PRAGUE UNIVERSITY OF ECONOMICS AND BUSINESS

Faculty of Finance and Accounting

Department of Banking and Insurance

MASTER THESIS

2022 Nikolai Pravdin

PRAGUE UNIVERSITY OF ECONOMICS AND BUSINESS

Faculty of Finance and Accounting

Department of Banking and Insurance

Field of study: Financial Engineering

Logistic Regression Improvements for Credit scoring development

Vylepšení Logistické regrese pro vývoj Kreditních skórů

Autor of the Master Thesis:	Nikolai Pravdin
Supervisor of the Master Thesis:	prof. RNDr. Jiří Witzany, Ph. D
Date of submission:	2022

The Declaration of Authorship

I hereby declare that I carried out the master thesis „**Logistic Regression improvement for Credit scoring development**“ independently, using only the resources and literature properly marked and included in the bibliography.

Prague,

....

Nikolai Pravdin

Acknowledgements

I`d like to thank all my teachers and mentors, who guided me during studying years. My greatest thanks are addressed to prof. RNDr. Jiří Witzany, Ph.D., whose guidance made this thesis possible.

Abstract

The current Study investigates the impact of non-linear relationships between input variables during credit risk modeling. Particularly, it is meant to analyze the burden of variables' non-linear behavior on the current modeling benchmark – Logistic regression – using some general set of retail clients' data. The main focus done on application of alternative solutions to efficiently mitigate nonlinearity's influence. Among them is well-known strong learner - Random Forest, as well as relatively new approach – Penalized Logit Tree Regression – a transparent and intuitive combination of Logit and Decision Trees. Mentioned methods are modeled on both the real-world data and specific simulated data with step-by-step description of the process. Finally, results are analytically compared using popular performance metrics like Gini Index and K-S statistic.

Key words

Credit risk, Probability of Default, Scoring models, Logistic regression, Random Forest, Penalized Logit Tree Regression, predictive power.

Abstrakt

Táto práce zkoumá dopad nelineárních vztahů mezi vstupními proměnnými při modelování kreditního rizika. Především je zaměřená na analýzu zatížení, které nelineárně chovající proměnné můžou způsobovat při používání skutečného benchmarku – Logistické regrese – v případě že pracujeme s obecnými údaji fyzických klientů. Hlavním záměrem je zkusit alternativní řešení, které by efektivně zmírnily negativní dopad případně nelinearity. Mezi nimi je dobře známý Náhodní Les, a taky poměrně nový přistup – Penalizována Logit Tree regrese – transparentní a intuitivní kombinace Logitu a Rozhodovacích stromů. Zvolené metody jsou aplikovány a jak na reálná data, tak i na simulovaný soubor, včetně detailního popisu celého procesu. Výsledkem je analytické porovnání výsledků při použití oblíbených ukazatelů výkonnosti jako Giního index a K-S statistika.

Klíčová slova

Kreditní riziko, Pravděpodobnost Defaultu, Skóringové modely, Logistická regrese, Náhodní les, Penalizována Logit Tree regrese, síla předpovědi.

Table of Contents

1 Introduction

How to valuate a client, how to decide about giving a loan and on what conditions, what loss may unreliable borrower cause – these are the questions that moneylenders need to answer on daily basis to properly manage their credit risks. In particular, we are going to talk about credit default risk – a risk that a borrower will be unable to repay its debt obligation, which naturally causes certain losses and directly influences lender's financial state. Now imagine all these banks, credit unions, financing companies etc. with the great variety of products they offer - what a tremendous part of world economy they present. No wonder that not only lenders themselves but also the regulator is highly interested in their business to run smoothly.

But how can we know if borrower is going to default and how safe are we lending him money? The decision may be based on expert opinion, but some numerical expression is still required **-Probability of Default (PD) –** is a likelihood of a **default event** to occur in a particular time horizon.

But what counts as a default event? One **Definition of Default (DoD)** is provided by **European Banking Authority (EBA**) for purposes of harmonization and improvement of consistency for European banks' application of regulatory requirements. According to EBA definition "default shall be considered to have occurred with regard to a particular obligator when … the obligator is past due more than 90 days on any material credit obligation to the institution the parent undertaking or any of its subsidiaries[1]."[2] Since definition may vary and change, we will simplify and consider PD to be a probability of obligor entering default status by any valid and currently applicated DoD.

Probability of Default may be used to directly valuate client`s credibility (for example, in scorecards) or to assign client into pre-defined rating groups and based on that to decide about loan conditions. PDs are also used to calculate **Expected Loss** and as a parameter for **Capital requirements**. So, the importance of PDs in the Credit risk management can`t be underestimated.

How PD can be estimated though? Statistical and mathematical methods provide us a whole palette of possible approaches to model PDs. Probably the most popular technique nowadays is estimation with **Logistical regression (LR)** or simply **Logit**.

Logit offers a lot of benefits, but also has certain shortcomings. The **assumption of linearity** is the one we are going to primary address in this Study. Truly, strong non-linear relationships between variables may significantly worsen model's power. There are alternatives like **Random Forests (RF)** that are more fit to solve such tricky behavior, but its complexity makes such sophisticated methods to be viewed as a black

[1] That's only the first part of the definition that does not mention materiality criteria, unlikeness to pay, technical default and so on. However, the rest is irrelevant for our Study.

[2] EUROPEAN BANKING AUTHORITY. Capital Requirement Regulation (CRR), Article 178. *eba.europa.eu* [online].

box, thus turning modelers away. However, attempts to overcome these drawbacks and beat the benchmark are still performed.

One such solution is suggested by the collective of authors in the **Penalized Logit Tree Regression** study (**PLTR study**)[3] that served as an inspiration for our current Study. Our main goal is to challenge benchmark Logit's performance using new PLTR technique by simulating real modeler's activity and assess its applicability in practice.

Theoretical part is meant to describe three chosen models of ours: Logit, Random Forest and PLTR, so that reader may obtain a clear idea of models' functionality, strong sides, and weaknesses. It also describes modeling process in general taking into account data preparation and back-testing. Finally, our reader may find there an information about commonly used metrics to express model's prediction power for purposes of later validation and comparison.

Analytical part consists of applications of all 3 models based on the real-world data. It describes in detail the whole procedure of models development[4] as transparent as possible, so that reader can reconstruct any done calculations or make its own using these chapters as a guideline. Each application section is followed by a brief comparison of models' performance. Computational part is completed by an example based on simulation data, meant to better highlight differences of studied models.

Quite a few studies have been already written based on the comparison of different modeling approaches. Our reader may find interesting these following studies:

"Classification Models for Software Defect Prediction." of Lessmann and Baesens[5], where authors compare 22 modeling algorithms, including Logit and Random Forest, and its following extended update.[6]

"Improving the Art, Craft and Science of Economic Credit Rik Scorecards Using Random Forests." of Sharma[7], focused on analysis the superiority of Random Forest over Logistic regression.

[3] DUMITRESCU, E., HUÉ, S., HURLIN, C., TOKPAVI, S. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*. 2021.

[4] All calculations are performed in specialized statistical software – R-studio. Some code parts for different computational tasks were borrowed from www.rdocumentation.org, web-forums stackoverflow.com and stats.stackexchange.com, and related guidelines MONDAL, Ariful. Classifications in R: Response Modeling/Credit Scoring/Credit Rating using Machine Learning Techniques. *rstudio-pubs-static.s3.amazonaws.com*

[5] LESSMANN, S., BAESENS, B., MUES, C., PIETSCH, S., Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings. 2008.

[6] LESSMANN, S., BAESENS, B., SEOW, H., Benchmarking state-of-art classification algorithms for credit scoring: And update of research. *European Journal of Operational Research.* 2015.

[7] SHARMA, D., Improving the Art, Craft and Science of Economic Credit Rik Scorecards Using Random Forests: Why Credit Scorers and Economists Should Use Random Forests. *SSRN*. 2011.

“A Comparison of Classification/Regression Trees and Logistic Regression in Failure Models”[8], comparing Classification Trees and Logistic Regression based on credit data of micro-entities from the United Kingdom.

“Credit Risk Analysis Using Machine and Deep Learning Models”,[9] where authors additionally compared performance of more sophisticated machine and deep learning models, such as gradient boosting and neural network.

“A Comparative Assessment of Credit Risk Model Based on Machine Learning”,[10] studying precision and accuracy of five popular classifiers: Naïve Bayesian model, Logistic regression, Decision Trees, Random Forest, and K-nearest Neighbor classifier.

“Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets”[11], where changes in models’ performance were for different number of observations, explanatory, and noise variables, and for different variance in these variables.

Additionally, of particular interest might be “Non-Linearity Issues in Probability of Default Modelling” by Klinkers[12], studying impact of non-linearity on PD models, and “Weight of evidence transformation in credit scoring models: How does it affect the discriminatory power?*”* by Persson[13], focused on the impact of WoE-transformation of explanatory variables, which were proposed as a possible solution of non-linear relationships in this study.

However, we found no articles studying PLTR approach besides the original one mentioned earlier, which became another reason for our current study to exist.

The Study ends with the overall summary of our findings and made conclusions, while also contains corresponding recommendation regarding models’ applicability as well as few suggestions for further studies.

[8] IRIMIA-DIEGUEZ, A.I., BLANCO-OLIVER, A., VAZQUEZ-CUETO, M.J., A Comparison of Classification/Regression Trees and Logistic Regression in Failure Models. *Procedia Economics and Finance, Volume 26*. 2015.

[9] ADDO, P.M., GUEGAN, D., HASSANI, B., Credit Risk Analysis Using Machine and Deep Learning Models. *Risks.* 2018.

[10] WANG, Y., ZHANG, Y., LU, Y., YU, X., A Comparative Assessment of Credit Risk Model Based on Machine Learning. *Procedia Computer Science, vol. 174*. 2020

[11] SMITH, T., KIRASICH, K., SADLER, B., Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets. *SMU Data Science Review.* 2020

[12] KLINKERS, L., Non-Linearity Issues in Probability of Default Modelling. *University of Twente.* 2017.

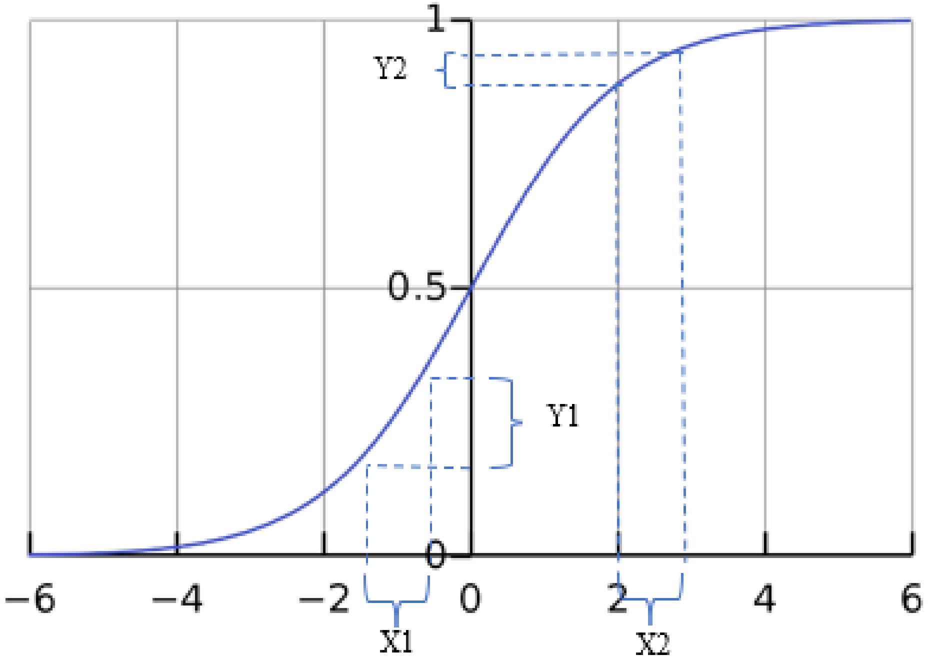
[13] PERSSON, R., Weight of evidence transformation in credit scoring models: How does it affect the discriminatory power? *LUP Student Papers.* 2021.

2 Modeling techniques

2.1 Classic Logistic regression

Among the most popular approaches to model Probability of Default, logistic regression might have the widest usage in practice.[14] Primarily it`s due to the relatively simple construction procedure and due to logistic curve been S-shaped.[15] The desired trait is well demonstrated by the following chart of Logistic curve:

Figure 1 - Logistic curve



Source: Own modification of Wikipedia chart[16]

With the function curve of such shape PD won`t exceed 0 and 1, moreover, the curve has increasing shape on the whole range, just as we would expect our model to act (with predictor's value increasing/decreasing PD would also increase/decrease).

The Default Probability function can be written as:

$$P(Y = 1|X_1, \dots, X_k) = \frac{e^{(b_0 + b_1 x_1 + \dots + b_k x_k)}}{1 + e^{(b_0 + b_1 x_1 + \dots + b_k x_k)}} = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + \dots + b_k x_k)}}, \qquad (1)$$

Meanwhile, Logit function (also called log-odds, since it represents logarithm of odds of variable x) can be expressed as:

[14] Based on GREENE, W.H. Econometric Analysis. 5th Edition. *Prentice Hall, Upper Saddle River.* 2003.

[15] By common sense we would expect Default Probability to obtain values on the range from 0 to 1 (or from 0% to 100%). Hence, we prefer to use the function that limits possible results to 0 at one end and 1 at another end such as logistic function.

[16] Source: Wikipedia. Logistic regression. *en.wikipedia.org*.

$$logit(x) = \log\left(\frac{x}{1-x}\right), \qquad (2)$$

Combination of both functions leads to the desired Logit model:

$$logit\big(P(Y = 1|X_1, \dots, X_k)\big) = b_0 + b_1x_1 + \dots + b_kx_k, \qquad (3)$$

where x_i is the *i-th* explanatory variable used in the developing model and b_i is the *i-th* variable's coefficient, that is unknown to us and needs to be estimated. One of possible solutions to compute unknown parameters is application of the Maximum likelihood estimation. The mentioned approach allows us to obtain unknown b_i parameters by maximizing the following log-likelihood function:

$$L(b) = \sum_{i=1}^{n}[y_i * \log(\frac{1}{1 + e^{-(b_0 + b_1x_1 + \dots + b_kx_k)}}) + (1 - y_i) * \log(1 - \frac{1}{1 + e^{-(b_0 + b_1x_1 + \dots + b_kx_k)}})], \qquad (4)$$

with n being the number of observations in the modelling data sample.

