**AKTUAR MOLIYA VA BUXGALTERIYA HISOBI ILMIY JURNALI**

**Vol.** 6 Issue 03 | **pp.** 1-12 | **ISSN:** 2181-1865
Available online https://finance.tsue.uz/index.php/afa

# USING MACHINE LEARNING IN BANKS TO FORECAST LOAN DEFAULTS: PART 2

**Berdiyorov Bekzod Shoymardonovich,**
*Turin Polytechnic University in Tashkent*
*ORCID: 0009-0008-6021-3549*
*bekzod.berdiyorov@polito.uz*
**Berdiyorov Jahongir Shaymardon o'g'li,**
*JSC "KDB Bank Uzbekistan"*
*ORCID: 0009-0007-0029-9363*
*jakhongir.berdiyorov@kdb.uz*

**ABSTRACT.** For most traditional banks, lending activity remains the major source of revenue. However, there is always a possibility that some borrowers default on their loans and this situation creates credit risk for banks. To manage this risk, banks utilize different qualitative and quantitative methods to predict loan defaults. Currently, traditional algorithms such as logistic regression and decision trees are among the most popular models used to predict loan defaults. Their simple implementation and high accuracy make them preferred choice among credit analysts. However, rapid technological progress and increased computing capabilities of computers are creating opportunities to apply more advanced machine learning algorithms for managing credit risk. For instance, deep neural networks and recently developed Extreme Gradient Boosting algorithms (XGB) have been shown to exhibit high accuracy in wide range of classification tasks and have potential to exhibit high accuracy in loan default prediction.

*Keywords: banks, lending, loan defaults, machine learning, deep neural networks, Extreme Gradient Boosting algorithms.*

**ANNOTATSIYA.** Ko'pgina an'anaviy banklar uchun kredit berish asosiy daromad manbai bo'lib qolmoqda. Biroq, ba'zi qarz oluvchilarning kreditlarini kechiktirish ehtimoli har doim mavjud, bu esa banklar uchun kredit xavfini tug'diradi. Ushbu xavfni boshqarish uchun banklar kredit to'lovlarining kechikishini bashorat qilish uchun turli xil sifat va miqdoriy usullardan foydalanadilar. Hozirgi vaqtda logistik regressiya va qaror daraxtlari kabi an'anaviy algoritmlar kredit to'lovlarining kechikishini bashorat qilish uchun ishlatiladigan eng mashhur modellar qatoriga kiradi. Ularning amalga oshirish qulayligi va yuqori aniqligi ularni kredit tahlilchilari orasida afzal ko'riladigan tanlovga aylantiradi. Biroq, tezkor texnologik yutuqlar va hisoblash quvvatining ortishi kredit xavflarini boshqarishda yanada murakkab mashina o'rganish algoritmlarini qo'llash imkoniyatlarini ochmoqda. Masalan, chuqur neyron tarmoqlari va yaqinda ishlab chiqilgan ekstremal gradient kuchaytirish (XGB) algoritmlari turli xil tasniflash muammolarida yuqori aniqlikni namoyish etdi va kreditlarning qaytarib bo'lmaydigan holatlarini yuqori aniqlik bilan bashorat qilish imkoniyatiga ega.

*Kalit so'zlar: banklar, kredit berish, kreditlarning qaytarib bo'lmaydigan holatlari, mashinani o'rganish, chuqur neyron tarmoqlari, ekstremal gradient kuchaytirish algoritmlari.*

**АННОТАЦИЯ.** Для большинства традиционных банков кредитование остается основным источником дохода. Однако всегда существует вероятность того, что некоторые заемщики не выполнят свои обязательства по кредитам, и эта ситуация создает кредитный риск для банков. Для управления этим риском банки используют различные качественные и количественные методы прогнозирования дефолтов по кредитам. В настоящее время традиционные алгоритмы, такие как логистическая регрессия и деревья решений, являются одними из самых популярных моделей, используемых для прогнозирования дефолтов по кредитам. Их простота реализации и высокая точность делают их предпочтительным выбором среди кредитных аналитиков. Однако стремительный технологический прогресс и расширение вычислительных возможностей компьютеров открывают возможности для применения более совершенных алгоритмов машинного обучения для управления кредитным риском. Например, глубокие нейронные сети и недавно разработанные алгоритмы экстремального градиентного бустинга (XGB) продемонстрировали высокую точность в широком диапазоне задач классификации и имеют потенциал для высокой точности прогнозирования дефолтов по кредитам.

