model. As a result, each model will have k estimated metrics, there can be computed means of these estimates, standard deviation and t-test on difference of means of two samples can be taken. However, it was noted by (Dietterich, 1997) that this kind of tests can give underestimated p-values, what can lead to the fact that researchers will overestimate the significance of relative performance of their models. It was explained by Thomas Dietterich by the fact that the key assumption of the t-test was violated – samples tested should be independent, while in Cross-Validation samples are dependent: the training samples overlap, what leads to the fact that each pair of training folds overlap by $\left(1-\frac{2}{k}\right)*100\%$ observations. As a result, cross-validation underestimates variances of metrics and lead to incorrect tests.

There are certain ways how to correctly compare models: one method is the use of nested Cross-Validation, for example, Dietterich advices to repeat 5 times 2-fold Cross validation, what decreases the dependence of samples, since training set are not overlapping anymore.

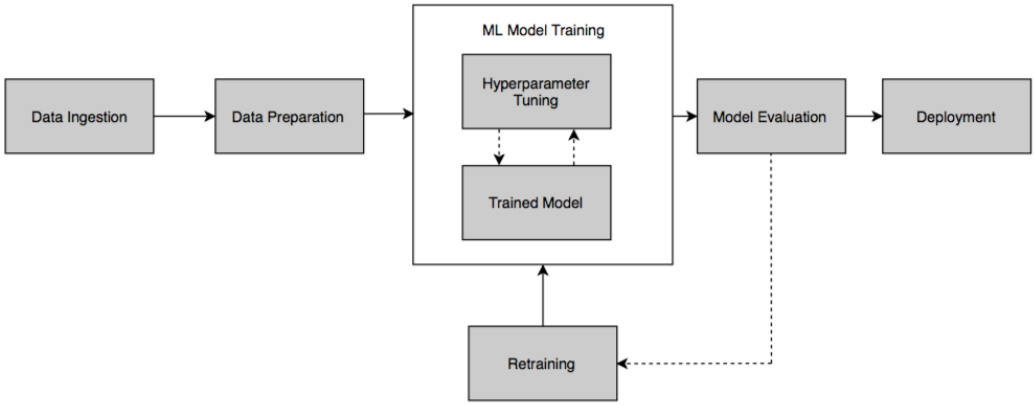
Another approach to get correct intervals and significance tests was proposed in (Efron, 1983), (Efron, 1986) and in (Efron, 1995). There he offers to use bootstrap procedure – technique when from sample with n observations n observations are randomly taken with replacement k number of times and model is trained on n such observations; errors are estimated on the observations which were not taken. In (Efron, 1983) there was offered to use the point632 estimate, which is the linear combination of train error and the out-of-bag error (error on the observations, which did not get in the train set), obtained by the bootstrapping. It was shown that this estimate has lower variance than cross-validation estimates and can be used for testing the significance of difference of models' performance.

The methods discussed above were recently combined in the (Bates, 2021), where another specification of nested Cross-validation was proposed by researchers from Berkely and Stanford universities. There original algorithm of cross-validation was nested (smoothed, so that it has lower variance) with the help of bootstrap procedure.

In this paper there will be used method proposed by Bradley Efron, since with bootstrapping there can be taken correct significance tests and there is vast opportunity of visualization of model performance. As a result, in this way there can be analyzed the significance of preprocessing steps in the model pipelines and there can be compared the relative performance of different models' specifications.

Methods of research

In this chapter there will be described in detail the methods which I will use in my research. To start with, a good machine learning model can be described as pipeline - a set of subsequent blocks, in each of which we try to solve some specific type of problems. Usually there can be distinguished the following pipeline's blocks: Data ingestion, Data preparation, Model Training and finally Model Evaluation. Possible general view of model pipeline can be summarized in the picture below:



Picture 1. General form of ML model pipeline

Source: Sunith Shetty, 2018[1]

Each should be described separately. In this chapter there will be analyzed in details Data Preparation, Model Training, Hyperparameter tuning and Model Evaluation block. The info about Data Ingestion you can find in the next chapter "Data description".

Data preparation

The main goal of this block is to "prepare" data in such way that it can efficiently be used by the model. So, we translate the initial data to the language that the model will understand the best. One moment should be pointed out: each preprocessing step is fitted at the same data, where the model is trained. Then all data (including test and train) is transformed using this fitted preprocessing. For example, in standard scaling the mean, which is subtracted from each value, is calculated as the mean of the train sample. This is made to prevent leakage – when model gets some information from test set on the training set, what violates the main assumption of test data – it is the set, which should be completely hidden from the model up to evaluation stage. Further you can find several preprocessing methods (namely, modules) that are used in this paper.

[1] Retrieved from https://hub.packtpub.com/automl-build-machine-learning-pipeline-tutorial/

Dealing with qualitative data

The first possible problem is that model can work only with quantitative variables (even if in the guide of some model it is stated that model works successfully with qualitative data, be assured that instead this model is a small pipeline, which indeed inside transform quantitative data to qualitative). So, the question is how exactly we would transform quantitative data to qualitative: there are certain ways to perform it:

1. One-hot encoding, when we create $n-1$ new dummy columns with zero and ones, where n is the number of unique values in the initial quantitative column. One of the pluses of this method is that all information is transmitted, but the drawback is that we significantly increase the dimension of our model, what can lead to increased number of time for fitting the model, moreover, there can be a case of overfitting.
2. Label encoding, when we create one column: for each unique value of initial column we design some new integer value. The advantage of this method is that our model will be parsimonious. However, while implementing this method, we assume some order of initial values and the difference between them, what can lead to lose of some information, what can bias results of the final model. What is more, there is a question, which values exactly to impute instead categorical values: in general, there will be no info of ranking between the categorical values, so there should be made some manual work, what contradicts the goal of making the modeling process as automized as possible. Thus, in this research there will be used WoE method of encoding categorical values. It is not pure label encoding, since it imputes not integer values, but its mechanism is similar: it replace one categorical column by one numerical column. Specifically, it replaces categorical features by their Weight of evidence value, which is calculated by the formula:

$$WoE = \ln\left(\frac{number\ of\ non-defaults\ in\ the\ category}{total\ number\ of\ non-defaults}\right) - \ln\left(\frac{number\ of\ defaults\ in\ the\ category}{total\ number\ of\ defaults}\right)$$

In the analysis there will not be tested the significance of this module as a whole: firstly, model could not work without any encoding of categorical features and if instead the difference between the model with using of categorical features and the model, using only numerical features is tested, it will be the test on the significance of categorical features, but not the significance of the module. Thus, here there will be tested One-Hot encoding vs Label encoding: there will be checked whether the parsimony, created by label encoding, can compensate the loss of some information.

Data Scaling

There are some models (like KNN) which demand that each feature has the same scale, so that all features have the same influence in model. For example, for linear model different scaling is not a problem, but even in this case of this model the results of the mode can be improved, since the data will become more homoscedastic. Possible scaling technics are:

1. Standard scaling, when mean is subtracted and the result is divided by standard deviation of the feature, hence it is sometimes called z-transformation:

$$z = \frac{x - \bar{x}}{sd(x)}$$

   Where $\bar{x}$ is the mean and sd(x) is standard deviation

   As a result, all features in the dataset will have 0 mean and 1 standard deviation, what can benefit the model.

2. Minimax scaling, when minimal value is subtracted and the result is divided by (maximal and minimal values difference). As a result, all values will be in the range of 0 and 1, what again can be more comfortable for the model.