What important for the case of this study is model`s interpretability. It can be seen with function (3) that Logit model is indeed linear in parameters. In other words, target variable on the left side is linearly dependent on each explanatory variable X on the right side. Hence interpretation of the impact of each individual predictor can be obtained quite easily - a quality highly appreciated in practice by model's users and the regulator. Thus, for any positive coefficient b_i the increase of corresponding predictor value x_i will lead to higher values of PD and vice versa. Opposite is true for a negative coefficient b_i as the increase of corresponding predictor`s value x_i will lead to lower values of PD and vice versa. Same logic is well applicable for explanatory variables being ordinal or binary. In the case of non-numerical predictor (e.g., **Marriage status** or **Education degree**) the same result can be achieved by a simple transformation to the numerical form, either by using dummy variables or a more advanced technic like WoE-ization[17] - replacing non-numerical values with its Weight of Evidence (WoE) values:

$$WOE = \log\left(\frac{\%\ of\ non - events}{\%\ of\ events}\right), \qquad (5)$$

However, the absolute impact of predictor value's change on PD is not constant for Logit model and does not equal to the coefficient's value like in general linear model. Instead, it depends on the initial PD value for which we compute the impact. Result will be weaker for extreme values of PD and stronger for PDs being close to its mean. The dependency can be better visualized on the *Figure 1* by comparing impacts that shifts at the x-axis (x_1 and x_2) have on the y-axis values (y_1 and y_2) in different segments of the chart.[18] Yet the logic of positive/negative impact remains the same and its absolute value for each situation can be calculated individually.

[17] Based on ENGELMANN, B., RAUHMEIER, R. The Basel II Risk Parameters. *New York: Springer*. 2006

[18] While X1 and X2 shifts are identical, resulting Y1 and Y2 shifts are noticeably different.

Alternatively, if needed, average impact of x_i variable can be described even better by **marginal effects**.[19] We will consider two main approaches that exist for such solution. Either we may use an Average Marginal Effect (AME), that measures the mean of marginal effects for all observations for each individual variable x_i:

$$AME_j = \frac{1}{n}\sum_{i=1}^{n} f\left(x_i\hat{\beta}\right)\hat{\beta}_j, \tag{6}$$

or a Marginal Effect on the Mean (MEM), that measures marginal effect for the mean of each individual variable x_i:

$$MEM_j = f\left(\bar{x}\hat{\beta}\right)\hat{\beta}_j, \tag{7}$$

For

$$\bar{x} = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} x_i, \tag{8}$$

These two approaches have slightly different interpretation of the results, as well as its own advantages and disadvantages, however none of them should not cause any difficulties to calculate. Hence, Logit model meets our requirement for simplicity of interpretation.

As was mentioned before, the reason of an easy interpretation is model’s linearity in parameters, which presumes a strict linear relationship between dependent and independent variables. Such simplicity can`t really be achieved without a cost. In a situation when relationships are happens to be non-linear, the model may fail to catch these dependencies efficiently and thus suffer losses in predictive power, returning poor or even misleading results.

Of course, in real world it is hard to find a perfect linear dependency between two variables (unless they are just a linear transformation of each other), so it is often sufficient if relationship is mostly linear. However, even presumption of linearity is often not hold.[20]

Let`s describe in few words some common sources of nonlinearity with possible solutions that are applicable on the variables’ level and don`t require global modifications of the whole Logit model.

[19] Based on WILLIAMS, R. Using the margins command to estimate and interpret adjusted predictions and marginal effects. *Stata Journal* 12: 308-331. 2012.

[20] Mainly, nonlinearity will be a problem for numerical continuous or ordinal variables. It is a common practice to check such variables during univariate analysis and take appropriate actions to fix nonlinearity so logistic regression remains efficient.

1) Obvious nonlinear relationship caused by the nature of the variable (can be easily seen on the chart of the dependent variable vs predictor, e.g., exponential growth of the curve).

   Possible solutions: Variable's transformation with exponential or quadratic term, for example, Quantile binning for continuous variables or group binning for ordinal variables; WoE-ization.

2) Threshold effect caused by the nature of the variables (e.g., for "Income" variable having monthly income beyond certain level may lead to the rapid non-linear drop of PDs)

   Possible solutions: Quantile binning for continuous or group binning for ordinal variables; WoE-ization.

3) Violation of independency between two or more predictors (as a result, univariate analysis' chart may show a false linear or non-linear dependency, since univariate analysis does not take into account predictor`s high correlation with another potential predictor, that in multifactor model will influence relationship between former predictor and PD).

   Possible solutions: Exclusion of one of correlated predictors; Incorporation of interactions into the model; Substitution of correlated variables with a single index that includes both original variables (for example, using well-known or specific financial indexes).

On top of that, mistakes done in the model specification may interact with each other causing further nonlinearity in the model, which naturally leads to even worse performance.

It is debatable, of course, how well mentioned procedures may reduce the negative impact of non-linear relationships. Since the study is practically orientated, simulation of real-life working environment is aimed, hence, the main Logit model will include all common practices to deal with non-linear relations, as we would expect to be performed by the experienced modeler. For deeper understanding of how significant impact might mentioned problematic have on our Logit model, the second model based on same predictors will be constructed with explanatory variables entering second model in their original form (more details in the Logit application section).

For now, let`s stop with Logit and have a look at other modelling solutions.

2.2 Random forest

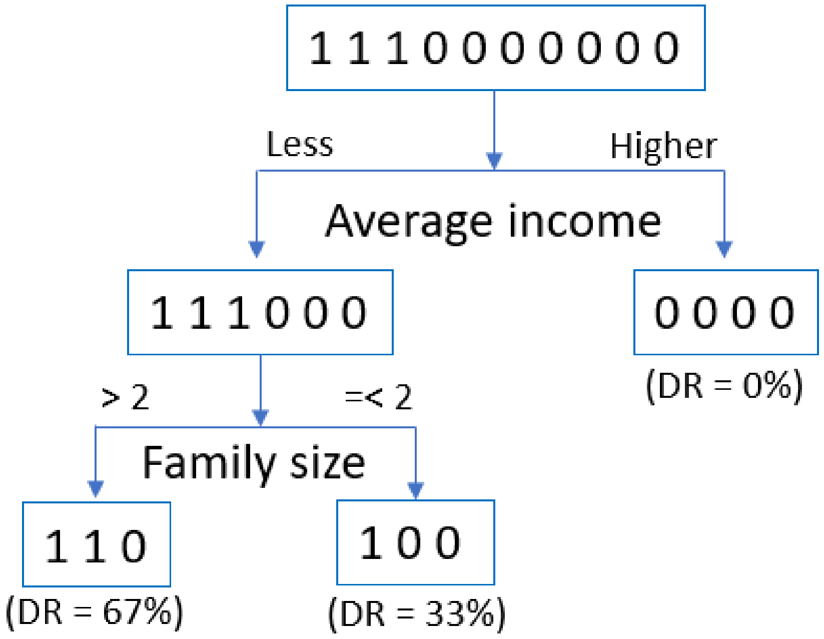
As already been discussed in the previous chapter, when predictor shows nonlinear relationship towards the explanatory variable, such complication can be somewhat mitigated using variables transformation. But let`s consider another model building approach that will solve nonlinearity issue on a structural level. A good candidate for this purpose is a method known as Decision Tree. To demonstrate the way Decision

Trees work and what advantages it provides us we will use following simple and intuitive example:

Let`s consider 10 observations with Default Status being a target variable such that "1" indicates a defaulted client, resp. "0" indicates a non-defaulted client. **Monthly income** and **Size of the family** will be used as explanatory variables during our example. There is a theoretical reason to expect non-linear relationship for these two variables: PD quickly dropping for higher incomes and households, higher number of family members having higher financial burden, which leads to higher PDs.[21]

One of possible Decision Trees to construct based on that example is presented by Figure 2:

Figure 2 - Two-variables Decision Tree example



Source: Own construction

In the first step decide if client's income above or below chosen threshold (some theoretical average value was taken for representation) to classify our observations, after that sample is divided according to that decision. The "higher" branch shows perfect identification, while the "less" branch still contains both defaulted and non-defaulted observations, thus can be split further. In the second step Family size variable is used, leading to additional separation into two groups. As a result, we end with three possible categories meant to classify an input client and return default expectation.

It can be seen that result in our example is not ideal, and though we prefer a perfect fit, it would be almost impossible to achieve without falling into the pit of overfitting.[22]

[21] It can be argued that, on the contrary, larger families are more financially stable while showing more responsible behavior that should actually cause lesser PDs, but for our example it is irrelevant

[22] By overfitting we mean situation, when model reach high results on train sample, but suffers significant performance reduction on new test sample.

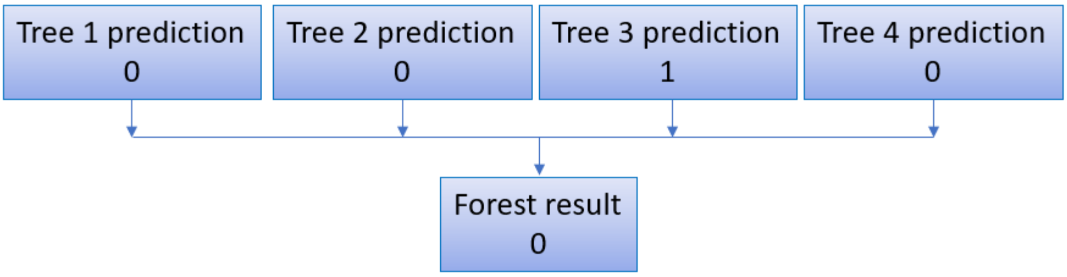
Hence in practice we seek only to split sample such that resulting groups are as different as possible and observations inside each group are as similar as possible. Naturally, that requires an appropriate splitting criterion, we are going to use for this purpose **Diversity**, measured with **Gini Index**. The best split will be the one that reduces diversity the most. More about the splitting procedure and other parameters of Decision Trees will be described later in its application part.

What is important for purposes of the study is how Decision Tree handles input variables. In our example Income is divided into two categories so each of them now has its own relationship with PD, thus if the variable suffers from non-linear relationship due to the threshold effect, then Decision Tree can effectively handle it by splitting variable respectfully. More branches can be used to catch multiple threshold effects inside single variable or among few independent variables, same can be applied to categorical explanatory variables.

Another advantage of using multiple nodes for the Tree allows is to cover possible interactions between two predictors. Back to our example: we use Family size in the second note of the Tree that interact with the first predictor – Income. Thus, Decision Tree categorize our clients by number of family members with respect to their Income, so Family size has a different meaning for different levels of income, which is logical and can be expected to present in real life.[23] In addition, the same variable can be used multiple times to handle more complex non-linear relations.

More than that, we aren't going to use only a single Tree to model PDs, but many of them – the extension of Decision Tree classifier known as Random Forest. The goal is to obtain a better prediction by using the mean result of some number of Trees rather than rely on a single one. Practically it means that for any new observation a number of predictions will be made equal to the number of Trees in the Forest, then the most frequent result will become model's final prediction. This way we reduce the risk of classification error, since now few models will provide us prediction, clearing away individual model's errors. The logic is simple – where one is likely to mistake, many will mistake unlikely.[24]

Figure 3 - Four-trees example of Random Forest



Source: Own construction

[23] In our case Family size is not significant for higher incomes, it was only used as separation feature for lower values, since right branch does not need further splitting

[24] Based on BREIMAN, Leo. Random Forests. *Machine Learning* ***45,*** *5–32*. 2001.

Of course, there is one important assumption to make, that individual Trees in the Forest need to remain uncorrelated or rather as little correlated as possible. To ensure an adequate level of diversity between models few actions might be taken.

First one is to use the Bagging technique – allowing each individual Tree to sample from training data randomly with replacement, thus each Tree will be constructed on a slightly different data sample but close to the original one, which may resolve some uncertain decisions, where the uncertainty was mainly caused by specific structure input data.[25]

Another technique to support lesser correlation is to use different explanatory variables for each individual node. A feature for each split will be chosen from reduced set of predictors randomly selected out of all short-listed candidates. This way every Tree is not only grown on different data due to Bagging, but also have a slightly different structure of splits. Using both techniques together in combination with an appropriate data management allows Random Forest to build different enough Trees for receiving uncorrelated predictions.

All these techniques combined are a strong tool to deal with non-linear relations. Authors of PLTR[26] showed on their own examples how Random Forest may significantly outperforms Logit in terms of prediction accuracy. As a secondary objective, this statement will be tested in current study as well - Random Forest will be built on the same data as Logit model to check the actual difference between models' power.

An expected question would be – if Random Forest proves to perform better, why not to use it instead of Logit then? Well, as was said before in practice it is often preferred to have transparent model, that is easy to understand and easy to interpret. However, if we look deeper into Decision Tree's construction process, what we can see is a sequence of mathematical or handmade split-decisions that categorize clients into homogeneous groups with as many interactions as the maximum number of nodes allows us. That makes massive Trees hard to interpret. On top of that, for Random Forest we construct many such Trees, each have its own structure and splitting values, and the final prediction is based on the average output of all Trees. In such situation it is almost unreal to form a unique numerical interpretation for the whole model, unlike Marginal effects for Logit models. At most we are able to calculate importance of individual features[27] and get intuitive understanding of the structure of not-too-extensive RFs.

Though the Random Forest method has some attractive advantages, the lack of interpretability makes it an unpopular tool for risk modelers and regulator.

[25] For example, if inside one node proportions of default and non-defaults are close, DT will always prefer category that slightly overweight, while bagging may cause proportions to change, resulting in opposite conclusion for another Tree, that will somewhat reflect mentioned uncertainty in the Forest.

[26] DUMITRESCU, E., HUÉ, S., HURLIN, C., TOKPAVI, S. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*. 2021.

[27] For example, by using Gini-based mean decrease impurity.

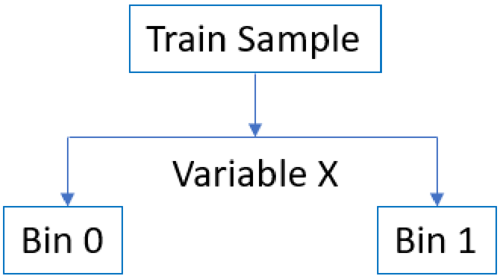
Nevertheless, Random Forest will fit nicely into our comparison for purposes of the study.

2.3 Penalized Logit Tree regression

Penalized Logit Tree Regression (PLTR) is a combination of two earlier described methods. Authors presented the model as a balanced alternative, meant to incorporate best of both worlds: simplicity and interpretability of the logistic regression, as well as flexibility and adaptability of the Random Forest classifier. The idea described in the PLTR study was "to build a logistic regression model based on univariate and bivariate threshold effects. The latter are obtained using decision trees that rely on each predictive variable (singleton) and each couple of predictive variables at a time..."[28].