*Ключевые слова: банки, кредитование, дефолты по кредитам, машинное обучение, глубокие нейронные сети, алгоритмы экстремального градиентного бустинга.*

## INTRODUCTION

For this research, dataset for 2016-2025 period by Lending Club (2025) could be used. 80% of data is recommended to be used for training and 20% should be used to test model performances. The results are expected to show that Extreme Gradient Boosting algorithms (XGB) exhibits high prediction accuracy compared to other traditional models can only reach accuracy of 92%.

The recommended methodology provided in this research could be used to show the methods of training Multilayer Perceptron (MLP) neural network and XGB algorithms to predict loan defaults and compare their effectiveness against traditional models currently used in finance industry, namely logistic regression and random forests algorithms.

This paper represents a part of an extended research on the sense of continuation of "Using Machine Learning in Banks to Forecast Loan Defaults" by Berdiyorov (2024). This part of the extended research explains further literature review conducted and the second part of the recommended methodology.

## LITERATURE REVIEW

A recent study by Kumar et al. (2018) tried to predict loan default using neural networks. For research purposes 9500 observations from Indian bank with 13 explanatory variables was used. After several steps of feature engineering and principal component analysis, 18 explanatory variables were prepared. Next, the data was split into training and test set in the ratio of 90:10. The authors constructed a multilayer perceptron algorithm with 18-node initial layer, two 20-neuron hidden layers and 2 output layers, with softmax activation function. Although it was not possible to directly measure the relative performance of parameters in multilayer perceptron algorithm, the analysis exhibited 93% accuracy score. Based on trial-and-error approach, the authors identified 7 variables that have the greatest influence on loan default. These were annual income, relationship between the bank and borrower, debt-income ratio, occupation, home ownership, employment length and others.

Another research devoted to predicting bank loans was published in 2002 by Zurada. For his research, the author used 3,364 consumer lending data obtained from a credit institution for 2000-2002. Out of 3364 loans, 300 loans have 'default' status, indication of 8.9% default rate. 60% of data was used to train models and 40% was used for test purposes.

To predict loan defaults, 13 explanatory variables were used. Zurada (2002) implemented three models, logistic regression, decision trees and simple version of MLP algorithm, and compared their effectiveness. In addition, the author created an ensemble model which is based on average of the three models. Initially, the author trained and tested models on all data which was highly unbalanced. Under this scenario, all models exhibited overall accuracy of around 93.5%. However, the models' accuracy in classifying default ('bad') loans was poor, with MLP exhibiting the best performance (42.4%).

| Parameter | Explanation |
|---|---|
| Loan_Amt | Amount of the current loan request. |
| Mort_Due | Amount due on existing mortgage. |
| Prop_Value | Value of current property. |
| Debt_To_Inc | Debt to income ratio |
| Years_On_Job | Years on current job. |
| Derog_Rep | Number of major derogatory credit reports. |
| Crd_Ln | Number of credit lines. |
| del_crd_ln | Number of delinquent credit lines. |
| Crd_Inq | Number of recent credit inquiries. |
| Age_Trd_Ln | Age (in months) of oldest trade line. |
| Reason_For_Ln | Reason for loan (debt consolidation or home improvement). |
| Job_Cat | Applicants' job categories. |