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}$$

3. Robust scaling, when median value is subtracted and the result is divided by IQR. The advantage of this method is that this scaling is robust (what follows from its naming) and is insensitive to outliers, what can work extremely well in heterogeneous data.

$$z = \frac{x - \text{median}(x)}{IQR(x)}$$

   Where IQR is the difference between the third (75% percentile) and first (25% percentile) quantile;

   Median is the second quantile (50% percentile).

In the research there will be tested the significance of this module in the situation with no scaling at all vs the model with best scaling in terms of performance on the evaluation stage.

Data Transformation

It is possible that data behaves in a more complex way than a model assumes. For example, linear model assumes linear dependence of target value and features, but this dependence can be quadratic or of a higher polynomial degree. So, there can be a case that an increase in the complexity of the model will drop the bias of the model more than increase of its variance if the concepts are analyzed in bias-variance tradeoff paradigm. In this case transformation of the model can be used – a mechanism when some feature transformation will help to obtain a more complex

model without changing the model basement and assumptions. For example, for linear model it is important indeed that the model is linear in coefficients, since in this case even if logarithm transformation of the original feature is used, the model can still be fitted.

Another possible aim of data transformation is to make data less skewed, more normally distributed and to stabilize variance, what can help some models.

Among transformers that that will be considered in this research:

1. Box-Cox Transformation. It belongs to power transform family. One of the possible drawbacks of this method is that original features should be positive, what decreases the applicability of this method. The formula of transformation is:

$$z = \begin{cases} \frac{x^{\alpha} - 1}{\alpha}, & if\ \alpha \neq 0 \\ \ln(x), if\ \alpha = 0 \end{cases}$$

   Where α is the power coefficient, which estimated by the likelihood function estimation
2. Yeo-Johnson Transformation is another transformer from the family of power transformers. Its advantage, compared to Box-Cox Transformation is that it is applicable for zero and positive values of features. The formula of transformation is:

$$z = \begin{cases} \frac{(x+1)^{\alpha} - 1}{\alpha}, & if\ \alpha \neq 0, x \geq 0 \\ \ln(x+1), if\ \alpha = 0, x \geq 0 \\ -(-x+1)^{2-\alpha} + 1, if \alpha \neq 2, x < 0 \\ -\ln(-x+1), if \alpha = 2, x < 0 \end{cases}$$

3. Winsorizer transformation is technique which allows to fight with the outliers, working in the following way: using likelihood function the method finds the optimal window to look for outliers, for example, from 5% to 95% percentiles. All data outside these intervals will be equalized to the percentiles, found previously.
4. Log Transformer – takes logarithm of feature. It has the same drawback, as Box-Cox Transformation: all values should be positive, what limits its application.

In the research there will be tested the significance of this module in the situation with no transformation at all vs the model with best transformation in terms of performance on the evaluation stage.

Feature selection

After the feature engineering (or after original data ingestion) there can be a lot of features, what can lead to model overfitting and will require a dozen of time to train the model. So, some technics should be used to select the features:

1. Best subset selection, when algorithm runs through all possible set of features and select the model based on minimization of AIC or BIC criteria:

$$AIC = -2\ln(L) + 2p$$

$$BIC = -2\ln(L) + p * ln(n)$$

Where p is the number of features used, n is the number of observations, L is the likelihood function. The first terms of these function response for goodness of fit of the model – how well it predicts the data behavior, while the second term penalize the model, which has too many features and, thus, overfit.

The problem of this method is that it should run through 2^p, where p is the number of original features, what grows extremely fast with the increase in number of features. For example, in dataset with 21 features there should be fitted more than 2 million models, what can be really time-consuming. Thus, in this research there will be used further methods.

2. Forward subset selection, when null model without any features is fitted and then 1 feature is added to the model based on the simple likelihood criteria (there is no aim to use AIC, since all models here have the same number of features). Then the algorithm is repeated for p number of features. As a result, there will be $p + 1$ best models, that we can compare based on AIC or BIC criteria. The advantage of this algorithm is that only $1 + \frac{p(p+1)}{2}$ of models are fitted, so it grows only with quadratic speed. For example, in case of 21 original features, only 232 models should be fitted instead of 2'097'152 in BSS
3. Backward subset selection looks similar to FSS, the only difference is that firstly full model is fitted and then 1 feature is deleted from it, based on likelihood criteria. Then algorithm is repeated for p number of features and there will be $p + 1$ best models, that we can compare based on AIC or BIC criteria.

Dealing with imbalanced data

The problem, which is specific to the context of this research is the problem of imbalanced data. It occurs when in the target value there are too little observations of some class relatively to other classes, what leads to the situation when model has a desire to predict in all cases the second class over the first. For example, in credit samples there are usually 5% of observations, which are defaulted, so if model predicts that all observations are not defaulted, then it will get an accuracy of 95% what can be viewed as good result in other cases, but not in this specific case: using this model, bank will give credits to all potential consumers, what will surely result in the bankruptcy of the bank. There are the following solutions of the problem:

1. Undersampling: makes classes balanced by randomly deleting values from the original data with overrepresented class. The obvious drawback of this method is that it drops data, which can be possibly valuable in decision making.
2. Oversampling: makes the classes balanced by taking randomly with replacement more observations from the original data with underrepresented class. The drawback of this methos is that, in fact, some points are heavier than other points, what can bias the model, since the data becomes unrealistic.
3. SMOTE. This method is the further development of the oversampling model. It uses bootstrap method, but now adds some noise, which depend on the central of mass of underrepresented class, so that points that are far from center of mass do not become outliers (indeed, for each observation it finds its nearest neighbor, and put new point randomly on the line between these points). So, the advantage of this algorithm that data is not duplicated, new synthetic data is generated, which is slightly different form the original.
4. ADASYN models. The further extension of SMOTE. Now it generates examples in such way that more new data is generated in such places, where it is harder for the model to learn, so the probability of generating a new point is higher in the area, where the density of underrepresented class is relatively low.

In the research there will be tested the significance of this module in the situation with no imbalance data handling at all vs the model with best imbalance data handling technique in terms of performance on the evaluation stage.

Model training

Choice of model is one of the most important steps in the model pipeline. There are various models, which can be used starting from classic classification models, which will be considered in this chapter later and ending with neural networks. There is certain evidence that neural networks show a better results in classification problems (Tripathi, 2020), however, they will not be considered in this research by the two main reasons. Firstly, neural networks have a low degree of feature interpretability: they can give lower errors than classic methods, but there is no information, what helps to reduce these errors, it is unknown what internal relationships between features helped to get lower errors, what can be unacceptable in econometric research. So, some part of the information that we want to get from the model is hidden (even the layers of neural networks are called “hidden”), what can be crucial for external regulators (like Central Bank of Russia), which demand from banks that all the steps of the models which influence the decision-making process is open. Secondly, neural network, especially the most advanced one like convolutional or recurrent neural networks can be themselves considered as the whole model pipeline, since they

by construction can make some preprocessing like convolution or pooling, so they do not suit the aim of my paper to connect the steps of pipeline, mainly preprocessing, and model training.

In the following parts there will be reviewed classification models, which offer a good level of interpretability.

Logistic regression

Usually, linear regression is the starting point for all machine learning not because of the fact that there are a lot of phenomena in the world that behave linearly (usually they do not), but because linear function is a reasonable approximation of all continuous and differentiable functions.