Decision tree for each individual predictive variable X is constructed the following way:

Figure 4 - Example of singleton-based Decision Tree



Source: Own construction

As the Figure 4 shows, each variable X is to be divided by the Tree classifier into two groups based on the Gini Index as a split criterium. New groups are encoded as a binary (0,1) variable with "0" being first category, and "1" being second category. When such variable enters Logistic regression, it represents univariate thresholds effect of the corresponding original predictor. Procedure is repeated for each short-listed explanatory variable.[29]

Application of Random Forest later in the Study will show, that not every variable can be effectively handled by the Decision Tree classifier. Mainly for categorical variables with a low number of categories (for example, Car Ownership Status) it fails to provide meaningful result.[30]

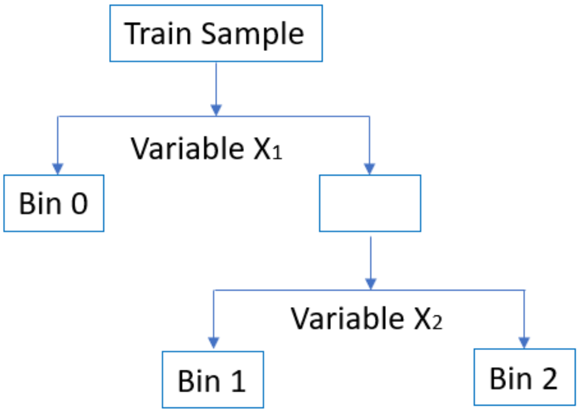
[28] DUMITRESCU, E., HUÉ, S., HURLIN, C., TOKPAVI, S. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*. 2021.

[29] Indeed, for PLTR approach we also do apply pre-selection to reduce the number of predictors we need to work with.

[30] In the PLTR study author do not describe their solution for the mentioned issue. They might have avoided it in other datasets, and thus used only Decision Trees' classification as predictive variables. Since the problem is caused by imbalance target variable, under-sampling that we later implement will solve it.

After univariate effects are solved, we would like to continue and incorporate also bivariate threshold effects. We construct additional set of Decision trees for each **couple** of two unique explanatory variables. The resulting Tree can be generalized in the following form:[31]

Figure 5 - Couple-based Decision Tree with 3 bins output

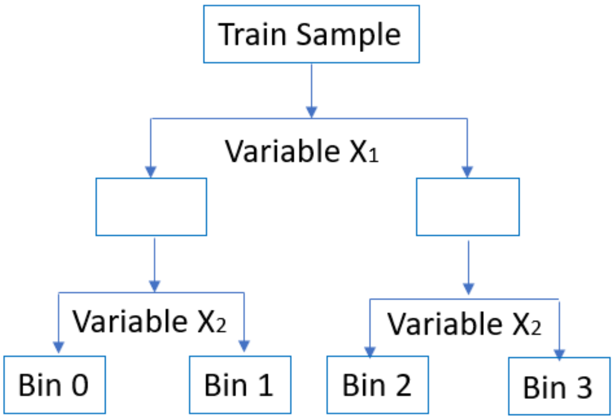


Source: Own construction

New variable is produced with not two, but three categories. With respect to the Figure 5 each file may end in group 0, 1 or 2, depending on values of corresponding variables X_1 and X_2. Newly generated predictors take into account interactions of each couple of explanatory variables. These predictors are viewed as categorical variables and enters the logistic regression as a set of dummy variables with first “zero” group being a referenced group.

Of course, there is possibility that both branches of the first variable can be split further as shown with the Figure 6 below. Logically, in such case four groups are derived and thus three dummies are created.

Figure 6 - Couple-based Decision Tree with 4 bins output



Source: Own construction

[31] Placing of variables depends on their importance.

Like in the case of univariate threshold predictors, two-level-depth Decision Trees may also suffer from low number of categories in applicated variables. If such happens, we will get a Tree of a single split or of no spits at all.[32] To prevent such trees entering the regression all couple-based trees are checked, and only those fully grown are kept.

By using constructed one- and two-level-depth Trees' categorization results as input variables, we obtain new Logistic regression model in the familiar form:

$$P(Y = 1|X_1, \dots, X_k) = \frac{e^{(b_0 + b_1 x_1 + \dots + b_k x_k)}}{1 + e^{(b_0 + b_1 x_1 + \dots + b_k x_k)}} = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + \dots + b_k x_k)}}, \qquad (9)$$

Where x_i is a dummy variable for a certain category of each grown Tree, and b_i is a to-be-estimated coefficient of that category.

Earlier we talked a lot about easy interpretability of the estimation results of the Logit model. One can see that since the structure is the same, the overall meaning of the coefficients also remains the same. It is also true for marginal effects if we would like to calculate some. The only thing that changes is a meaning of explanatory variables. Just like for any other categorical predictor we use zero-one dummy variables, so if an observation belongs to the particular category, its i-th dummy will be equal to 1, and thus Logit changes by corresponding b_i. The difference is that each category now represents a certain node of Decision Trees. Since all Trees are small, they can be easily printed and studied for validation purposes.

However, interpretation becomes slightly more complicated due to the incorporation of predictors' interactions. Each single variable will now be represented not by one variable (or rather a set of few dummies), but also by every couple formed from this variable with any other variable. Hence, the total impact on Logit will be:

$$\Delta\, logit = b_i + \sum b_{ij}\,, \qquad i \neq j, \qquad (10)$$

Where b_i is a coefficient of the *i-th* singleton, b_{ij} is a coefficient of couple formed by variable i with another variable j.[33] Total impact remains easy to calculate and clear to understand.

The PLTR model is not limited by univariate and bivariate threshold effects. It can be extended further to include interactions of three and more variables using the same logic we described above, and thus fit more complex non-linear relations. In the PLTR study authors do not go beyond couple-based Decision Trees, so the model remains less intricate and finely interpretable, which is in dictated by our main goal.

Of course, using same variables multiple times will lead to the enormous number of predictors in the model, where every original explanatory variable will be included

[32] First option will prove to be identical to the singleton we have already built for this variable and the second option is of no use to us.
[33] The number of coefficients we need to sum depends on the number of couples we have successfully implemented into Decision Trees.

into few predictors. Both things combined leads to the risk of high multicollinearity. Penalized part of PLTR is responsible for mitigation of the multicollinearity effect.

The idea is to add "a penalty term to the negative value of the log-likelihood function … that penalizes the estimates during the estimation process"[34]. To include penalty term we can applicate, for example, the **Least Absolute Shrinkage and Selection Operator**, also known as **Lasso**. To overcome some limitation of the technique, adaptive Lasso regression is preferred over its classic form.

Let`s consider $L(X_i;\beta_i)$ to be our log-likelihood function during Logit model's parameter estimation with X_i being dummy variables and β_i being its coefficients. Now we add a penalty term to log-likelihood's negative value, and thus receive:

$$L(X_i;\beta_i) = -L(X_i;\beta_i) + \lambda \sum \widehat{\omega}_i |\beta_i|, \tag{11}$$

where λ is the tuning parameter that is chosen based on the results of 10-fold cross-validation[35], β_i are coefficients to estimate, and $\widehat{\omega}_i$ are Adaptive Weights meant to regulate the level of penalization for each coefficient. Adaptive Weights can be calculated the following way:

$$\widehat{\omega}_i = \frac{1}{\left(|\hat{\beta}_i^{ini}|\right)^{\gamma}}, \tag{12}$$

where $\hat{\beta}_i^{ini}$ is an initial estimation of the coefficients that can be obtained, for example, with Ridge regression beforehand, and γ is an adjustment term for the Weights vector.

Now that the model itself is explained, let`s describe our expectations.

If we compare PLTR to the benchmark Logit, we can notice that Logit Tree Regression also transforms explanatory variables into categorical form, yet divides them in two groups only, unlike classic Logit where the number of bins is based on the results of the binning procedure. Naturally, due to higher number of categories classic Logit will have more options to fit data better.

On the other hand, PLTR also includes all meaningful pair interactions, which provide us new unique ways to fit data. If the true data generation process is a subject of strong non-linear cross-variable relationships, the PLTR method should significantly improve model's performance and make it closer to the Random Forest's results.

[34] DUMITRESCU, E., HUÉ, S., HURLIN, C., TOKPAVI, S. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*. 2021.

[35] By 10-fold cross-validation we understand estimating model 10 times with train sample being divided into 10 folds, so for each estimation 9 folds are used to train data and the remaining one is kept out for back-test's purposes. After that some mean values are taken as a result. This way we guarantee robustness, while using whole sample as training data.

3 Performance quality metrics

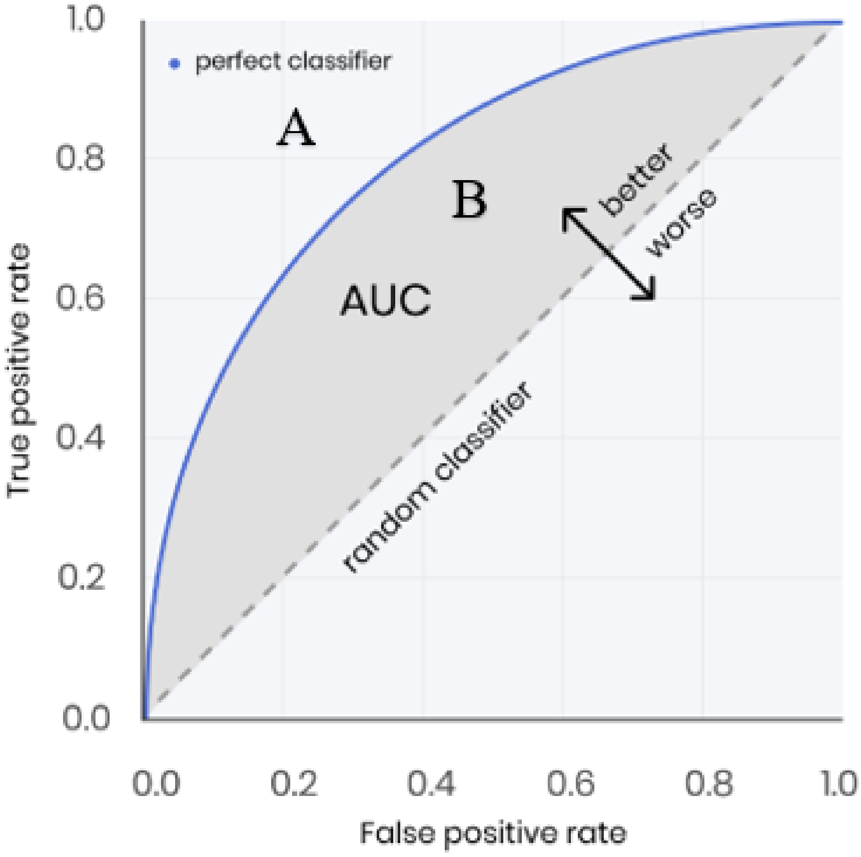
We have compared models theoretically, yet what really matters for usage, is how these models perform in practice. To valuate models' quality, we will look at the accuracy of its predictions using different well-known metrics. Our final decision will be primarily based on their results. Theoretically we expect Random Forest to outperform classic Logit model, and modified Logit models to behave somewhere in between. If obtained results happen to be in line with our expectations, then the next question to answer: Is a gain in the accuracy of predictions worth all troubles of dealing with the more complex models? We`ll see it in the comparison part of the study.

Now let's describe what metrics are we going to use to validate model's performance.

3.1 Area under the Receiver Operating Characteristics

The first metric to measure performance quality of the model is the Area Under the Receiver Operating Characteristics (AUROC)[36]. The Figure 7 right below visualizes what AUROC represents.

Figure 7 – AUROC



Source: evispot.ai[37]

[36] Based on WITZANY, J. Credit risk management: pricing, measurement, and modeling. *Cham: Springer*, 2017.

[37] Evispot. Area under the ROC Curve (AUC). *evispot.ai*.

On the Y-axis we have **True positive rate** (TPR) calculated as $\frac{True\ Positives}{True\ Positives+False\ Negatives}$ and on the X-axis is **False positive rate** (FPR) calculated as $\frac{False\ Positives}{True\ Negative+False\ Positives}$. TPR tells us the proportion of defaulted clients that were successfully identified with the model to all actually defaulted clients in the sample, and TPR tells us the proportion of non-defaulted clients that were incorrectly identified as defaulted to all actually non-defaulted clients in the sample. Naturally, we would like to construct a model with TPR been possible maximum and FPR been possible minimum - best solution that is marked as "perfect classifier" at top-left corner of the Figure 7.

Blue line represents **ROC** – a probability curve constructed on the predictions from our model with respect to the observed defaults. Our position on the ROC is determined by the PD threshold we chose to separate defaulted and non-defaulted clients. Meanwhile, the diagonal "random classifier" line is a pure **random model**, as if we decide about future defaults by tossing a coin.

Now AUROC is an area under the ROC. It shows how capable our model to distinguishing between defaults and non-defaults, taking values between 0 and 1, with 1 been a perfect separation. Pure random classifier will have AUC equal to 0.5.

Naturally, the higher AUROC is the better. However, if AUC is below 0.5 it may be appropriate to consider opposite score values and thus simply reflect ROC to the other side of the random diagonal line.

3.2 Gini Index

Gini Index, or Gini Coefficient, is another popular way to measure prediction quality that we are going to use in this study. Basically, it shows how much our model overperforms some random model and is closely connected to the AUROC.

If we look at the earlier AUROC Figure 7 we can express Gini following way:

$$Gini = \frac{B}{(A+B)} \quad \text{or} \quad Gini = \ 2 * B, \tag{13}$$

Alternatively, Gini can be derived from the AUROC directly:

$$Gini = \ 2 * (AUROC - 0.5), \tag{14}$$

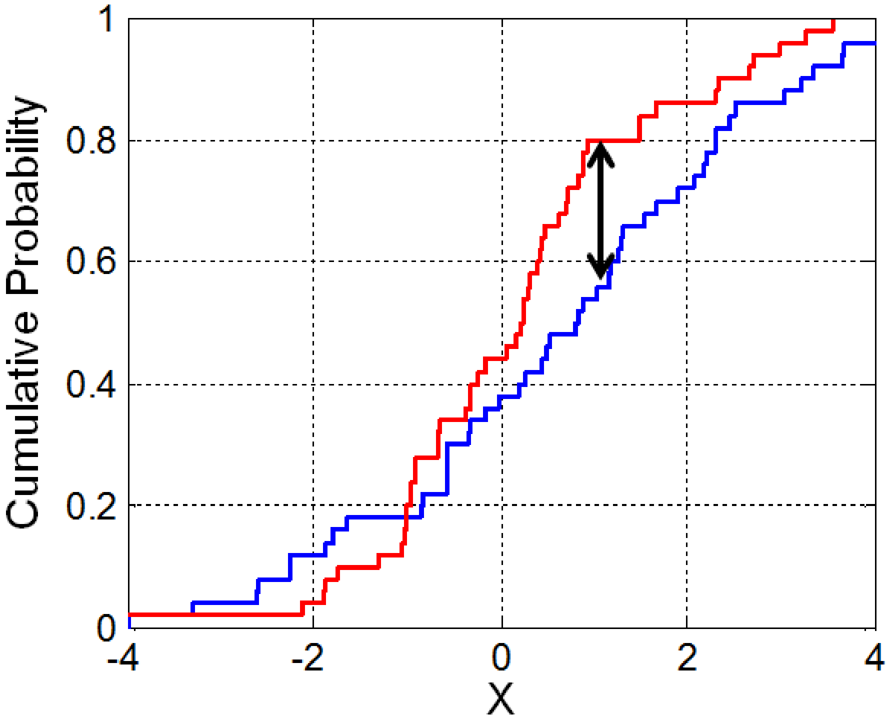
Both approaches will provide us the same result, since AUROC is basically equal to $(B + 0.5)$.