*Table 1. Parameters. Zurada (2002)*

To manage this issue, author randomly selected 300 'non-default' loans and joined 300-'default' loans and created balanced data of 600 observations. Out of these, 480 observations were used for training and 120 observations were used for testing model performances. The author run three versions of models for each algorithm: 1) no data transformation; 2) no data transformation, but eliminated highly correlated variables; 3) with data transformation. The following table summarizes, Zurada's results. The table shows that the ensemble model, which is based on the combined results of the three models, exhibits the best performance, with overall accuracy reaching 80% (2$^{nd}$ case –

eliminated correlated variables). The best performance in classifying 'default' loans was also achieved by ensemble model in the 1st case, where no data transformation was applied.

In 2012, Zhou and Wang conducted research to predict loan defaults using random forests algorithm, which is considered an ensemble model and implements multiple decision trees to predict defaults. The data used, which contained over 150,000 samples, was highly skewed (13:1 ratio). Therefore, oversampling algorithm was used to build trees. This algorithm over-samples minority class ('defaults') and down-samples the majority class ('non-defaults'). For analysis, the dataset was split in to training and test data in the ratio of 80:20. To evaluate model, overall accuracy score and balanced accuracy score was used:

$$Overall\ Accuracy = \frac{True\ Positive + True\ Negative}{No\ of\ Positives + No\ of\ Negative}$$

$$Balanced\ Accuracy = 0.5\frac{True\ Positive}{True\ Positive + False\ Negative} + 0.5\frac{True\ Negative}{True\ Negative + False\ Negative}$$

Using 500 trees, the random forests algorithm provided accuracy score of 93.6% and balanced accuracy score of 57.6%.

| **Decision Tree** | **No Transformation, 28 vars** | **No Transformation, eliminated correlated variables, 14 vars** | **With Transformation, 28 vars** |
|---|---|---|---|
| Overall | 77% | 77% | 77% |
| Non-default | 84% | 84% | 84% |
| Default | 70% | 70% | 70% |
| **MLP** | | | |
| Overall | 73% | 76% | 72% |
| Non-default | 74% | 77% | 72% |
| Default | 71% | 75% | 71% |
| **Logistic Regression** | | | |
| Overall | 71% | 73% | 70% |
| Non-default | 69% | 72% | 72% |
| Default | 73% | 73% | 68% |
| **Ensemble** | | | |
| Overall | 79% | 80% | 75% |
| Non-default | 79% | 85% | 80% |
| Default | 80% | 75% | 73% |

Table 2. Zurada (2002) results

Another paper by Odegua (2020) focuses solely on XGB and its performance in predicting credit defaults. The author obtained data provided by lending bank on 4,346 consumer loans on 31 parameters.

| | Predicted Class | |
|---|---|---|
| | Good | Bad |
| Actual Class Good | 175 | 15 |
| Actual Class Bad | 14 | 670 |

*Table 1. Classification Report. Odegua (2020)*

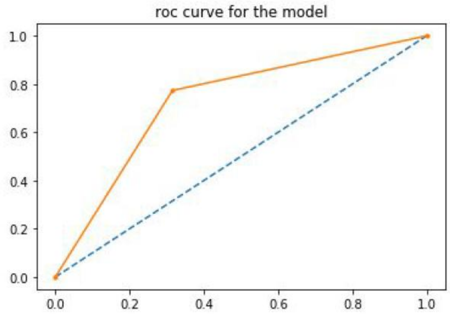


*Figure 1. ROC Curve. Odegua (2020)*

After series of data preprocessing and feature engineering stages, 20% of data was randomly selected for testing. During the model training process, Odegua (2020) performed 5-fold Grid Search algorithm to find optimal parameters. The results on the test data showed that XGB model's overall accuracy was 79%, with 97% precision and 79% recall score. Obtained AUC score was 0.71. More detailed analysis of confusion matrix showed that XGB correctly classified 670 'default' loans (out of 684), exhibiting 98% percent accuracy. The results identified age and location of the loan borrower as the most important predictors of default.