However, in the problems of classification it has relative drawbacks, for example, the target predicted by this model can be higher than 1 or less than 0 what is impossible to interpret, since the probability must be in the range of 0 and 1. That is why the first model which will be considered is logistic model, which cures some drawbacks of linear regression. In logistic regression logistic function is used:

$$p(X) = \frac{\exp(X^T * \beta)}{\exp(X^T * \beta) + 1}$$

Where X is vector of p predictors and 1: $X = (1, X_1, X_2, \dots, X_p)$

β is the vector of $p + 1$ coefficients: $\beta = (\beta_0, \beta_1, \beta_2, \dots, \beta_p)$

And indeed, it can be easily shown that the target in this case is limited by the range of 0 and 1:

$$\lim_{X \to -\infty} \left(\frac{\exp(X^T * \beta)}{\exp(X^T * \beta) + 1} \right) = 0$$

$$\lim_{X \to \infty} \left(\frac{\exp(X^T * \beta)}{\exp(X^T * \beta) + 1} \right) = 1$$

$p(X)$ is monotonically increasing function, hence target indeed belongs to the interval from 0 to 1.

Linear Discriminant analysis

LDA uses Bayes' theorem to classify objects:

$$p_k(x) = \frac{\pi_k f_k(x)}{\sum_{i=1}^{2} \pi_i f_i(x)}$$

Where π_k is the prior probability of belonging to k^{th} class (can be obtained from real distributions or we can use theoretical priors, like Jeffrey's priors).

f_k is the density function of X, p_k is the posterior probability of belonging to k[th] class

LDA assumes normal distribution of f_k:

$$f(x) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left((-0.5(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

Where Σ is the Covariance matrix of X and μ is the mean of X

Plugging it in the Bayes' formula, there can be obtained that observation is assigned to the class for which the value of below function is lowest:

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + log\pi_k$$

From which it can be derived that the decision boundary is linear.

At this point LDA and Logit models should be compared: both of them have linear decision boundary. However, LDA assumes the normal distribution of training features, so logistic regression can outperform it in case the assumption of normality is violated.

KNN

KNN is one of the oldest of machine learning models. This model is non-parametric, contrary, to LDA and logistic regression, what means that it has low degree of interpretability, namely there will not be obtained the importance of features, used by the model. At the same time, because of the fact that it does not assume linearity, it can predict much more complex data distribution, since its decision boundary is not linear (but piecewise linear). This model has rather simple mechanism of work: for externally specified value of k (which is the hyperparameter of this model) the model finds for each observation its k closest neighbors (based on Manhattan or Euclidian distance – it is other model's hyperparameter) and classify the observations based on the average class of its k closest neighbors:

$$p_i(Y = j|x) = \frac{1}{K}\sum_i I(y_i = j)$$

The final decision boundary will be piecewise linear, which results in the diagram called Voronoi diagram.

Trees and Gradient Boosting

Gradient boosting model is a complex model, which I am going to use in my work as reference model. It is based on trees, so analysis of this model should be started with trees.

Classification tree is the method, which divide the feature space by axis aligned binary splits. In each of such subspaces every row, belonging to this region, is classified as the most often occurring class in this subspace. Usually as a metric for split it uses classification error rate – the algorithm runs through the sample of possible splits, calculate the possible error and choose the split, which results in lowest classification error. Trees have a plenty of hyperparameters that can be tuned: maximum depth of the tree - the maximal number of consecutive binary splits; minimal number of observations in the node: if the number of observations in the node is too low, then there is no aim in further splitting; maximum number of features to analyze while making the decision of split. All of these hyperparameters have the aim to make the tree not overfitted: if the tree grows too deep and too much, then it is possible that it will memorize the training data, perceiving the white noise as some real dependence. All of these hyperparameters can be successfully transferred to boosting models.

The main aim of boosting is to combine somehow various tree – make the ensemble of models – and to get the better predictions from them than from a single tree. The boosting mechanism is the following: first, it grows a single simple tree, which can be rather small, and then the next tree learns from the errors, obtained by the first model. So, the procedure of boosting is sequential fitting of small trees, which learns slowly, but with big number of iterations, the predictions can be improved significantly. Additionally, to the hyperparameters, inherited by the boosting from trees, it has the number of trees hyperparameter – the higher is the number of trees, the better possibly the predictions of the model, but the higher is the possibility of overfitting; and shrinkage parameter, which determines the rate at which boosting model learns – its aim is to decrease the variance of the model by increasing the bias a little.

In the research there will be used Light Gradient boosting machine (LGBM) package to implement the boosting model, so in the further work this model will be called LGBM.

Model evaluation

First of all, there should be specified what metrics should be used. There are 2 main types of metrics in classification problems: metrics based on confusion matrix and metrics based on ROC curve.

The confusion matrix is the matrix, where there are specified all possible outcomes of our classification task:

- true positive (TP), the number of outcomes where model correctly classify positive class;
- true negative (TN), the number of outcomes where model correctly classify negative class;
- false negative (FN), the number of outcomes where model wrongly classify negative class

- false positive (FP), the number of outcomes where model wrongly classify negative class

And there are lots of metrics that follows from this matrix:

- $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ – the most popular metric, but the problem is it is overoptimistic with imbalanced data, which is the case of datasets with defaults data: for example, if there are 5% of defaults in data, accuracy will show only 5% error, if model predicts that all observations are non-defaults, so absolutely non-working model gets 95% accuracy, what is misleading
- $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$ – two metrics, which should be targeted in imbalanced dataset: precision shows how model can find the positive classes in general and recall show how the model can distinguish positive class from other classes. The problem is that the model cannot target both metrics together, so the researcher have to set threshold on some metrics and target the other metrics, what prevents the automatization of process. Another method is to combine both these metrics in another target, using F1 score or Balanced accuracy
- $F1\ score = \frac{2*precision*recall}{precision+recall}$ – this metric is the harmonic mean of recall and precision
- $F1\ score = \frac{sensetivity+specificity}{2}, where\ sensetivity = \frac{TP}{TP+FN}, specifity = \frac{TN}{TN+FP}$

The main disadvantage of metrics based on confusion matrix is that there should be chosen threshold of probability, given by the model, when the class are classified as 1 instead of 0. Usually this threshold is 0.5, since it is a reasonable value of border probability, but this does not mean that this threshold will always be optimal. For this problem there was designed a ROC Curve -the curve in the coordinates (sensitivity, 1- specifity), which is drawn by varying the threshold parameter. This curve goes through the points (0,0) and (1,1), it is monotonically increasing and the closer it goes to the (0,1) point, the better is the model, if this curve goes linearly from (0,0) to (1,1) then the model could not correctly classify the objects and takes the decision by random. Metrics that follows from ROC curve:

- ROC AUC which can be only in the range of 0.5 and 1, since is the area between the curve and the line, which goes through (0,0) and (1,1) points. The intuition of this metric in the default estimation setup is that it shows the probability that some defaulted observation will have a higher chance to be defaulted (by the prediction of model) than the non-defaulted object. The good thing about this metric is that it is stable towards imbalanced target

- GINI is just the linear transformation of ROC AUC, so that this metric was in range between 0 and 1

$$Gini = 2 * ROCAUC - 1$$

In my work there will be targeted Gini metrics, since it is stable towards imbalanced classes, and it is independent of the choice of threshold. What is more, there will be analyzed the behavior of Balanced Accuracy and F1 score metrics, since they take into consideration the imbalanced problem, are easier interpretable and offer an opportunity to look at the classification error from a different angle.

Another question that should be discussed is the validation method. In the literature review it was discussed that K-fold cross validation approach, which is the most popular method of model validation, leads to unbiased errors, but these errors will have underestimated variability due to the violation of independence of samples. Thus, in this research there will be used point632 method, which is the linear combination of train error and the out-of-bag error, obtained by the bootstrapping:

$$Err_b^{632} = (1 - 0.632)\overline{err} + 0.632 * \widehat{Err_b}$$

Where $\overline{err}$ is the train error, obtained by the model, if it is trained on the whole sample

And $\widehat{Err_b}$ is the out-of-bag error, obtained on the bth bootstrap sample, b = (1, 2, …100)

The bootstrap process is the following: for b times (100 in this research) n observations are randomly taken with replacement from sample with n observations and model is trained on n such observations; out-of-bag errors are estimated on the observations which were not taken in each of bootstrap samples. Thus, there will be obtained 100 errors, distribution of which can be estimated and tested.