The Gini Coefficient can take values between -1 and 1, with 1 been a perfect result. Logically random model will have Gini equal to 0. Just like in the AUROC example, we prefer higher values for Gini, negative or positive all the same, since in the case of negative Gini we can simply use opposite scores.

3.3 Kolmogorov-Smirnov test (KS test)

KS test is a nonparametric test of continuous, one-dimensional probability distributions that can be used to compare a sample distribution with a reference probability distribution, or to compare two samples' distributions as shown on the chart below.

Figure 8 - Illustration of the K-S statistic



Source: Wikipedia[38]

We are interested in two samples comparison presented on the Figure 8. In case of scoring model, we can use defaulters' and non-defaulters' sub-samples. Doing so we are able to measure the quality of classification by computing the differences between two cumulative distributions, where the first one is the cumulative distribution of defaulters $F_{bad}(C)$ with respect to the cut-off score C and the second one is the cumulative distribution of non-defaulter $F_{good}(C)$ with respect to the same cut-off C.[39] The desired statistic is then found as a maximum of absolute differences for each possible cut-off. Mathematically KS statistic can be expressed the following way:

$$KS = \max\left|F_{bad}(C) - F_{good}(C)\right|, \tag{15}$$

For this metric we would also expect higher results for the better model, meaning that two classes are separated as much as possible. Hence, in the perfect model we would

[38] Source: Wikipedia. Kolmogorov-Smirnov test. *en.wikipedia.org*.

[39] Based on WITZANY, J. Credit risk management: pricing, measurement, and modeling. *Cham: Springer*, 2017.

witness KS test statistic been equal to 1, so defaulters are completely segregated from non-defaulters.

3.4 Brier Score

The Brier Score is used to measure accuracy of probabilistic predictions. It measures the mean squared difference between the predicted probability (our model's PD prediction) and the actual outcome (observed defaults). Brier Score formula can be written as:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(f_i - o_i)^2, \qquad (16)$$

where f_i is an i-th client's predicted default probability, o_i is the corresponding default status and N is the total number of observations.

Naturally the possible range of the result values is <0; 1>. From the BS formula follows that for the more accurate predictions the BS will move closer to 0, and for the less accurate it will be closer to 1. Considering this interpretation, we expect obtain lower BS values for the better model.

4 Application on real data

4.1 Data description

Quality of data is extremely important subject during development of any Probability Default model. Mistakes in a data preparation process may jeopardize the whole modeling process making whatever obtained results meaningless. That's why data analysis is a fundamental part of PD model development in practice, neither should it be overlooked in this study. For that reason, we will dedicate the whole chapter to the preparation of an adequate data sample, so us and readers can rest assured that estimations won't be compromised by a poor data management.

Since results of the study are meant for potential practical usage, another objective of data preparation will be to maintain working sample as close to the reality as possible. For this purpose, real data were searched for, which also contain a sufficient number of observations and a vast variety of potential predictors. The final choice was a popular dataset "Home Credit Default Risk", found on the **kaggle.com** website.

The sample was released in year 2018 under the terms of Home Credit Group competition, and thus can be viewed as a sufficiently recent dataset.[40] It includes 307511 observations in the Train sample – a sample used for model construction and estimation, and 48744 observations in the Test sample – a sample used as "new data" for back-testing purposes. The total number of observations in both sample and its ratio 86:14 are believed to be adequate for approximately accurate results.

During further data analysis it was discovered that for unknown reasons Test sample does not include information about **default status** – our target variable. Regretfully it thus can`t be used to calculate the quality of predictions. To keep advantages of out-of-sample back-testing it was decided to randomly split Train sample into two newly generated samples with ratio 80:20[41] (246008 observations for the new Train sample and 61503 observations for the new Test sample).

The data were provided in raw form within few samples presenting different sources (*Table 1*):

Table 1 - Data sources segmentation

Source	Description
Home Credit Application data	Main sample. Static data for all applications.
Credit Bureau data	All client's previous credits provided by other financial institutions that were reported to Credit Bureau.
Credit Bureau balances data	Monthly balances of previous credits in Credit Bureau.

[40] Although the competition is for a long time over and many studies were already written based on this dataset, those studies were not considered during the current Study writing, thus it remains fully independent with respect to the used data sample.

[41] As an alternative to random split, n-fold cross-validation may be considered. However, taken into account huge number of available observations it was assumed that deviation of back-test results using random split from results obtained by cross-validation won`t be significant, but will greatly reduce computation time allowing more models and their settings to be tested.

Home Credit loans history	Monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit.
Home Credit credit-cards history	Monthly balance snapshots of previous credit cards that the applicant has with Home Credit.
Home Credit previous Application data	All previous applications for Home Credit loans of clients who have loans in our sample.
Home Credit installments payments history	Repayment history for the previously disbursed credits in Home Credit related to the loans in our sample.

Source: Own construction based on Home Credit's dataset description[42]

Since provided data are not aggregated and there are a lot of options to construct different explanatory variables based on past information covered by Credit Bureau data and Home Credit histories tables, it was decided that only application information of Home Credit Application Data will be used for this Study purposes. Such decision was additionally supported by the fact that not for every applicant information about its past credit activity exists or can be gathered, which would further complicate the process due to extended missing value treatment. An assumption to keep for our analysis only Application data is thus viewed as optimal, also because of the fact that main sample already provides us 120 variables to work with. Moreover, this study does not aim to develop the best possible model, but a sufficiently good and correctly designed model for comparison of different techniques.

However, it`s worth to mention, that application data contain in general much weaker predictors than those obtainable from so called "Behavioral data", which provide information about client's behavior regarding past loans and other obligations. In practice, for the initial PD model commonly both financial and behavioral (if available) predictors are used to reach higher prediction power.

Other important checks to perform for potential variables are:

- Does variable actually may influence Probability of Default? (Decision was made based on own experience and common sense)
- Does variable include outliers? (Appropriate treatment will be applied for each of studied models)
- Does variable suffer from missing values? (Appropriate treatment will be applied for each of studied models)

More about of data preparation and variables selection processes can be found in the application part for each tested model.

4.2 Pre-selection

The Train data sample provides us 122 variables to work with (including target variable and observation's identification number). However, not all potential explanatory variables have strong relationships with default probability. To not extend

[42] Source: Kaggle. Home Credit Default Risk. *Kaggle.com*.

modeling time with meaningless calculations the univariate analysis procedure were firstly applied, following by in an initial preselection that will be kept for all studying models.

The procedure starts with summary statistic for each variable is generated and analyzed. Based on its structure and description variables are classified into sub-groups:

- continuous - the variable is numeric and contains large number of unique values
- categorical - the variable is non-numeric or numeric with small number of unique values[43]

Additionally, each variable is checked for ordinality – if there is a logical order of values of such variable (e.g., education level) – an important step for future binning procedure to allow merging only of neighboring categories.

As a part of missing values and outliers' treatment all continuous variables were quantile binned. This ensures us following qualities of data:

First, since missing values are not always exclusively data collection errors but may provide us an impactful information of relationship between predictor and target variable, it will be irresponsible to simply remove corresponding observations from the working sample. By binning variable's values, we separate all missing values into its own category, keeping information in the model but also solving corresponding computational issues.

Second, outliers are assigned to the first or last category, weighted as all other observations inside these categories, which keeps information that outlying observations provide us in the sample, but prevents extreme values from affecting our estimations.

For initial binning of numerical continuous variables deciles were used. The whole range of values was divided into 10 nearly equal segments, where first group is 10% of observations with lowest values, second is next 10%-20% observations and so on.[44] Missing values are taken as an additional independent group.

After the initial binning procedure, the performance of each individual predictor was tested using Gini Index as a measure of prediction power (Gini methodology is described in previous chapter).

As for the pre-selection, only variables permitted for the further analysis are those that successfully fulfill following checks:

[43] For automated procedure to separate categorical and continuous numeric variables the number of unique values was set to 10, however manual adjustment for individual cases was allowed based on summary results and author's modeler opinion.

[44] 10-quantile approach was chosen to create adequate number of categories to allow more-less flexible fit and prevent huge number of categories affecting degrees of freedom. Also 10 groups for binning corresponds to 10 unique values chosen earlier as separation criteria between continuous and categorical variables.

1) Univariate Gini of the tested variable is higher than 0.05
2) Correlation between the tested variable and any other explanatory variable is lower than 0.6[45]

For the second check, if happens that correlation between two explanatory variables is higher than tolerated level then only the best-performing (the highest **Information Value**) variable is kept in the list. Since all variables were transformed into categorical by decile binning, correlation can`t be measured directly, as an appropriate reverse transformation into continuous form is required. Thus WoE-ization is applied, so all categorical values are replaced with its Weights, allowing continuous form even for originally non-numerical variables. Instead of using Gini to choose between correlated predictors, we likely consider another metric – **Information Value (IV)**, that is smoothly derived from WOE.[46] More on how WoE-ization and IV work can be found in the Logit model's application part (chapter 8.2).

After preselection is complete all approved variables are expertly checked - they should show logical and economically meaningful relationships to the PD. That includes PD changing between categories linearly or in smile-shaped form (addressed with charts) and direction of these changes allowing reality-accurate interpretation. In case variable shows suspicious behavior or other non-computational errors are found, then such variable is removed from the further analysis and the preselection procedure is repeated until all variables satisfy described requirements.[47]

Results of the preselection and variables description can be found in the Table 2 below:

Table 2 - Pre-selection procedure results

Target variable

Variable[48]	Variable's code	Variable's description[49]	Gini Index	Information Value
Default status	TARGET	Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 - all other cases).[50]	N/A	N/A
Facility's Identification number	SK_ID_CURR	Unique ID of files in our sample.[51]	N/A	N/A

[45] Used threshold are set close to those applied in practice, requirements might be strengthened or relaxed a bit in order to obtain desired number of preselected variables.

[46] At this point WoE-ization is used only to make correlation check possible as a part of pre-selection. Obtained WoE-values won`t be used further and original binning values are kept, so all variables remain categorical. WoE-ization once again will be applied for the Logit model, however, whole WoE-transformation will be redone from the very beginning, taking into account additional manipulations described into Logit application section.

[47] Pre-selection is following the approach described in WITZANY, J. Credit risk management: pricing, measurement, and modeling. *Cham: Springer*, 2017.

[48] Few variables were removed from the analysis due to suspicious behavior or the lack of clear description.

[49] Description is given by data provider.

[50] Actual values of X and Y aren`t given in data description, so we can only assume that they are in line with regulator's Default Definition.

[51] A support variable only used for observation identification and manipulations with the data structure (e.g., selection, joining etc.) and does not enter any model.

Explanatory variables				
Gender	CODE_GENDER	Gender of the client.	0.0952	0.0386
Income	AMT_INCOME_TOTAL	Income of the client.	0.0507	0.0107
Annuity	AMT_ANNUITY	Loan annuity.	0.0878	0.0267
Good’s Price	AMT_GOODS_PRICE	For consumer loans it is the price of the goods for which the loan is given.	0.1631	0.0898
Education	NAME_EDUCATION_TYPE	Level of highest education the client achieved.	0.0942	0.0515
Family status	NAME_FAMILY_STATUS	Family status of the client.	0.0717	0.0216
Region Population	REGION_POPULATION_RELATIVE	Normalized population of region where client lives (higher number means the client lives in more populated region).	0.0883	0.0276
Age	DAYS_BIRTH	Client's age in days at the time of application, time only relative to the application.	0.1609	0.0813
New Employment	DAYS_EMPLOYED	How many days before the application the person started current employment (time only relative to the application).	0.1840	0.1083
New Registration	DAYS_REGISTRATION	How many days before the application did client change his registration (time only relative to the application).	0.0852	0.0266
New ID document	DAYS_ID_PUBLISH	How many days before the application did client change the identity document with which he applied for the loan (time only relative to the application).	0.1087	0.0371
Car age	OWN_CAR_AGE	Age of client's car.	0.0605	0.0225
Occupation	OCCUPATION_TYPE	What kind of occupation does the client have.	0.1515	0.0813
Region rating	REGION_RATING_CLIENT_W_CITY	Bank’s rating of the region where client lives with taking city into account (1,2,3).	0.0982	0.0506
Work in another city	REG_CITY_NOT_WORK_CITY	Flag if client's permanent address does not match work address (1=different, 0=same, at city level).	0.0775	0.0311
Organization	ORGANIZATION_TYPE	Type of organization where client works.	0.1441	0.0695
External rating 1	EXT_SOURCE_1	Normalized score from external data source.	0.1744	0.1511
External rating 2	EXT_SOURCE_2	Normalized score from external data source.	0.3044	0.3043
External rating 3	EXT_SOURCE_3	Normalized score from external data source.	0.3142	0.3314
Total area	TOTALAREA_MODE	Normalized information about total area of the building where the client lives.	0.0969	0.0362
New Phone	DAYS_LAST_PHONE_CHANGE,	How many days before application did client change phone.	0.1166	0.0465
Document 3	FLAG_DOCUMENT_3	Did client provide document 3.	0.0732	0.0281
Credit Bureau enquires	AMT_REQ_CREDIT_BUREAU_YEAR	Number of enquiries to Credit Bureau about the client one day year (excluding last 3 months before application).	0.0752	0.0194

Source: Own construction based on the dataset description[52] and personal computations.

[52] Source: Kaggle. Home Credit Default Risk. *Kaggle.com*.

After univariate analysis and preselection only 23 explanatory variables were kept. Among them are 8 categorical and 15 originally continuous variables. Those are to be tested later for the multifactor Logit. It was decided to use same set of short-listed variables for other alternative models to ensure a fair comparison among the final models.