The intuition of the point632 method is the following: $\overline{err}$ is the trained error, which often underestimates the real error, because of overfitting problem: since model trained and tested on the same datasets, it started to memorize even the random noise of the data, thus overfitting. $\widehat{Err_b}$, on the contrary, overestimates the error: model now sees only some part of the observations, what is more, some observations have heavier weight, because of the fact that after bootstrap some observation will be taken more times than other. And if this heavier points were originally outliers or so called “unreliable” points, then the error will be overestimated. Thus, taking the right linear combination of underestimated and overestimated error, we can get an unbiased estimator of general error. The only question is why exactly 0.368 vs 0.632 proportion was taken. The answer

is that 0.632 is the limit probability that some observation belongs to b^{th} bootstrap sample: $\lim_{N\to\infty}\left(1-(1-\frac{1}{N})^{N}\right)=1-e^{-1}\approx 0.632$

Which is true by the second wonderful limit, where N – is the number of observations in the sample.

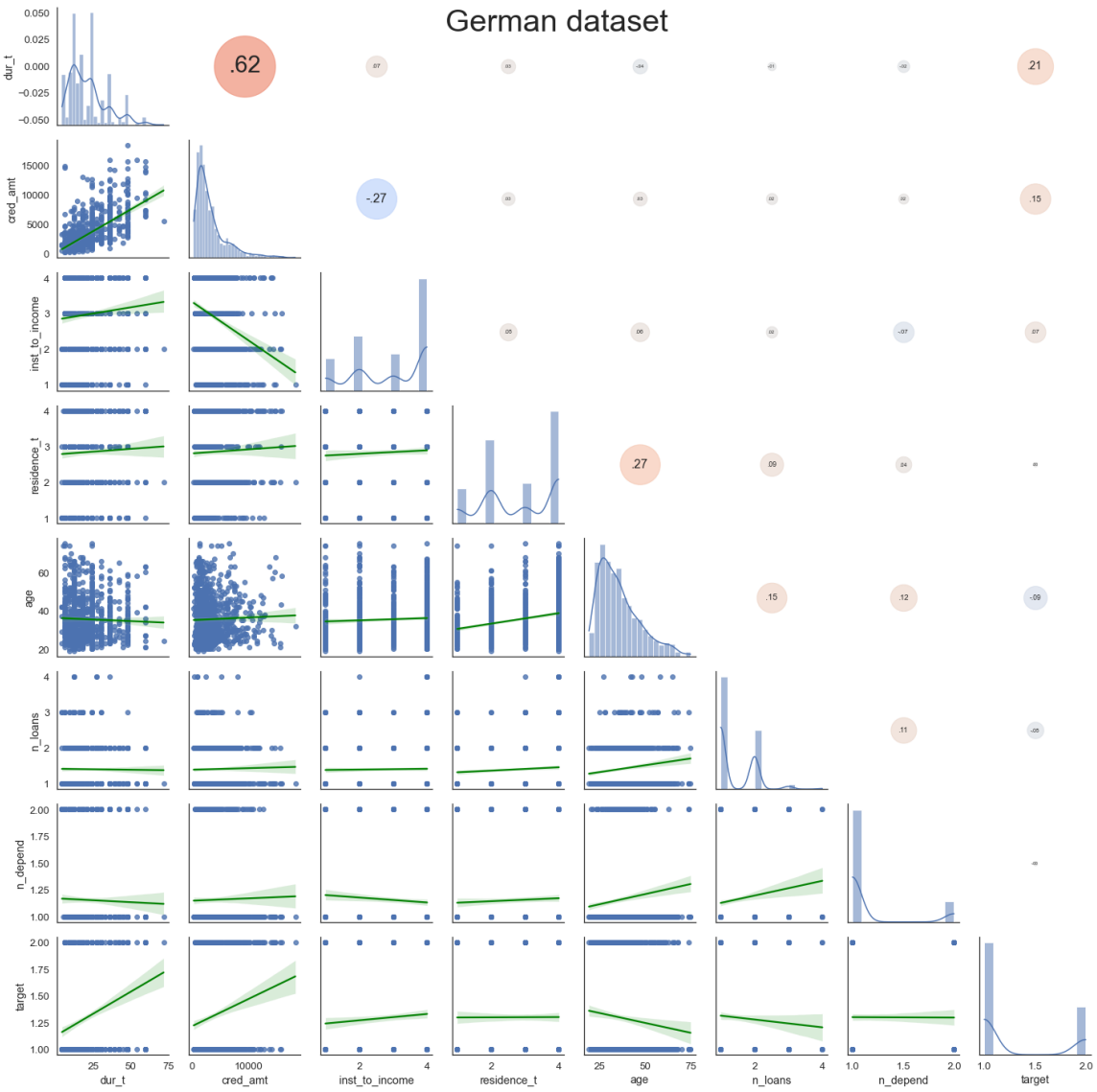
Hyperparameters tuning

It is important to explain the hyperparameters tuning process. For all 4 considered models there is its own hyper optimization.

In each model pipeline hyperparameters of the model and the modules of the pipeline are optimized: in each module the optimizer goes over all possible preprocessing methods and select one of them or neither. The target of the optimizer is the mean point632 error of Gini metrics. From the technical point of view there was used Hyperopt package, instead GridSearch: the difference is that Gridsearch goes over all possible values of hyperparameters and modules of pipeline, while Hyporopt uses ‘smart’ optimization, goes over only those hyperparameters, which it finds possible candidates for global maximum.

Data description

The research will be conducted with the help of open retail credit dataset, namely German dataset[2]. It is one of the most popular datasets in default estimation problems, but rather small: only 1000 observations and 20 features and 1 target. Here target is the quality of credit: 1 for good credit and 2 for bad credit. The target is relatively imbalanced: only 0.3 of observations are marked by 'bad' credits mark, so module on imbalanced data can be useful. There are 7 numerical features and 13 categorical features, in the picture below you can see the histograms of each numerical feature, correlation between each pair of features and finally mutual graph of each pair of features.



Picture 2. The mutual graphs, histograms and correlations of numerical features of German

Source: calculations of the author

[2] Retrieved from https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

From this graph there can be seen that the 3 features that have the highest absolute correlation with the target are *dur_t*, *cred_amt* and *age: dur_t* has the 0.21 positive correlation, which means that on average the higher the mean duration of the credit, the higher probability that it is bad, *cred_amt* has 0.15 positive correlation, which means that on average the higher the credit amount, the higher the probability that the credit is bad, and finally the age has negative correlation of -0.09, which means that the lower the age, the higher the probability that the credit is bad.

What is more, we can see that almost all distributions are far from normal: for example, duration, age and credit amount has negative skewness, so some kind of transformation can be useful.

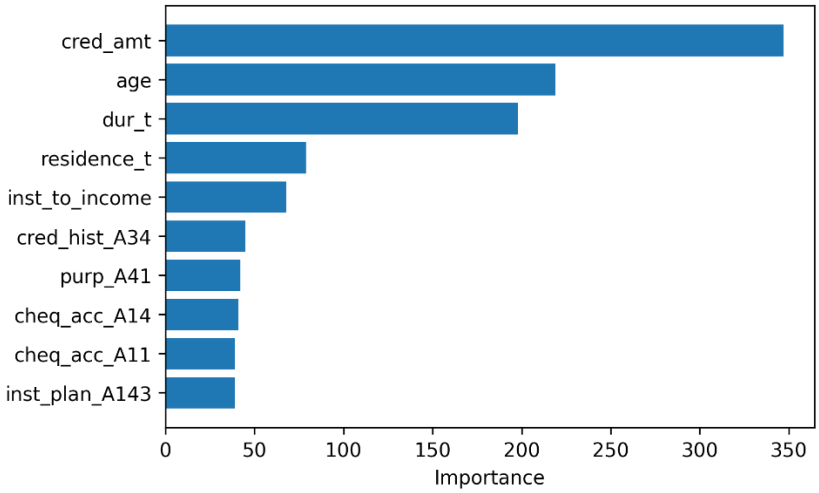
Testing of model performance

The goal of this chapter is to analyze the results of models and preprocessing modules described in the previous chapters and to compare the performance of models based on evaluating technics presented in the previous chapters.

LGBM

In the space of all possible models and preprocessing modules the model, which shows the best point estimate is the Light Gradient Boosted Machine Classifier with OneHot encoding in the encoding module, Robust Scaling in the scaling module and YeoJohnson Transformer in the transformation module, which shows 0.696 Gini point estimate. It is interesting that modules on features selection, dimension reducer and Imbalanced data are not included in this 'perfect' model, which means that worse results are shown with them. In case with imbalanced data I can explain this by the fact that target is not too heavily imbalanced (0.3 vs 0.7) and that this model successfully manages with this problem (further in KNN model there can be seen that Imbalanced module is needed to obtain better estimates). Speaking about selection and dimension reducers, it can be concluded that all features were significant in the initial dataset (which is not surprising, since the initial dataset was rather small and was cleaned from the unnecessary features from the dataset).

Speaking about model interpretability, in the picture below there can be seen the graph of feature importance, where top 10 important features are depicted. The importance of a feature is calculated as the number of splits made by the certain feature. So, the graph shows the features by which the model split the dataset most of the time (by the x-axe there is the number of splits).



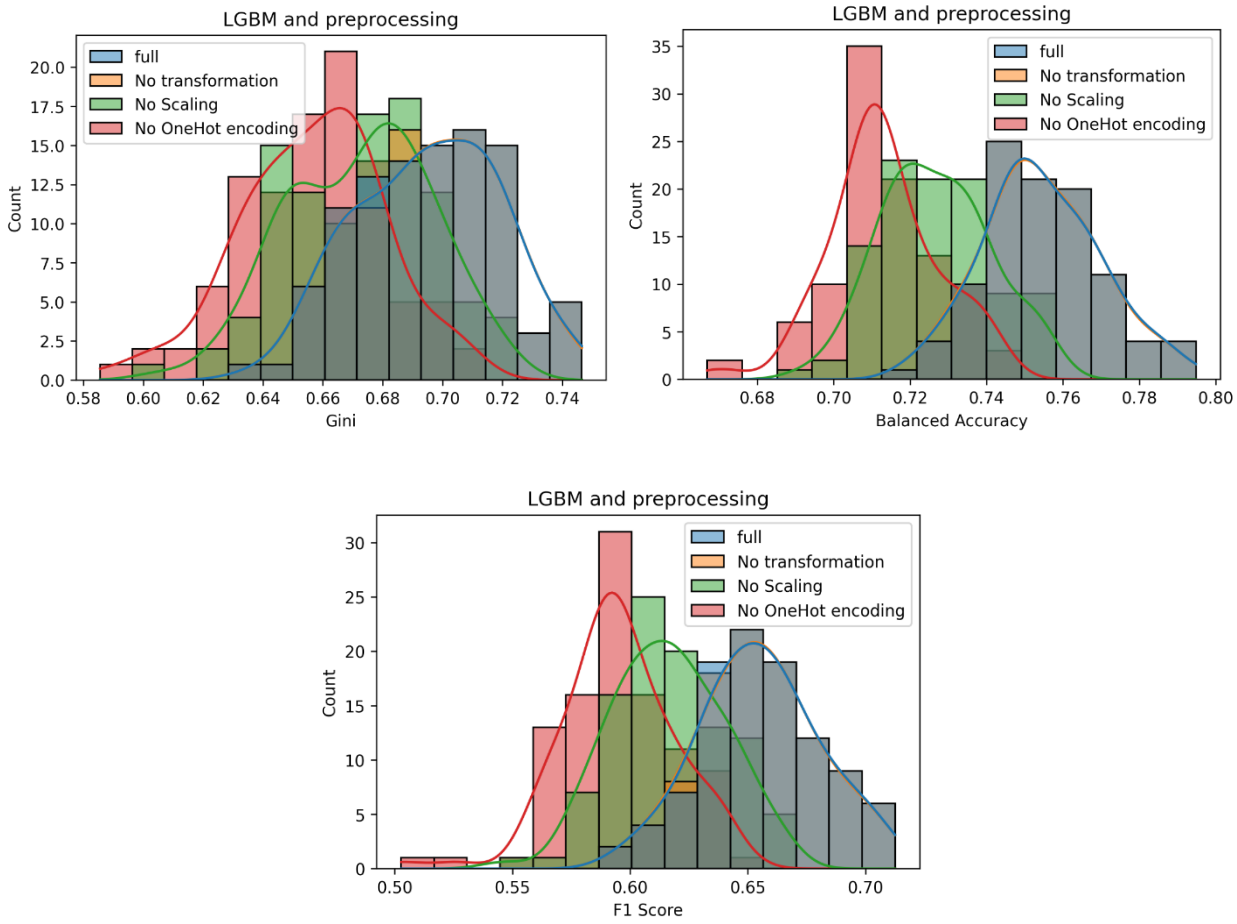
Picture 3. Feature importance graph by LGBM model

Source: calculations of the author

The following conclusions can be made from the graph: the most significance features are *cred_amt*, *age* and *dur_t*, what is supported by the EDA analysis in Data description chapter, where it was found out that these features have the most absolute correlation with the target feature. What is more, we can see that some categorical features, encoded by OneHot encoding, are important as well, so there can be predicted that the model using different encoding can get significantly worse results, but this should be tested further.

The next step is to evaluate the contribution of each of modules, which the model uses.

In the histograms below there can be seen distribution of 3 key metrics: Gini, Balanced Accuracy and F1 score, obtained using point632 estimation method resulting from bootstrap technique.



Picture 4. Distribution of point632 bootstrap errors of 3 key metrics, produced by LGBM model with varying preprocessing.

Source: calculations of the author

There can be seen that transformation module here has the lowest contribution in the LGBM performance, the differences are hardly measurable and should be tested. Other modules have much higher contribution, but it can be seen that the contribution of One-hot encoding is higher

than scaling. In the table below there can be find the key descriptive statistics of distributions means, standard deviations and the significance of difference of means compared with the full, using t-test for the means of two samples, without the assumption on equal variances.