4.3 Logit model 1 (with variables transformation)

Merging categories

We will start practical part of the Study with application of Logit. However, before running estimation, additional data modifications are required. Since all variables are now transformed into categorical form, it is necessary to secure that inside each explanatory presented categories are sufficiently differentiated with respect to default rates. A simple decile binning we used during the pre-selection does not separate categories efficiently enough, some of bins show nearly the same default rates, thus additional group merging is needed.

Few methods exist that can provide us an appropriate conclusion if two groups share the same level of default rate and are good candidates for merging. For this study we will utilize one known as **ArcSin test** of the heterogeneity of two binomial distributions.[53] Test can be applied for both categorical variables (to individual categories) and binned continuous variables (to individual quantile bins).

The idea of ArcSin test is that default events have binomial distribution, which is assumed for any two bins we would like to compare. Our goal is to test the hypothesis of both samples (in our case they are two categories of a single explanatory variable) to have the same distribution and consequently the same parameter (Default Rate), so they can be safely merged.

ArcSin transformation convert a binomial random variable into one that is approximately normal:[54]

$$y = arcsin\sqrt{\frac{X}{n}}, \tag{17}$$

where X is the number of defaults in one category and n equals to the number of observations for that category, so $\frac{X}{n}$ can be denoted as DR. Acquired variable y now can be used to compute t-statistic for testing the hypothesis of homogeneity of two groups. T-statistic is calculated as follows:

[53] Based on SHORE, H. Approximate Closed Form Expressions for the Decision Variables of Some Tests Related to the Binomial Distribution. *Journal of the Royal Statistical Society. Series D (The Statistician)* Vol. 35. 1986.

[54] BROMILEY, P.A., THACKER, N.A. The effect of an Arcsin Square Root Transformation on a Binomial Distributed Quantity. 2002.

$$t = \frac{y_1 - y_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}, \tag{18}$$

where y_i is the mean of normalized distribution for the i-th bin, n_i in number of observations in corresponding category and s_i^2 is distribution variance equaled to $\frac{1}{4n_i}$.

An automated cycle proceeds to calculate the p-value of the ArcSin test for all combinations of bins and merges bins for which the p-value is the highest and above the certain threshold. The procedure is iterated until no more bins can be merged.[55]

An important note is that while dealing with ordinal variable, we should consider the order of categories, so the test is applied only on two neighbouring bins, preventing merging of non-adjacent categories. Rest of the procedure is the same as for non-ordinal variables.

Main purpose of ArcSin test is to merge categories or quantiles for which the difference in their DRs is statistically insignificant.

Another goal of merging is to remove categories with number of observations too small to guarantee the stability of DRs. Thus, bins with total observations number less than 50 and defaulted observations below 5 are merged with another bin that has the closest default rate.

The merging procedure ends with author's expert-based manual adjustment of newly generated bins if such seems to be necessary.

Woe-transformation

Now with all variables finally being categorical and properly binned, we can build model using dummy variables to index each particular bin of each variable. However, it will significantly increase the number of predictors we are obliged to work with, since every category represents its own explanatory variable with its own estimated coefficient, also decreasing available degrees of freedom for hypothesis testing. To prevent it Woe-transformation is performed.

Weight of Evidence (WoE) displays a linear relationship with the natural logarithm of the odds ratio, which is also the dependent variable in logistic regression. Mathematically WoE can be written as:

$$WoE = \ln\left(\frac{\%\, of\ non - events}{\%\, of\ events}\right), \tag{19}$$

where ln stands for "natural logarithm" and "$\%\, of\ non - events$", resp. "$\%\, of\ events$" is distribution of non-defaulted cases, resp. defaulted cases.[56]

[55] As a cut-off for the ArcSin test was used a common p-value 0.05. No merging is performed if p-value is below the cut-off value.

[56] Based on ENGELMANN, B., RAUHMEIER, R. The Basel II Risk Parameters. *New York: Springer*. 2006

By replacing categorical values with its WoE values, we transform all predictors into set of continuous variables, greatly reducing the number of variables entering regression and thus saving some degrees of freedom. This way we also ensure the correct model specification since relationship between explanatory and target variables is now linear by the definition of WoE.

While speaking about Weight of Evidence it is important to also describe **Information Value** – parameter that we used as a variable pre-selection criterium. IV can be calculated like that:

$$IV = \sum (\%\, of\ nonevents - \ \%\, of\ events) * WoE, \tag{20}$$

The result can be viewed measurement of tested variable's importance. Higher IV means higher importance. A standard rule of thumb for using Information Value is presented in Table 3.

Table 3 – A standard rule of thumb for using Information Value

Information Value	Variable Predictiveness
< 0.02	Unpredictive
0.02 – 0.1	Weak predictive power
0.1 – 0.3	Medium predictive power
0.3 – 0.5	Strong predictive power
>0.5	Suspicious

Source: Own construction based on Siddiqi (2006)[57]

Hence IV can be used as another preselection criteria on pair with univariate Gini. Normally it holds that Gini and IV grow or decrease in parallel.

Stepwise selection

After all inputs are properly managed, an estimation is performed as was described in chapter 3.

During the estimation backwards stepwise variable selection was done to exclude statistically insignificant predictors and improve overall performance. The backwards stepwise selection starts with estimation of the full model that contains all short-listed variables. In the next step explanatory variable with the lowest significance is removed from the model. The process continues until no further variable exceeds significance threshold. The threshold can be determined using different criteria. Commonly used are variable's **p-value**, **Akaike Information Criterion** (AIC) and **Bayesian Information Criterion** (BIC).

In this study BIC will be used for the stepwise selection. The criterion is defined as:

$$BIC = k * \ln(n) - 2\ln(L), \tag{21}$$

[57] SIDDIQI, N., Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring. *SAS publishing*. 2006.

where L is the maximized value of our likelihood function, k is the number of estimated parameters and n is the sample size.

As function shows, BIC measures quality of model's fit while penalizing it for additional parameters, solving maximum likelihood disadvantage of choosing the highest possible dimension, thus helping to find optimal number of predictors from input set. The model will reach higher BIC value if the penalty for additionally included predictor outweighs this predictor's positive impact on the likelihood, meaning such variable is better to be dropped from the model. Logically, among two nested models we should prefer one with the lower BIC.[58]

Estimation results

As an outcome of stepwise selection procedure, model with lower BIC is chosen and estimated. Estimation results are presented in the Table 4.

Table 4 - First Logit model's estimation summary

Variable	Coefficient	Standard Error	Z-value	Pr (>t)[59]
(Intercept)	-2,4368	0,0082	-298,281	0
CODE_GENDER	-0,5891	0,0422	-13,9745	0
AMT_ANNUITY	-0,3724	0,0520	-7,1672	0
AMT_GOODS_PRICE	-0,5441	0,0282	-19,3125	0
NAME_EDUCATION_TYPE	-0,5131	0,0384	-13,3657	0
NAME_FAMILY_STATUS	-0,2644	0,0543	-4,8686	0
DAYS_BIRTH	0,2468	0,0361	6,8360	0
DAYS_EMPLOYED	-0,3531	0,0295	-11,9667	0
DAYS_REGISTRATION	-0,2887	0,0533	-5,4099	0
DAYS_ID_PUBLISH	-0,2796	0,0439	-6,3633	0
OWN_CAR_AGE	-0,6781	0,0559	-12,1221	0
OCCUPATION_TYPE	-0,2059	0,0311	-6,6111	0
REGION_RATING_CLIENT_W_CITY	-0,2756	0,0372	-7,4156	0
ORGANIZATION_TYPE	-0,3479	0,0398	-8,7377	0
EXT_SOURCE_1	-0,4973	0,0221	-22,4792	0
EXT_SOURCE_2	-0,7399	0,0151	-49,1353	0
EXT_SOURCE_3	-0,8444	0,0138	-61,3402	0
TOTALAREA_MODE	-0,3564	0,0430	-8,2844	0
DAYS_LAST_PHONE_CHANGE	-0,2116	0,0381	-5,5553	0
FLAG_DOCUMENT_3	-0,6973	0,0493	-14,1576	0

Source: Own construction based on the R-studio model's summary.

Let`s remind that our model has a following form:

$$logit\big(P(Y = 1|X_1, \dots, X_k)\big) = b_0 + b_1x_1 + \dots + b_kx_k, \qquad (22)$$

The Table 4 above is a classic output of regression's estimation performed with in-build functions in R-studio software. In the first column are listed explanatory variables that were kept for the final model based on BIC backward stepwise selection. The second column represents b_i coefficients estimated on the training data with respect to the WoE-ized explanatory variables x_i. Third column – **Standard error** –

[58] SCHWARZ, E., Estimating the dimensions of a model. *Annals of Statistics*. 1978.

[59] All values in Pr(>t) column are lower than 10^{-5} so they were rounded to zero.

measures the average amount that the coefficient estimates vary from the mean value of the response variable. Fourth column tells how far from zero the estimated coefficient is (measured in the standard errors), basically it is value of the coefficient divided by the standard error. “Pr” is a rounded probability of the coefficient been equal to zero derived from z-value and a standard normal distribution.

Results shows that out of 23 short-listed explanatory variables 19 made it to the final model – considering the fact that BIC penalizes high number of coefficients, it was expected to end up with smaller number of predictors, but such outcome is still acceptable. Based on p-value all final predictors can be considered statistically significant at least on 5% confidence level.

Applying estimated coefficients to modeling data we calculated PDs predictions and measured its quality with Gini index, receiving the value of **48.57%**. Noticeably the result is not sky-high, since for the model construction only Application data were used. Those data, as was said before, typically shows lower prediction power than Behavioral data. Hence, **48.57%** for in-sample Gini is an expected and tolerable outcome.

Back-test results

To secure robustness it is necessary to run performance tests on a new independent data sample – the Test sample. With the use of mapping derived from the training sample’s binning, test sample raw data were appropriately adjusted to the final model’s input structure and predictions for the Test sample were computed.[60]

Out-of-sample **AUROC** of the Logit model with WoE-ization results in **74.27%.**

Out-of-sample **Gini** of the Logit model with WoE-ization results in **48.55%**. Surprisingly model shows almost the same performance level on the test sample data, meaning we have successfully avoided overfitting. A bit unexpected but welcome outcome.

Out-of-sample **KS statistic** of the Logit model with WoE-ization results in **0.3630**.

Out-of-sample **Brier score** of the Logit model with WoE-ization results in **0.0691**.

For the purpose of this study the described model will be taken as a practical benchmark. Of course, used approach is mostly automatic and test-based, with expert opinion playing a minor role. There is still a space for improvement with alternative settings, techniques, and predictors, however, the constructed model is believed to be sufficiently fitted to fulfil study’s goals.

[60] To clarify, all model construction’s steps from the data analysis up to the final model’s estimation are done on the Train sample exclusively. Test sample is used only for back-testing purposes and does not influence the model construction anyhow. Same is true for any other model in the Study.

4.4 Logit model 2 (no variables transformation)

Data cleaning

In the PLTR Study authors claim that practices commonly applied to Logit model in order to reduce an impact of non-linear relationships are not much effective. We would like to verify this assumption on our dataset. To do so, we will rebuild previous Logit model excluding transformations of continuous variables and see the difference that unsolved non-linearity may cause. It will also allow us to decide about efficiency of binning and WoE-transformation.

Also, there is no need to rerun the whole pre-selection procedure as we will obtain the same results. Therefore our 23 already short-listed explanatory variables will be used, as was mentioned before.

Since tested non-linearity will only endanger continuous variables, for categorical variables binning is still applicable to merge small-size categories. For that reason, we will use already established mapping for non-continuous variables binning.[61]

For continuous variables, however, we will not use binning. Thus, we need to propose an alternative solution to deal with missing data and outliers.

For variables that suffer less than 5% of observations in missing data, incomplete observations were simply omitted.[62] If more than 50% of data are missing, such variable was dropped from the analysis.[63] Otherwise, missing values were replaced by variables' mean values.[64] For categorical variables missing values were incorporated into individual categories with respect to the applied mapping. Such approach is believed to be a simple yet sufficient solution.

To deal with outlier we will perform **winsorization**. Extreme values for continuous variables will be replaced with the chosen border values. In our case we have applied 99.9% (resp. 0.1%) quantile to replace values falling beyond these thresholds. This way we remove undesired impact of the most significant outliers by replacing only 0.2% of variables' values.

Model estimation

After major data issues are successfully solved, we can run estimation.

Continuous variables enter regression as they are, while categorical variables are replaced with dummy-variables. Backward stepwise selection described in previous chapter was used to find an optimal set of variables for the final model based on BIC.

Estimation results can be viewed in the table below:

[61] The binning is done without WoE-transformation, hence categorical variables are remains non-numerical and will enter model replaced by dummy- variables for each category.

[62] By doing so only 0.3% of observations were excluded, which is a tolerable number.

[63] Three variables were removed due to extreme number of missing values: OWN_CAR_AGE, EXT_SOURSE_1 and TOTAL_AREA_MODE.

[64]Mean replacement was done for variables: DAYS_EMPLOYED, EXT_SOURCE_3 and AMT_REQ_CREDIT_BUREAU_YEAR.