	Full model pipeline	No scaling	No transformation	No OneHot encoding
Gini	0.696 (0.0236)	0.672 *** (0.0238)	0.695 (0.0234)	0.687*** (0.0236)
Balanced accuracy	0.755 (0.0154)	0.727*** (0.0142)	0.754 (0.0154)	0.712*** (0.0143)
F1	0.655 (0.0256)	0.615*** (0.0237)	0.655 (0.0256)	0.595*** (0.0240)

Significance. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '°' 0.1 ' ' 1

Table 1. Key descriptive statistics of point632 bootstrap errors of 3 key metrics, produced by LGBM model

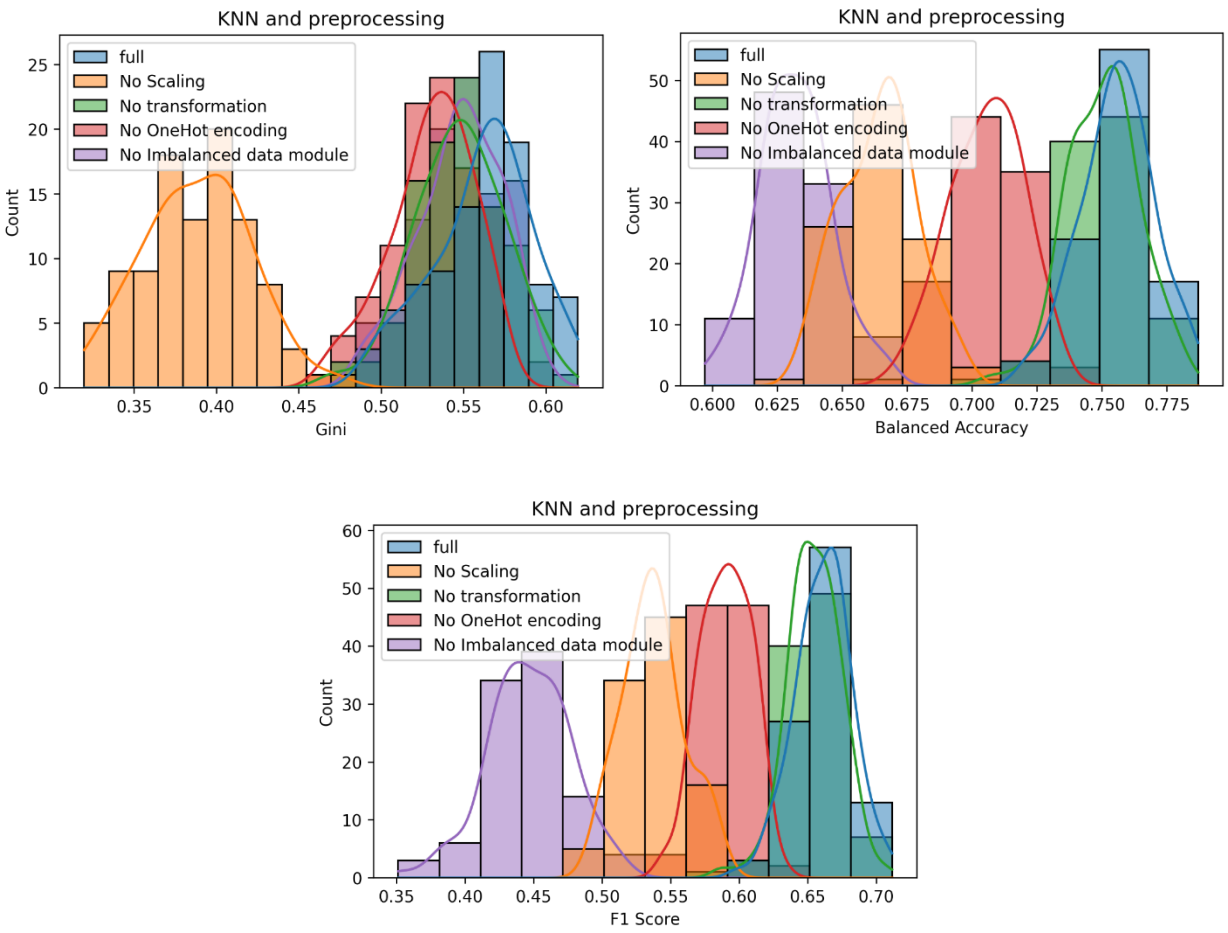
Source: calculations of the author

Indeed, there can be seen that the contribution of the transformation module is insignificant in all 3 metrics, while the contribution of Scaling and one-hot encoding modules is significant with the p-value close to zero.

KNN

The second model, which was considered is KNN. Tuning of hyperparameters show that Scaling (namely, Standard Scaling), transformation (namely, Einsorizing transformation), Onehot encoding and imbalanced data (namely, ADASYN) modules were chosen in the full model, which means that all of these modules contribute some value to the model. On the contrary, feature selection module was not chosen in hyperparameter tuning, which means that they lower the target metric.

In the histograms below there can be seen distribution of 3 key metrics: Gini, Balanced Accuracy and F1 score, obtained using point632 estimation method resulting from bootstrap technique.



Picture 5. Distribution of point632 bootstrap errors of 3 key metrics, produced by KNN model with varying preprocessing.

Source: calculations of the author

There can be seen that scaling and imbalanced data modules benefit the model the most. However, visually the contribution of other modules is significant too, but this hypothesis should be tested further. In the table below there can be find the key descriptive statistics of distributions: means, standard deviations and the significance of difference of means compared with the full, using t-test for the means of two samples, without the assumption on equal variances.

	Full model pipeline	No Scaling	No transformation	No OneHot encoding	No Imbalanced data module
Gini	0.570 (0.0277)	0.337*** (0.0298)	0.534*** (0.0230)	0.531*** (0.0252)	0.521*** (0.0256)
Balanced Accuracy	0.730 (0.0145)	0.628*** (0.0135)	0.701*** (0.0134)	0.705*** (0.0142)	0.646*** (0.0138)
F1 Score	0.616 (0.0192)	0.486*** (0.0206)	0.583*** (0.0182)	0.590*** (0.0183)	0.483*** (0.0257)

Significance. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '°' 0.1 ' ' 1

Table 2. Key descriptive statistics of point632 bootstrap errors of 3 key metrics, produced by KNN model

Source: calculations of the author

Indeed, there can be seen that all 4 modules are considered are significant by the tests. This means that KNN demands high degree of data preprocessing: 4 modules give a significant increase in its performance.

LDA

In case with LDA there is an interesting situation: tuning process shows that no module should be chosen, which means that the model shows its best performance in case of no preprocessing (except for One-hot encoding) is performed, what can be considered as interesting result.

In the table below there can be seen key statistics of performance of this model.

	Gini	Balanced Accuracy	F1 Score
LDA	0.590 (0.0207)	0.680 (0.0136)	0.545 (0.0220)

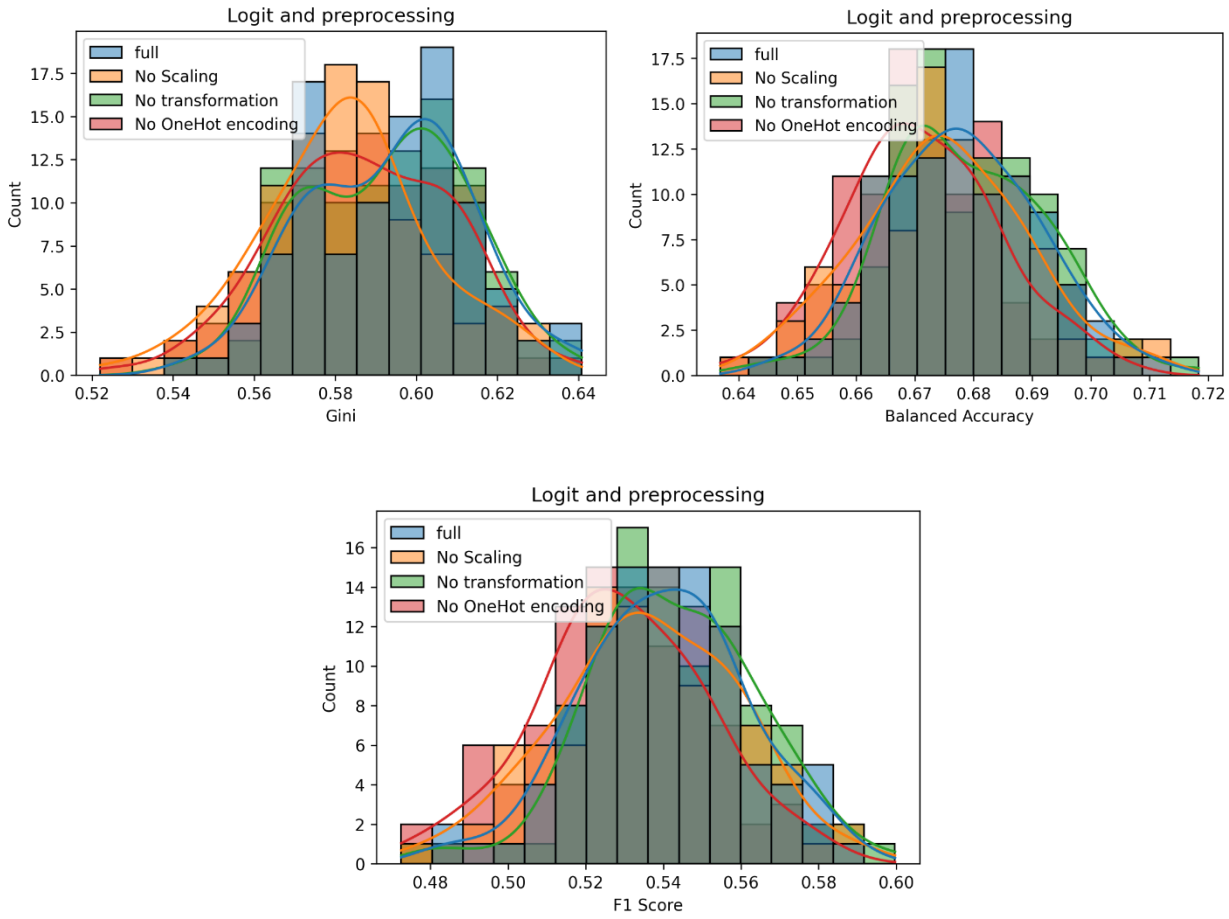
Table 3. Key descriptive statistics of point632 bootstrap errors of 3 key metrics, produced by LDA model

Source: calculations of the author

Logistic regression

Logistic regression tuning shows that Scaling (namely Robust scaling), transformation (namely, Winsorizing method) and one hot encoding modules have positive contribution in the model performance. On the contrary, feature selection modules and handling imbalanced data module were not used in the final model, hence they bring negative contribution to the model.

In the histograms below there can be seen distribution of 3 key metrics: Gini, Balanced Accuracy and F1 score, obtained using point632 estimation method obtained using bootstrap technique.



Picture 6. Distribution of point632 bootstrap errors of 3 key metrics, produced by Logit model with varying preprocessing.

Source: calculations of the author

There can be seen that the situation with significance of modules in logistic regression is hard: visually it is hard to tell which modules are significant. In the table below there can be find the key descriptive statistics of distributions: means, standard deviations and the significance of difference of means compared with the full, using t-test for the means of two samples, without the assumption on equal variances.

	Full model pipeline	No scaling	No transformation	No OneHot encoding
Gini	0.593 (0.0201)	0.582 *** (0.0209)	0.592 (0.0201)	0.587* (0.0213)
Balanced accuracy	0.677 (0.0129)	0.674 ° (0.0142)	0.678 (0.0133)	0.671*** (0.0131)
F1	0.539 (0.0212)	0.535 (0.0233)	0.542 (0.0212)	0.529*** (0.0221)

Significance. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '°' 0.1 ' ' 1

Table 4. Key descriptive statistics of point632 bootstrap errors of 3 key metrics, produced by Logit model

Source: calculations of the author

From the table it can be concluded that transformation module is insignificant in all 3 metrics. Scaling is significant in Gini metrics, but hardly significant in Balanced accuracy and F1 metrics (p-value is bigger than 0.05). Speaking about one-hot encoding, it is significant in all 3 modules. As a result, logistic regression demands a moderate number of preprocessing modules: only one-hot encoding bring a significant positive contribution to the model performance in all 3 considered metrics.

Model comparison

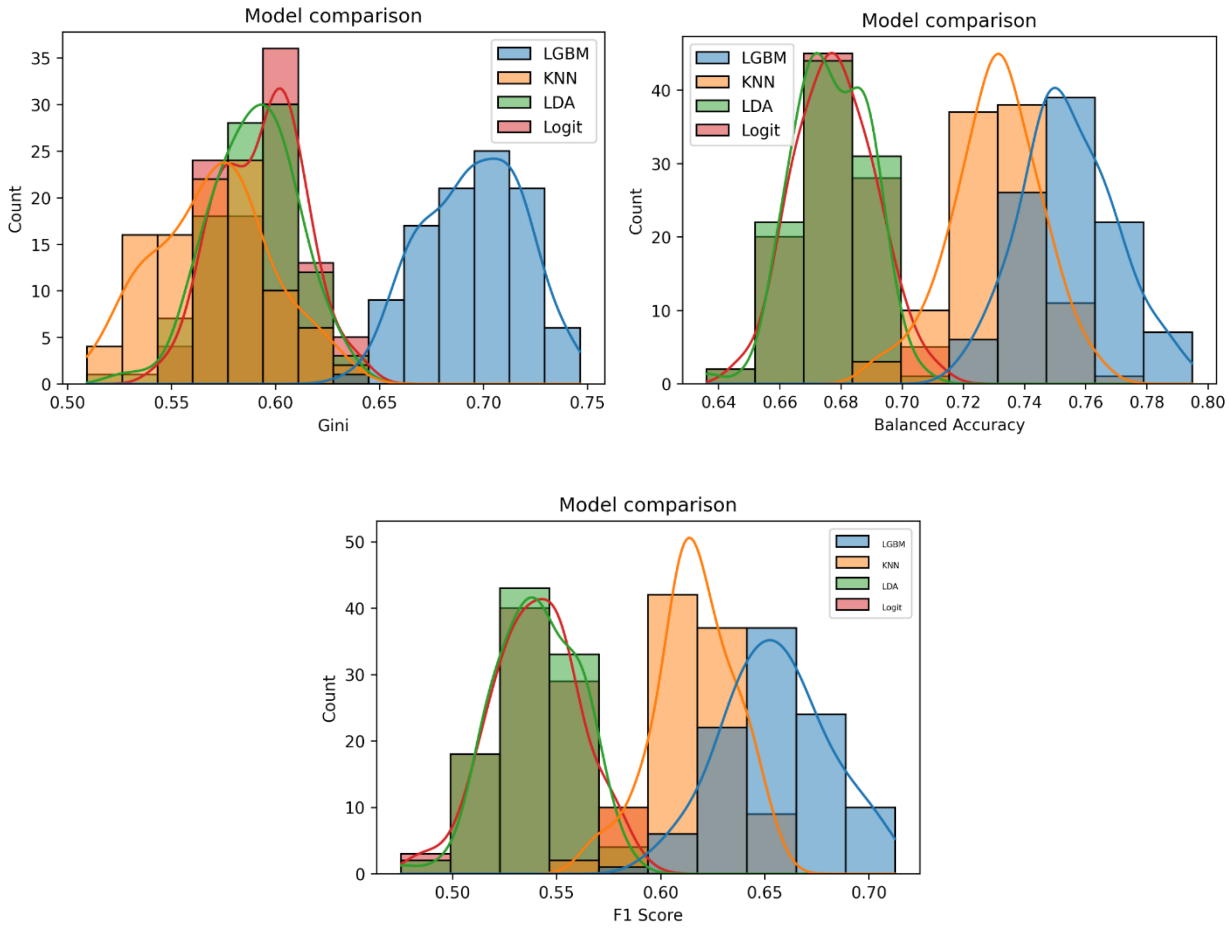
Finally, there is collected enough data to compare the performance of models, each having the optimal modules in their pipelines. In the table below you can find the composition of these full models. '!' symbol means that the contribution of the module to the model's performance is insignificant by all 3 considered metrics, while '— ' symbol means that the module brings negative contribution to the model's performance by Gini metric