Table 5 - Second Logit estimation summary

Variable	Coefficient	Standard Error	Z-value	Pr (>t)[65]
(Intercept)	-0,2977	0,1068	-2,786	0,0053
CODE_GENDER[2]: MALE	-0,2892	0,0186	15,570	0
AMT_INCOME_TOTAL	-6,89 * 10^{-7}	1,13 * 10^{-7}	-5,413	0
AMT_ANNUITY	0,00001	9,15 * 10^{-7}	12,853	0
AMT_GOODS_PRICE	-3,86 * 10^{-7}	3,64 * 10^{-8}	-10,603	0
NAME_EDUCATION_TYPE[2]	-0,1313	0,0619	-2,119	0,0341
NAME_EDUCATION_TYPE[3]	-0,5135	0,0648	-7,927	0
NAME_FAMILY_STATUS[2]: Married	-0,1577	0,0178	-8,879	0
NAME_FAMILY_STATUS[3]: Separated	0,0375	0,0334	1,122	0,2618
NAME_FAMILY_STATUS[4]: Unknown / Widow	-0,0981	0,0421	-2,328	0,0199
DAYS_EMPLOYED	0,00007	4,65 * 10^{-6}	15,832	0
DAYS_REGISTRATION	0,00001	2,39 * 10^{-6}	4,942	0
DAYS_ID_PUBLISH	0,00004	5,28 * 10^{-6}	7,630	0
OCCUPATION_TYPE[2]	0,1842	0,0561	3,284	0,0010
OCCUPATION_TYPE[3]	0,2468	0,1121	2,202	0,0277
OCCUPATION_TYPE[4]	0,2989	0,0587	5,091	0
OCCUPATION_TYPE[5]	0,3466	0,0577	6,011	0
OCCUPATION_TYPE[6]	0,3475	0,0631	5,513	0
OCCUPATION_TYPE[7]	0,5838	0,0885	6,597	0
REGION_RATING_CLIENT_W_CITY	0,1759	0,0162	10,851	0
ORGANIZATION_TYPE[2]	0,0003	0,0434	0,075	0,9401
ORGANIZATION_TYPE[3]	-0,1855	0,0452	-4,101	0
ORGANIZATION_TYPE[4]	-0,0931	0,0453	-2,055	0,0398
ORGANIZATION_TYPE[5]	-0,0361	0,0428	-0,842	0,3997
ORGANIZATION_TYPE[6]	0,0452	0,0585	0,772	0,4399
ORGANIZATION_TYPE[7]	-0,1426	0,0470	-3,037	0,0024
ORGANIZATION_TYPE[8]	-0,4274	0,0772	-5,537	0
ORGANIZATION_TYPE[9]	0,3899	0,1039	3,751	0,0002
EXT_SOURCE_2	-2,1570	0,0394	-54,733	0
EXT_SOURCE_3	-2,7830	0,0430	-64,749	0
DAYS_LAST_PHONE_CHANGE	0,00006	0,00001	5,656	0
FLAG_DOCUMENT_3[2]: 1	0,2861	0,0190	15,070	0

Source: Own construction based on the R-studio model's summary.

The Table 5 is conceptually identical to the Table 4.

Results shows that out of 23 short-listed explanatory variables 16 made it to the final model – categorical variables are presented as a set of dummy-variables with first category been a reference group. Based on p-value all final predictors can be considered statistically significant at least on 5% confidence level.[66]

Applying estimated coefficients to the modelling data, we calculated PDs predictions and measured model's power with Gini index, resulting in value of **47.21%**. However, we are more interested in the out-of-sample performance.

[65] Values in Pr(>t) column that are lower than 10^{-5} were rounded to zero.

[66] Even though p-value of some variables are above 0.05 confidence level, those dummies are a part of categorical variables and should be assessed all together.

Back-test results

To secure robustness it is necessary to run performance tests on a new independent data sample – the Test sample.

Out-of-sample **AUROC** of the Logit model without WoE-ization results in **73.36%.**

Out-of-sample **Gini** of the Logit model without WoE-ization results in **46.73%.** A slight but expected decrease comparing to the in-sample Gini.

Out-of-sample **KS statistic** of the Logit model without WoE-ization results in **0.3491**.

Out-of-sample **Brier score** of the Logit model without WoE-ization results in **0.0698**.

4.5 Comparison I: Logit vs Logit

Let`s look into results of two tested Logit models within the Table 6.

Table 6 - Logit vs Logit comparison

Model\Metric	**AUROC**	**Gini Index**	**K-S statistic**	**Brier Score**
Logit with WoE	74.27%	48.55%	0.3630	0.0691
Logit without WoE	73.36%	46.73%	0.3491	0.0698

Source: Own construction.

A minor reminder: Both models were built using logistic regression. However, the first model "Logit with WoE" includes decile-binning transformation of continuous variables, following by WoE-transformation back into continuous form, which was performed to fix a potential issue of variables' non-linear behavior. The second model, on the other hand, undergo no such transformations and continuous variables enters model in their original form (only with some minor tweaks for missing values and outliers).

By comparing results of these two approaches, we can conclude that there is a slight difference in the performance. Second model, that was expected to suffer from non-linear behavior, shows slightly worse results with all four metrics, yet the difference is so small, that it can be easily caused by the randomness of the original sample's split. Different splits or cross-validation can be used to make more precise conclusion on that term.

The result can be interpreted in two ways: Either tested continuous variables have not suffered much from non-linear relationships in the first place or that WoE-ization had a little effect.

The true reason might be somewhere in-between. It is important to remind that only half of variables are continuous, which limits the impact of possible nonlinearity. Also, we haven`t tested possible interactions between explanatory variables, which might be a major source of non-linear relations.

To answer some of these questions let`s look at the performance of Random Forest model.

4.6 Random Forest 1 (Logit short-listed predictors)

Data preparation

To assess superiority of Random Forest over Logit model in terms of prediction power, we are going to grow one Forest model on the same list of variables we used in Logit models.

We started with raw values and remade the data transformation process. Fortunately, we need not to stress about outliers due to the Tree's splitting mechanism, that is capable to handle extreme values automatically. Additionally, we refrain from binning and merging categorical variables, since we prefer our data to remain flexible and do not restrict Random Forest's classification ability.

Occurrences of missing values we handle the following way:

- For categorical variables missing values are taken as an individual category.[67]
- Numerical Variables with less than 10 unique values are treated as categorical variables, thus missing values are taken as an individual category.
- Numerical continuous variables with more than 30% of observation being missing values we transform into categorical variables using 5%-quantile binning and keep missing values as an individual category.
- For continuous variables with less than 30% of observation being missing values we replace all missing data with mean values of corresponding variables and keep them in the continuous form.

The whole procedure was performed on the Train sample exclusively and obtained mapping is applied to the Test sample to allow smooth out-of-sample performance measuring.

Imbalanced dataset and overfitting

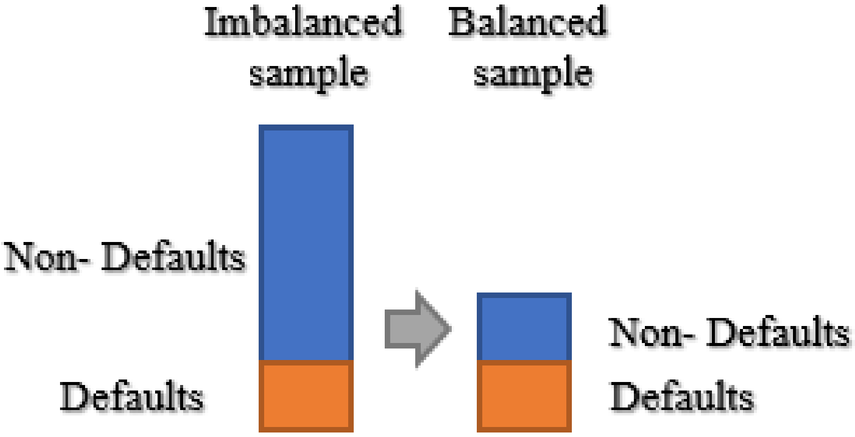
Parameters for Random Forest construction need to be properly adjusted if we wish for our model to perform well and without issues.

The dependent variable consists of two options: "defaulted status" and "non-defaulted status" with approximately 92:8 ratio, which results in the highly imbalanced dataset. As been said in the theoretical part, Random Forest assign the dominant group value to the formed homogenous nodes and use the majority of votes among all Trees to make prediction. If we think about it, while dealing with extremely imbalance dataset it becomes unlikely to hit the minor "defaulted" group. As a result, our model produces low error rates for the dominant category, yet extremely high error rates for the dominated one. Not only it weighs down prediction's accuracy, in the credit risk world false negative (permitting a loan to the future defaulter) outcome is more costly for the creditor than false positive one (rejecting loan to the non-defaulter). One solution is to rebalance dataset while keeping target variable's categories ratio close to 50:50.

[67] Only exception made for the variable ORGANIZATION_TYPE that contains 58 categories. Used software had trouble to deal with the variable of that many categories, thus it was decided to bin OCCUPATION_TYPE using the same approach we applied for the Logit model.

Rebalance is applied as a part of the bootstrap under-sampling during trees construction, so that formed training dataset retain 1:1 ratio between groups.[68]

Figure 9 - Example of under-sampling



Source: Own construction.

Another major issue is caused by the maximum depth of grown trees. If the minimum size of a single node is not somehow limited, Trees are allowed grow down until final node contains but one observation. Technically it may lead to the situation where Random Forest managed to perfectly fit input data, resulting in sky-high performance when back-tested on the developing data. Regretfully though, such performance is possible only for the Train data, while applying such model on the new Test sample leads to the significant drop in the power on all performance metrics. Such situation is called Overfitting.

To decrease the amount of overfitting we can limit splitting process. In our example such is done by setting the minimal size of nodes to the level that is enough to mitigate overfitting but not too high to suffer from model's strength going down. To find an optimal value a sequence of Forests was grown with the node size evenly increasing until the accuracy of the predictions on the test sample started to continuously diminish. The finally decided value of the minimum amount of observation in a single node is approximately equal to 1% of the size of rebalanced training dataset.

Model tuning

After crucial data issues are solved, we can start with tunning of less impactful parameters to achieve better performance.[69]

[68] Naturally it leads to the significant reduction of the inputted training data. Instead of original 246 000 observations we are left with about 40 000 observations. It is enough to grow a Forest but may have a negative impact on its performance, although this impact will be much lower than that one caused by imbalance data.

[69] Implementation of tuning in R is based on the guidelines DEEPANSHU BHALLA. A complete guide to random forest in R. *listendata.com*

Firstly, we will tune the number of trees to build for the Random Forest. Tuning starts with 100 individual trees to grow for a single Forest. The parameter is increased until out of bag error rate is stable and possibly minimal. Stability was verified by plotting the vector of error rates of predictions, where the i-th element is the out-of-bag error rate for all trees up to the i-th tree. This way we can see if error rate converges to some stable level while the number of trees grows. Obtained results show that already with 100 trees model`s performance is stable and increasing the number of trees leads only to a slight improvement.

After the optimal number of trees was found, we are to decide about the number of variables randomly sampled as candidates for each split. As been said before, randomness of used variables allows greater diversity among Trees, and as a result, helps to decrease correlation between these Trees and consequently the danger of multicollinearity.[70] The tuning starts with an empirically recommended value - square root of the total number of all predictors. The parameter is moved back and forth until out of bag error rate is stable and possibly minimal. We ended up with optimally 4 random variables to sample for each split.

After parameters are tuned, we rerun the Forest estimation with 1500 Trees[71] and save it for out-of-sample back-testing. The model's performance on the Train sample resulted in Gini of **57.64%**.

Back-test results

To secure robustness it is necessary to run performance tests on a new independent data sample – the Test sample.

Out-of-sample **AUROC** of the first Random Forest results in **75.2%.**

Out-of-sample **Gini** of the first Random Forest results in **49.4%.** Comparing to the in-sample outcome Gini dropped by 8.24 percent points (by 14.3%). Even with applied countermeasures a certain overfitting still presents.

Out-of-sample **KS statistic** of the first Random Forest results in **0.3733.**

Out-of-sample **Brier score** of the first Random Forest results in **0.2058.**

4.7 Random Forest 2 (pre-selection redone)

Data preparation

Considering specifics of the Random Forest approach it was decided that we are better to reconstruct the whole model from the scratch, while making changes to the pre-selection procedure, hopefully to achieve better results.

[70] In general, lower number of randomly sampled predictors reduces the correlation between trees, but also reduces strength of individual trees (and vice versa).

[71] A maximum number that current hardware allows, taking into account the dataset's size and parameters.

Univariate analysis up to the variables pre-selection fully repeat the same procedure for the Logit model, which includes data summary analysis and individual variables performance testing. The difference starts with setting pre-selection parameters. Since we do not much mind correlation between explanatory variables in the Random Forest, because of the RF model being capable to handle collinearity and multicollinearity by bootstrapping. On top of that, random variable sampling for split decision guarantees a certain level of diversity among trees. For that reason, we let loose the maximum correlation criterium up to 0.99 for the correlation check and maintain requirement of minimum 5% univariate Gini for the single-factor performance check. The result is 42 short-listed predictors, including 23 from the previous model. Theoretically, using more explanatory variables allows more options to construct the model and is expected to lead to the better performance.

Following data transformation procedure is identical to the one applicated for the first Random Forest. Mainly it refers to the missing data handling. Check chapter 11.1 for details.

The whole procedure was performed on the Train sample exclusively and obtained mapping is applied to the Test sample to allow smooth out-of-sample performance measuring.

Imbalanced dataset and overfitting

Similar to the first Random Forest, current model is also a subject of imbalance in target variable and overfitting.

Rebalance is applied with the bootstrap under-sampling during trees construction, so that training dataset retain 1:1 ratio for defaulted/non-defaulted observations.

Overfitting was mitigated by setting the minimum possible amount of observation in a single node to approximately 1% of rebalanced training dataset.

Model tuning

Alike for the previous model, we will first tune the number of trees to be built for the Random Forest. The tuning starts with 100 individual trees to grow. The parameter is increased until out of bag error rate is stable and possibly minimal. The results show that already with 100 trees model`s performance is stable and increasing the number of trees leads only to a slight improvement.

After the optimal number of trees was found, we are to decide about the number of variables randomly sampled as candidates at each split. The tuning starts with an empirically recommended value - square root of the total number of all predictors. The parameter is moved back and forth until out of bag error rate is stable and possibly minimal. We ended up with optimally 4 random variables to sample for each split.

After parameters are tuned, we rerun the Forest estimation with 1500 Trees[72] and save it for out-of-sample back-testing. The model's performance on the Train sample resulted in Gini of **57.68%.**

Back-test results

To secure robustness it is necessary to run performance tests on a new independent data sample – the Test sample.

Out-of-sample **AUROC** of the second Random Forest results in **74.2%.**

Out-of-sample **Gini** of the second Random Forest results in **48.4%.** Comparing to the in-sample outcome Gini dropped by 9.28 percent points (by 16%). Even with applied countermeasures a certain overfitting still presents.

Out-of-sample **KS statistic** of the second Random Forest results in **0.3686**.

Out-of-sample **Brier score** of the second Random Forest results in **0.2022.**

4.8 Comparison II: Logit vs Random Forest

Let`s look into results of tested Logit and Random Forest models within Table 7.

Table 7 - Logit vs Random Forest comparison

Model\Metric	AUROC	Gini Index	K-S statistic	Brier Score
Logit with WoE	74.27%	48.55%	0.3630	0.0691
Logit without WoE	73.36%	46.73%	0.3491	0.0698
Random Forest 1	74.7%	49.4%	0.3733	0.2058
Random Forest 2	74.2%	48.4%	0.3686	0.2022

Source: Own construction.

For the first model we can see, that despite our expectations, the Random Forest classifier just slightly outperform Logistic regression with respect to the AUROC, Gini and K-S metrics. However, the gain in performance quality is insignificant and can be caused solely by randomness of the samples split. We again recommend using cross-validation in the future for more accurate results.