	LGBM	KNN	LDA	Logit
Scaling	Robust scaling	Standard scaling	—	Robust scaling
Transformation	Yeo-Johnson transformations !	Winsorizing	—	Winsorizing !
Encoding	One-hot	One-hot	WoE	One-hot
Handling imbalanced data	—	ADASYN	—	—
Feature selection	—	—	—	—

Table 5. The preprocessing modules used in each model in its ultimate performance

Source: calculations of the author

In the histograms below there can be seen distribution of 3 key metrics: Gini, Balanced Accuracy and F1 score, obtained using point632 estimation method resulting from bootstrap technique.



Picture 7. Distribution of point632 bootstrap errors of 3 key metrics, produced by 4 considered models

Source: calculations of the author

Here interesting results can be seen: KNN visually looks inferior to LDA and Logit model: it has lower mean and higher variance, what can be considered as vital drawback in model risk. However, in Balanced Accuracy and F1 score metrics KNN significantly outperforms LDA and Logit, what can be explained by the fact that it has imbalanced data module in its pipeline.

What is more, it can be seen that LGBM significantly outperforms other models visually in all 4 metrics.

In the table below there can be find the key descriptive statistics of distributions: means, standard deviations and the significance of difference of means compared with the best performing model (LGBM), using t-test for the means of two samples.

	LGBM	KNN	LDA	Logit
Gini	0.695 (0.0237)	0.570*** (0.0277)	0.590*** (0.0209)	0.593*** (0.0201)
Balanced accuracy	0.754 (0.0153)	0.730*** (0.0146)	0.677*** (0.0121)	0.678*** (0.0129)
F1 score	0.655 (0.0256)	0.616*** (0.0192)	0.541*** (0.0199)	0.540*** (0.0212)

Significance. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '°' 0.1 ' ' 1

Table 6. Key descriptive statistics of point632 bootstrap errors of 3 key metrics, produced by 4 considered models

Source: calculations of the author

From table there can be seen that indeed LGBM significantly outperforms all other 3 models in all 3 metrics.

Conclusions

There were obtained two main groups of conclusions: theoretical and practical ones. Speaking about theoretical conclusions, in literature review there was analyzed main techniques of testing of model performance. Indeed, there was stated that nested cross-validation and bootstrap methods offer the estimates that have an accurate estimate of mean and standard deviation of errors, so these methods can be used to test accurately the model performance. In this research there was used point632 error, obtained from bootstrap samples. Speaking about practical results, with the help of the point632 error there was visualized and tested the performance of classification models with different preprocessing. Firstly, it was found out that scaling is the most useful module: out of 4 considered models 3 models have a significant increase in the performance: namely, KNN, LGBM and Logit. LDA was the only model, which performance was worse with this module. On the contrary, feature selection module was the least efficient: it was ineffective in all 4 models. The latter can be explained by the fact that the dataset, considered here has only 20 features, so the efficiency of this module should be tested in other datasets as well. The transformation and handling imbalanced data modules have moderate efficiency: they both have a significant positive contribution only in KNN model. Speaking about encoding module, 3 of 4 considered models prefer One-Hot encoding (it was significantly better than other encodings) and only LDA prefers WoE encoding. From the other hand, it can be concluded that LDA shows its best performance without any preprocessing steps, while KNN demands the highest degree of preprocessing, including scaling, One-Hot encoding, handling imbalanced data and transformation, but with the help of these steps it can show relatively good performance compared to other models. Logistic and gradient boosting models are moderately dependent from preprocessing, and at the same time gradient boosting model showed significantly better performance compared to other classification models.

Speaking about potential further research, the results obtained on German dataset should be validated on other (possibly bigger and more realistic) datasets as well. Namely, there can be tested the significance of handling missing values module, which was impossible to test, because German dataset does not have any missing values. Furthermore, there can be validated the performance of feature selection module: I have a strong suspicion that in bigger datasets it can show much better results.

References

1. Arora, N., & Kaur, P. D. (2020). A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment. Applied Soft Computing, 86, 105936.
2. Ashofteh, A., & Bravo, J. M. (2021). A conservative approach for online credit scoring. Expert Systems With Applications, 176, 114835.
3. Bao, W., Lianju, N., & Yue, K. (2019). Integration of unsupervised and supervised machine learning algorithms for credit risk assessment. Expert Systems with Applications, 128, 301-315.
4. Bates, S., Hastie, T., & Tibshirani, R. (2021). Cross-validation: what does it estimate and how well does it do it?. arXiv preprint arXiv:2104.00673.
5. Carta, S., Ferreira, A., Recupero, D. R., Saia, M., & Saia, R. (2020). A combined entropy-based approach for a proactive credit scoring. Engineering Applications of Artificial Intelligence, 87, 103292.
6. Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. Neural computation, 10(7), 1895-1923.
7. Efron, B. (1983). Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. Journal of the American Statistical Association, 78(382), 316–331. https://doi.org/10.2307/2288636
8. Efron, B., & Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Statistical science, 54-75.
9. Efron, B., & Tibshirani, R. (1995). Cross-Validation and the Bootstrap : Estimating the Error Rate of a Prediction.
10. Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: the 632+ bootstrap method. Journal of the American Statistical Association, 92(438), 548-560.
11. Engelmann, J., & Lessmann, S. (2021). Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning. Expert Systems with Applications, 174, 114582.
12. Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.
13. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.

14. Junior, L. M., Nardini, F. M., Renso, C., Trani, R., & Macedo, J. A. (2020). A novel approach to define the local region of dynamic selection techniques in imbalanced credit scoring problems. Expert Systems with Applications, 152, 113351.
15. Kozodoi, N., Lessmann, S., Papakonstantinou, K., Gatsoulis, Y., & Baesens, B. (2019). A multi-objective approach for profit-driven feature selection in credit scoring. Decision support systems, 120, 106-117.
16. Lappas, P. Z., & Yannacopoulos, A. N. (2021). A machine learning approach combining expert knowledge with genetic algorithms in feature selection for credit risk assessment. Applied Soft Computing, 107, 107391.
17. Louzada, F., Ara, A., & Fernandes, G. B. (2016). Classification methods applied to credit scoring: Systematic review and overall comparison. Surveys in Operations Research and Management Science, 21(2), 117-134.
18. Tong, E. N., Mues, C., & Thomas, L. C. (2012). Mixture cure models in credit scoring: If and when borrowers default. European Journal of Operational Research, 218(1), 132-139.
19. Tripathi, D., Edla, D. R., Kuppili, V., & Bablani, A. (2020). Evolutionary extreme learning machine with novel activation function for credit scoring. Engineering Applications of Artificial Intelligence, 96, 103980.
20. Webb, A. R. (2003). Statistical pattern recognition. John Wiley & Sons.

Appendix

The realization of model pipelines, described in the text of research can be found in GitHub repository:

https://github.com/Matteus1904/Lukianov_diploma