Brier score results are not exactly comparable though, because for RF models we were forced to use dataset rebalance. Naturally, by using the majority of votes on rebalanced sample we will obtain higher PDs.[73] Thus, an appropriate calibration of modeled scores is required, if we are to use Brier Score for the comparison of Logit and RF.

Surprisingly, the second Forest showed worse results than the first one, closely the same as the WoE-ized Logit model. Since for the second Forest we used extended set

[72] A maximum number that used hardware allows, taking into account the dataset's size and parameters.

[73] Higher predicted PDs are good to minimize difference between predicted scores and values for defaulted clients (which equals to 1), but at the same time it also causes greater differences between PDs and non-defaulted clients (their observation values are 0). Since non-defaulters are a dominating group, it leads to the negative effect of the PDs' shift being bigger than the positive effect, thus we observe higher value of Brier Score.

of variables, it was expected that newly added predictors will lead to the better fit and higher performance.

One possible explanation to the Random Forest "failure" is that the original application data does not suffer much from the non-linear relationships, which would otherwise allow Random Forest to significantly outperform Logit. Another reason may be that data structure is not very suitable to apply RF due to extreme categories imbalance in the target variable, which leads to the significant reduction of observations in the Train sample cause of required under-sampling.

Additionally, tuning and estimating Random Forest is quire time-consuming and hardware demanding process. Taking into account model's complexity there is no reason to prioritize Random Forest over WoE-ized Logit. Of course, such conclusion is only true for the currently tested dataset and situation may differ significantly for some other sample.

There is still a hope to for improvement by combining strong advantages of two tested approaches. Application of one such combination – Penalized Logit Tree Regression – is described in the following chapter.

4.9 Penalized Logit Tree Regression

Data preparation

Like the previous models, our PLTR will use the same set of 23 short-listed variables. As was described in the Random Forest application part, when working with Decision Trees we need not to solve outliers, but missing values issue still requires our attention. Those were solved in the similar to RF manner:

- For categorical variables missing values are taken as an individual category.[74]
- Numerical Variables with less than 10 unique values are treated as categorical variables, thus missing values are taken as an individual category.
- Numerical continuous variables with more than 30% of observation being missing values we transform into categorical variables using 5%-quantile binning and keep missing values as an individual category.
- For continuous variables with less than 30% of observation being missing values we replace all missing data with mean values of corresponding variables and keep them in the continuous form.

The whole procedure was performed on the Train sample exclusively and obtained mapping is applied to the Test sample to allow smooth out-of-sample performance measuring.

[74] Only exception made for the variable ORGANIZATION_TYPE that contains 58 categories. Used software had trouble to deal with the variable of that many categories, thus it was decided to bin OCCUPATION_TYPE using the same approach we applied for the Logit model.

Decision Trees for singletons and couples

Firstly, we need to grow all necessary one- and two-level Decision Trees as was explained in the theoretical part for the PLTR model in chapter 5.

As we have already discovered during the Forest model establishment, our dataset is dangerously imbalanced which leads to the poor performance of the DT classifier. Thus, we applied the familiar random under-sampling technic to maintain 1:1 ratio for the target variable's categories, unfortunately resulting in the similar decrease in the number of observations.

In the first step we have grown all one-level trees for each single individual variable. The maximum depth of the Tree was set to 1 to prevent further growth and minimum number of observations for each single node was set to 100. Again, Gini index was used as a measurer of diversity to find an optimal split. After is finally dataset been rebalanced, all 23 singleton-based trees were successfully constructed.

Next, we have combined all possible pairs of each two individual variables and grew two-level trees using same parameters as for singletons.[75] Expectedly it was not possible to build trees for some couples, signalizing that no meaningful interaction between these two predictors was found. In the end, based on 23 short-listed variables we have successfully constructed 150 unique trees out of 253 possible.

After all trees were grown and checked, we transformed our original not under-sampled Train dataset into the set of categorical predictors based on the classification results of all these trees, leading to the total 173 explanatory variables. Obtained dataset now describes all meaningful univariate and bivariate effects in our data caught by Decision Trees and is meant to be used as an input for the following regression.

Ridge regression for Adaptive Weights

Before entering Lasso regression, initial coefficients $\hat{\beta}_i^{ini}$ was estimated using **Ridge regression**. Applying:

$$\hat{\omega}_i = \frac{1}{\left(\left|\hat{\beta}_i^{ini}\right|\right)^{\gamma}}, \tag{23}$$

with weight adjustment coefficient γ been set to the standard 1. Hence, we have computed **Adaptive Weights** for later use in the penalty term of **Adaptive Lasso regression**.[76]

Adaptive Lasso regression

After Weights were successfully calculated, we re-estimate coefficients with Adaptive Lasso regression, using $\hat{\omega}_i$ weights as a penalty factor parameter.[77]

$$\hat{\beta}_{aLasso(\lambda)} = argmin_{\hat{\beta}} \left\{ -L(X_i; \beta_i) + \lambda \sum \hat{\omega}_i |\beta_i| \right\}, \tag{24}$$

[75] Except maximum possible depth, which is now logically set to 2.

[76] Application of Ridge and Lasso regressions is proposed by PLTR's authors.

[77] Implementation of Adaptive Lasso Regression is based on the guidelines CARVALHO, R. Adaptive Lasso: What it is and how to implement in R. *ricardocarvalho.com*

As in case of Ridge regression, an optimal λ was found using 10-fold cross-validation.

Applying estimated coefficients on the modeling sample we have calculated PDs predictions, then measured prediction power with Gini Index, resulting in **45.1%.**

Back-test results

To secure robustness it is necessary to run performance tests on a new independent data sample – the Test sample. Dataset was firstly transformed using classification map of Decision Trees, after that PDs were computed on estimated Lasso-coefficients. Lastly, prediction accuracy was measured.

Out-of-sample **AUROC** of the PLTR model results in **72.5%.**

Out-of-sample **Gini** of the PLTR model results in **45.0%.** Model shows almost the same performance level on the test sample data, meaning we have avoided overfitting.

Out-of-sample **KS statistic** of the PLTR model results in **0.3387**.

Out-of-sample **Brier score** of the PLTR model results in **0.0702.**

4.10 Classic Logit + PLTR

As reader may recall, we haven`t used any interaction term in our Logit model. To compensate this disadvantage, we would like to combine two already established model: the first Logit model we used as benchmark and our latest PLTR model we built in the previous chapter.

During Logit establishment we applied binning procedure to transform all original raw variables into optimally merged categorical variables, thus we would expect them to be more effective for fitting the model than one-leveled Decision Trees. For that reason, we will change all singleton-based Trees in the PLTR model with corresponding binned variables from the first Logit model.

Data preparation

For both Train and Test sample we have used binning and merging procedure for our 23 short-listed variables with respect to the chapter 8.1 Then we have reconstructed all couples' Decision Trees for these variables and created interaction-describing variables with respect to the chapter 14.2.[78]

Obtained dataset is to be our input into the following Ridge and Lasso regressions.

Ridge regression for Adaptive Weights

Once again we apply Ridge regression to estimate initial coefficients and calculate Adaptive Weights for the later usage in Adaptive Lasso regression.

The procedure is identical to the one described in the PLTR application chapter 14.3.

[78] Technically we end up with the same final input of 173 variables we had in the PLTR application. Just all 23 singletons were replaced with 23 binned explanatory variables from the Logit application.

Adaptive Lasso regression

Obtained Weights are used in the penalization term of Adaptive Lasso regression to re-estimate model's coefficients.

The procedure is identical to the one described in the PLTR application chapter 14.4.

Applying estimated coefficients on the modeling sample we have calculated PDs predictions, then measured prediction power with Gini Index, resulting in a value of **50.34%.**

Back-test results

To secure robustness it is necessary to run performance tests on a new independent data sample – the Test sample. Binned Test sample was firstly expanded for couples' variables based on the classification results of two-variables Decision Trees. After that PDs were computed on estimated Lasso-coefficients. Lastly, prediction accuracy was measured.

Out-of-sample **AUROC** of the PLTR model results **in 74.88%.**

Out-of-sample **Gini** of the PLTR model results in **49.76%.** Model shows almost the same performance level on the test sample data, meaning we have successfully avoided overfitting.

Out-of-sample **KS statistic** of the PLTR model results in **0.3750.**

Out-of-sample **Brier score** of the PLTR model results in **0.0688.**

4.11 Comparison III: PLTR vs Logit vs Random Forest

Let`s look into results of ALL tested models within Table 8.

Table 8 - PLTR vs Logit vs Random Forest comparison

Model\Metric	AUROC	Gini Index	K-S statistic	Brier Score
Logit with WoE	74.27%	48.55%	0.3630	0.0691
Logit without WoE	73.36%	46.73%	0.3491	0.0698
Random Forest 1	74.7%	49.4%	0.3733	0.2058
Random Forest 2	74.2%	48.4%	0.3686	0.2022
PLTR	72.5%	45.0%	0.3387	0.0702
Logit + PLTR	74.88%	49.76%	0.3750	0.0688

Source: Own construction.

We can see that the PLTR model we had so many hopes for is actually performing by all metrics worse than the benchmark WoE-ized Logit, and that difference is significant. It is still a dissent result yet provides no reason to prefer PLTR over a simpler well-worked Logit.

It was expected to receive lower performance for PLTR in comparison with Random Forest, since we used similar technique but with smaller Trees. However, using more complex combination of variables in PLTR (like for tri- and quadri-variate threshold effects etc.) should occasionally leads to even results.

What was not expected - is to get performance worse than with Logit build on non-binned data (second model). It seems that building model only on univariate and bivariate effects is not very effective, as long as data allow us to achieve a dissent result using simple linear regression. Of course, situation may change radically if acquired data extremely suffer from nonlinearity, which seems not to be our case.

On top of that, relatively poor performance of PLTR may be caused by outgoings of random under-sampling that we were forced to apply to solve target variable's imbalance issues, since the number of observations used to grow Decision Trees decreased significantly.

Further expansion of the PLTR to include Trees that are constructed on interactions between tree and even four variables may lead to the noticeable improvement. However, doing so exponentially increases complexity of model building and makes it quite hardware-demanding due to huge number of input predictors.

Far better results have been achieved with a combination of Logit and PLTR approaches in our last (sixth) model. Naturally, if we extend Logistic regression (which already provide us with good prediction power) by incorporating interaction terms (as we did for PLTR), it is likely to further raise model's performance. However, obtained gain in the performance is just 1 p.p. in our example, certainly a welcome but not significant improvement. Although such model remains relatively simple and transparent, the costs to rebuild already implemented model with all accompany expenses would probably exceed or at least nullify all profit from that enhancement. On top of that, there exist better and easier options how to incorporate interactions into the model.

5 Application on simulated data

Now that we have checked the performance of all 3 selected approaches to PD modelling, we would like to test correctness of statements we made based on the results. Specifically, the one assuming there is but a small impact of non-linear relationships in the original dataset, not enough to cause significant problems to the Logit model, so that both Random Forest and PLTR are unable to outperform the benchmark.

To somehow check the reliability of this statement it was decided to run simulations on a dataset specifically designed for that solo purpose. It is believed that the difference can be better viewed on the extreme examples, so we simulated Data Generation Process while including there a ridiculous number of interactions. And those made wonders.

5.1 Dataset simulation

For our purposes, we need not to simulate the whole new sample but only the target variable. Using the familiar Logit model estimated in the chapter 8 and adding there all possible couples' interactions of final 19 explanatory variables, we obtained new PDs vector, that were later transformed into binary "default status" variable keeping the same observed default rate like in the original sample. Mathematically the simulation function may be written as:

$$logit\big(P(Y = 1|X_1, \dots, X_k)\big) = b_0 + \sum_{i=1}^{p} b_i x_i + \sum_{i=1}^{p}\sum_{j=1}^{p} x_i x_j + \epsilon, \qquad i < j, \qquad (25)$$

where first two terms are our original Logit model with b_i being estimated coefficients and x_i being woe-ized explanatory variables. The third term is newly added interactions for all possible couples without repetition built on its woe-ized values. Finally, ϵ is a random term from normal distribution $\epsilon \sim N(0,1)$ that brings a bit of randomness into the simulation. Simulated PDs are than divided into two groups – defaulters and non-defaulters – using upper 8.06%-quantile, that equals to the original dataset's default rate.[79]

Table 9 - List of variables used for interactions simulation

Variable		
Gender	**New Employment**	**External rating 1**
Annuity	**New Registration**	**External rating 2**
Good's Price	**New ID document**	**External rating 3**
Education	**Car age**	**Total area**
Family status	**Occupation**	**New Phone**
Age	**Region rating**	**Document 3**
	Organization	

Source: Own construction.

[79] The whole procedure is performed on both train and test sample combined. After the new dependent variable is simulated, the dataset is again randomly resampled into new Train and Test while keeping previous ratio 80:20.

Now, if we think about it, using such simplified approach leaves us with the whole bunch of unintuitive and controversial relationships that would make no sense in the real world. The one cannot expect to observe such behavior of these variables in practice, but similar pattern may occasionally be found in different specific data. Moreover, the purpose of this simulation is not to be realistic, but to make a point using unnaturally extreme example - that in specific circumstances the combination of Logit and Decision Trees may outperform the benchmark and shouldn`t be easily discarded.

5.2 Simulated Logit

Of course, with new target variable our old models are no longer valid and thus complete redesign is required.

Since we have used in our DGP explanatory variables from the short-list, there is no need to rerun pre-selection procedure. Logit model construction starts with decile binning and ArcSin-test merging of our 23 predictors. Procedure's logic was described in chapter 8.1.

Binned variables went through woe-transformation and parameters estimation is carried out. BIC-based backward stepwise selection is applied to choose final set of predictors.[80] The estimated coefficients are then used to compute PDs on the Test sample and calculate our standard set of performance metrics.

Meanwhile, on the development Train sample the model achieved **69.1%** Gini.

5.3 Simulated Random Forest

Similar to the Logit, a whole new Random Forest was grown on the simulated data. Data preparation and further approach corresponds to those described in chapter 11.1.

Missing values were solved in familiar way:

- For categorical variables missing values are taken as an individual category.
- Numerical Variables with less than 10 unique values are treated as categorical variables, thus missing values are taken as an individual category.
- Numerical continuous variables with more than 30% of observation being missing values we transform into categorical variables using 5%-quantile binning and keep missing values as an individual category.
- For continuous variables with less than 30% of observation being missing values we replace all missing data with mean values of corresponding variables and keep them in the continuous form.

[80] Interesting, even though we had used only 19 variables to simulate defaults, our newly created Logit model ended up with 20 final predictors, while some original predictors were excluded by stepwise selection. This was probably caused by relatively high correlation between these predictors.

The Forest was grown using random under-sampling with replacement to solve dataset's imbalance issue. Afterwards new meta parameters were tuned - minimal node size and maximum number of tested variables for each split. Meta parameters that provided best performance results were kept.[81]

Using tuned parameters, the Random Forest of 1300 trees was created, reaching **97.92%** Gini on the developing sample. Again, overfitting is to be expected, thus performance metrics based on the Test sample is preferred and can be found in the fourth comparison (chapter 18).

5.4 Simulated PLTR

Based on the modeling sample results of previous models we already can see that RF handles new data much better than Logit. We are eager to know though, how PLTR model will perform comparing to these two.

We solved missing values in a similar to the RF manner and rebalanced dataset for the first step - Decision Trees building. Using all possible combinations of explanatory variables, we recreated one-level trees for all singletons and two-level trees for all couples of variables, keeping fully grown trees only, while failed trees were omitted.

Trees' mapping was applied at non-rebalanced Train sample, which was used as an input for Ridge regression to calculate Adaptive Weights for Lasso's penalty term. After that, Adaptive Lasso regression is used to re-estimate b_i coefficients.

Gini Index is computed to measure the model's performance on Training sample, resulting in **85.74%.** Test sample's results can be found in the following comparison chapter.

5.5 Comparison IV: Simulated data models

Let`s have a look at the results of models based on simulation data in Table 10.

Table 10 - Simulation data comparison

Model\Metric	AUROC	Gini Index	K-S statistic	Brier Score
Logit	84.14%	68.28%	0.5266	0.0615
Random Forest	96.83%	93.66%	0.8176	0.0813
PLTR	92.50%	85.00%	0.6940	0.0448

Source: Own construction.

That outcome is much closer to our theoretical expectations. Overall numbers are higher than those we have seen before for real-life data, which is expected, since Data Generation Process is known to us, and the same variables were used for modeling. Without the random component that we included into the DGP function we would probably obtain results even closer to 100% with a strong learner like Random Forest.

[81] In comparison to the original RF meta parameters were relaxed. Maximum number of random predictors for splitting was increased to 5, and minimum node size was reduced to only 20 observations.

As we can see, due to enormous amount of interaction included into DGP our data suffer more from non-linear relationships than it was in original sample, which allows Random Forest to fit data much better than Logit. The difference between models became quite significant on all performance metrics we used, even Brier Score provides better result[82]. An absolute difference of **Gini** is **25.38 p.p.** and relative is **37.17%.** Same astonishing results are shown by **K-S statistic** with **55.26%** relative difference. It makes RF an unquestionable winner, though its complexity may not allow its usage in practice (if we can still refer to any practice with such bizarre example).

Meanwhile, PLTR model is also performing quite well. We were assuming earlier for this model to achieve prediction power somewhere in-between, and so it did. **PLTR** provides **16.78 p.p. (24.49%) Gini growth** and **0.1674 (31.79%) K-S growth** against **Logit**, respectfully suffering **8.66 p.p. (9.24%)** and **0.1236 (15.11%) drop** against **Random Forest**. **Brier score**, which is now fairly comparable for both logistic regressions, shows **0.0167 points (27.15%)** improvement.

Although PLTR does not achieve prediction accuracy of Random Forest, it provides quite significant boost to the performance over the Logit model. Taking into account transparency and intuitiveness of the PLTR approach it will be wise to consider using Logit Tree instead of classic Logit regression. Of course, once again, this simulated example we provide is extreme, and since the process of PLTR construction is not automated neither properly implemented yet into statistical software like R-studio, it might be problematic to challenge every model with it. However, if there is a doubt about predictors' linearity or if a modeler find himself in a situation when Random Forest significantly outperform Logit, it might be a wise decision to give PLTR a try. To make more accurate proposal regarding practical usage it requires to study a wider range of different real-life datasets.

Let us also have a quick look at the results authors of PLTR presents in their study (Table 11).[83]

Table 11 - "Housing" dataset results comparison

Method	AUC	PGI	KS	BS
Linear Logistic Regression	0.7910	0.5524	0.4426	0.1230
Non-Linear Logistic Regression	0.8092	0.5677	0.4773	0.1130
Random Forest	0.9501	0.8364	0.7800	0.0670
PLTR	0.8977	0.7271	0.6599	0.0868

Source: Constructed based on the Table 2 of the PLTR study.

As follow from the study, authors built their models on real data from the "**Housing**" dataset that is available in SAS library. Results of their real data example are quite

[82] It`s still not correctly to compare RF and Logit by Brier scores, since dataset rebalance leads to entirely different PDs' scale as was explained earlier.

[83] Some parts of the original table were omitted since they contain methods or metrics that were not described in our study, thus only relevant part is kept.

promising: Logistic regression shows the lowest predictions quality, Random Forest performs the best and PLTR maintains its accuracy in-between. On top of that, for more advanced PLTR that incorporate tri- and quadri-variate effects authors mention performance being even closer to the Random Forest's values. It was concluded in the study that overall PLTR "outperforms traditional linear and non-linear logistic regression while being competitive compared to random forest"[84].

However, it is not clear from the study what exact techniques were applied during models' construction, especially for Logit[85]. For that reason, we can only take authors conclusion as an advice, but not as evidence, until detailed models' construction is found or independent recalculations on the same data are done.

[84] DUMITRESCU, E., HUÉ, S., HURLIN, C., TOKPAVI, S. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*. 2021.

[85] Authors mention linear, quadratic and interaction term for non-linear regression, but not precise data handling description is present. It can be guessed that, for example, our recommended WoE binning was not performed to effectively manage non-linear behaviors of variables. Also, such huge difference between Logit and Random Forest is quite suspicious.

6 Conclusion

The last part of the Study summarizes all work done in its previous parts. It also provides an overview of the results to the reader, while making some propositions for findings' practical application and regarding possible following studying.

In the first half of the Study, we properly described 3 models (Logit, Random Forest and PLTR) that can be effectively used to score new clients for the credit risk's purposes. Thus, our readers can receive a clear picture of these approaches' functionality, its most noticeable advantages/disadvantages, and about motivation beyond using them.

The second part consists of models' applications and intermediary comparisons of their results, which is meant to support or refute our theoretical expectations. Every step of models' construction was in detail described, so that anyone interested in independent recalculations or own implementation might proceed smoothly, which also allows deeper and more accurate critique of the Study.

The results of the Study once again are combined in the Table 12 right below. Description and analysis of presented numbers may be found trough corresponding comparison chapters I – IV.

Table 12 – Results' summary

Real-life data				
Model\Metric	**AUROC**	**Gini Index**	**K-S statistic**	**Brier Score**
Logit with WoE	74.27%	48.55%	0.3630	0.0691
Logit without WoE	73.36%	46.73%	0.3491	0.0698
Random Forest 1	74.7%	49.4%	0.3733	0.2058
Random Forest 2	74.2%	48.4%	0.3686	0.2022
PLTR	72.5%	45.0%	0.3387	0.0702
Logit + PLTR	74.88%	49.76%	0.3750	0.0688
Simulated data				
Model\Metric	**AUROC**	**Gini Index**	**K-S statistic**	**Brier Score**
Logit	84.14%	68.28%	0.5266	0.0615
Random Forest	96.83%	93.66%	0.8176	0.0813
PLTR	92.50%	85.00%	0.6940	0.0448

Source: Own construction.

As for the answer to our main question **"If PLTR is capable to outperform our benchmark Logit regression in terms of prediction quality?"** – we may conclude that calculations made on the real-life "Home Credit" dataset **DOES NOT** provide sufficient proof of PLTR superiority. Instead, it shows **PLTR underperform** all other tested models, which was assumed to be caused by **low impact of non-linear relationships** in the tested predictors.

The benchmark Logit model performs sufficiently well on its own, with quality of predictions being close to the level of Random Forest. **PLTR proved to be a possible but questionable solution for incorporation of interactions' terms into original Logit**, meant to improve performance even further. However, performance gain was not astronomical.

On the simulated data, which were specifically modeled to contain a **high number of interactions** to allow us fully applicate advantages of PLTR, it was proved that **PLTR may significantly outperform Logit under certain circumstances**. It was proposed to test PLTR performance on other datasets that are known to be heavily weighted with non-linear relations. Other market segments may also be studied for PLTR application if such behavior is observed.

In the future we suggest staying vigilant for the situation when PLTR can be efficiently applicated, as it might **turns up quite profitable**, but we also see **no reason for immediate challenging** already working steady models with this new approach.

Overall conclusion can be summarized in few main points:

- It was shown on the real-world dataset that well-constructed Logit reaches similar level of prediction power as Random Forest and thus PLTR can`t provide significant improvement.
- It was shown on simulated data that PLTR may provide significantly better results in case of high impact of non-linear relationships between explanatory variables.
- Strong non-linear relationships are believed to be the key to the PLTR and RF dominance over Logit.
- PLTR may be used to incorporate interactions' term somewhat effectively into the Logit model.

Based on the results we propose some possible directions for further studies:

- Test if PLTR, as an extension to the classic Logit to incorporate interaction (Logit + PLTR model), is an optimal solution comparing to other applicable techniques.
- Test PLTR vs Logit performance on more samples to receive some average outcome with different datasets.
- Test PLTR on data from other areas beyond credit risk sector, maybe in fields that show systematic occurrence of strong non-linear relationships in its popular predictors, also that suffer less from imbalance data in target variables.

Additional studies will allow more accurate judgment of Penalized Logit Tree Regression and are likely to open more possibilities for its application.

Bibliography

Literature

BREIMAN, Leo. Random Forests. *Machine Learning* ***45,*** *5–32*. 2001. [no direct citation] Retrieved from https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

BROMILEY, P.A., THACKER, N.A. The effect of an Arcsin Square Root Transformation on a Binomial Distributed Quantity. Tina Memo No. 2002-007, Internal Report. 2002. [cit. 27.03.2022] Retrieved from https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.497.5549&rep=rep1&type=pdf

CARVALHO, R. Adaptive Lasso: What it is and how to implement in R. *ricardocarvalho.ca.*

DEEPANSHU BHALLA. A complete guide to random forest in R. *listendata.com.*

DUMITRESCU, E., HUÉ, S., HURLIN, C., TOKPAVI, S. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*. 2021. Elsevier, vol. 297(3), pages 1178-1192. [cit. 27.03.2022]

ENGELMANN, B., RAUHMEIER, R. The Basel II Risk Parameters. *New York: Springer*. 2006. ISBN 978-3-540-33087-5 [no direct citation]

EUROPEAN BANKING AUTHORITY. Capital Requirement Regulation (CRR), Article 178. *eba.europa.eu*.

GLENN W. BRIER. Verification of Forecasts Expressed in Terms of Probability. *Monthly Weather Review*. 1950. [no direct citation] Retrieved from https://web.archive.org/web/20171023012737/https://docs.lib.noaa.gov/rescue/mwr/078/mwr-078-01-0001.pdf

GREENE, W.H. Econometric Analysis. 5th Edition. *Prentice Hall, Upper Saddle River*. 2003. ISBN 978-0130661890 [no direct citation]

MONDAL, Ariful. Classifications in R: Response Modeling/Credit Scoring/Credit Rating using Machine Learning Techniques. *rstudio-pubs-static.s3.amazonaws.com.*

SCHWARZ, E., Estimating the dimensions of a model. *Annals of Statistics*. 1978. [cit. 27.03.2022] Retrieved from https://www.jstor.org/stable/2958889?seq=1

SHORE, H. Approximate Closed Form Expressions for the Decision Variables of Some Tests Related to the Binomial Distribution. *Journal of the Royal Statistical Society. Series D (The Statistician)* Vol. 35. 1986. [no direct citation] Retrieved from https://www.jstor.org/stable/2987803?seq=1

SIDDIQI, N., Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring. *SAS publishing*. 2006. [cit. 27.03.2022] Retrieved from https://support.sas.com/content/dam/SAS/support/en/books/cedit-risk-scorecards/59376_excerpt.pdf

WILLIAMS, R. Using the margins command to estimate and interpret adjusted predictions and marginal effects. *Stata Journal* 12: 308-331. 2012. [no direct citation] Retrieved from https://www.researchgate.net/publication/254395746_Using_the_Margins_Command_to_Estimate_and_Interpret_Adjusted_Predictions_and_Marginal_Effects

WITZANY, J. Credit risk management: pricing, measurement, and modeling. *Cham: Springer*, 2017. ISBN 978-3-319-49799-0. [no direct citation]

Relevant studies

ADDO, P.M., GUEGAN, D., HASSANI, B., Credit Risk Analysis Using Machine and Deep Learning Models. *Risks.* 2018. [no direct citation] Retrieved from https://www.mdpi.com/2227-9091/6/2/38

IRIMIA-DIEGUEZ, A.I., BLANCO-OLIVER, A., VAZQUEZ-CUETO, M.J., A Comparison of Classification/Regression Trees and Logistic Regression in Failure Models. *Procedia Economics and Finance, Volume 26*. 2015. [no direct citation] Retrieved from https://www.sciencedirect.com/science/article/pii/S2212567115004931

KLINKERS, L., Non-Linearity Issues in Probability of Default Modelling. *University of Twente.* 2017. [no direct citation] Retrieved from https://essay.utwente.nl/73912/

LESSMANN, S., BAESENS, B., MUES, C., PIETSCH, S., Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings. *IEEE Transactions of software engineering, vol. 34, no. 4.* 2008. [no direct citation]

LESSMANN, S., BAESENS, B., SEOW, H., Benchmarking state-of-art classification algorithms for credit scoring: And update of research. *European Journal of Operational Research.* 2015. [no direct citation]

PERSSON, R., Weight of evidence transformation in credit scoring models: How does it affect the discriminatory power? *LUP Student Papers*. 2021. [no direct citation] Retrieved from https://lup.lub.lu.se/student-papers/search/publication/9066332

SHARMA, D., Improving the Art, Craft and Science of Economic Credit Rik Scorecards Using Random Forests: Why Credit Scorers and Economists Should Use Random Forests. *SSRN*. 2011. [no direct citation] Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1861535

SMITH, T., KIRASICH, K., SADLER, B., Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets. *SMU Data Science Review*. 2020. [no direct citation] Retrieved from https://scholar.smu.edu/datasciencereview/vol1/iss3/9/

WANG, Y., ZHANG, Y., LU, Y., YU, X., A Comparative Assessment of Credit Risk Model Based on Machine Learning. *Procedia Computer Science, vol. 174*. 2020. [no direct citation]

Web sources

Evispot. Area under the ROC Curve (AUC). *evispot.ai*.

Kaggle. Home Credit Default Risk. *Kaggle.com.*

Rdocumentation. *rdocumentation.org*

Stack Exchange. *stats.stackexchange.com*

Stack Overflow. *stackoverflow.com*

Figures and Tables