**METHODOLOGY**

As a continuation of the research done by Berdiyorov (2024), the Research question on predicting Bank Loan Defaults: Do advanced machine learning algorithms outperform traditional models used in finance industry?

The research is conducted using Python programming language and supplementary machine learning libraries. Specifically, pandas, matplotlib, numpy, seaborn and xgboost libraries will be used for data cleaning and model training purposes. Additionally, tensorflow external library will be connected to python to train MLP model. The research is based on Lending Club data, which contains information about more than one million customers on more than 40 features for 2016-2018 period. The research is divided into two parts: the first part focuses on predicting loans default for borrowers who already obtained loan. This is important for banks as identifying possible defaulter in advance will allow banks to measure risk and take preventive measures such as refinancing loan, changing contract terms so that the borrower will be able to fully repay the loan.

It should be noted that this research is based on machine learning models which rely on data mining. Therefore, instead of formulating theory in advance and testing hypothesis, algorithms rely on empirically identifying patterns and parameters that help to predict loan defaults by investigating data. In this context, the priority is given to developing models that maximize prediction accuracy and to examining relative importance of parameters. Investigating the exact marginal impact of explanatory variables is of secondary importance. In fact, for a bank it is more important to accurately estimate risk and profit rather than examining the exact marginal impact of individual parameters on default. Despite this, the obtained results will be compared to past papers and differences will be analyzed.

**Predicting Loan Defaults for Those Who Already Obtained a Loan**

As per Berdiyorov (2024), the objective could be to find answers to the following questions:

- *"Will the borrower fully repay his/her loan?*
- *If the borrower defaults on his/her loan, what percentage of total loan payment can be recovered?"*

The analysis will be conducted in the following order:

- ✓ Train, test logistic regression, random forests, XGB and MLP algorithms on data to classify loans as default and 'non-default' loans and compare their effectiveness.
- ✓ Obtain predictions based on the 4 models and compare with actual results. The comparison metrics are AUC score, ROC, accuracy, f1-score, precision and recall for both default and 'non-default' loans. However, the primary metric is recall, which measures how well the model identifies true defaults and non-defaults.

$$Accuracy\ Score = \frac{Total\ Number\ of\ Correct\ Predictions}{Total\ Predictions}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- ✓ Choose the best model for classification. The next steps will be based on the best-performing classification algorithm.
- ✓ Train and test OLS regression, random rorests regression, XGB regression and MLP regression models to estimate the amount of loan payment that can be recovered. For regression, total (loan) payment data will serve as dependent variable.
- ✓ Based on the classification predictions of the best model, for those borrowers who were identified as possible loan defaulters, estimate the amount payment that can be recovered in case of default. In other words, it is used to estimate the amount that can be recovered and the loss given default. The metric used is the Mean Squared Error (MSE). To be more precise, all four regression models will estimate loan payments for default borrowers, which are identified by the best performing classification algorithm.

Based on classification and regression results, estimate risk and expected profits for bank.

**Credit Scoring**

The second part of the research deals with credit scoring for new loan applicants. As per Berdiyorov (2024), "*the model can be easily modified to create a credit scoring model. For this purpose, only parameters which will be available at the time of loan application will be used. In the loan data, only last payment, revolving balance and utility information will not be available* *until loan is provided and, therefore they will be omitted. All other variables can be obtained for all borrowers requesting a loan. Obtained probabilities will be considered as credit scores with 0 being the perfect score, which means that the loan applicant has 0 probability of default and thus perfect credit score. The opposite is true for the score of 1.*

*Credit scoring model is performed in the following steps:*

- ✓ *Train, test logit, random forests, XGB and MLP algorithms on data to classify loans as default and 'non-default' loans and compare their results.*
- ✓ *Obtain predictions based on the 4 models and compare with actual results. The primary comparison metric is the number of defaults that are incorrectly identified as non-default. Since credit scoring model is used for new loan applicants, it is important for a bank to accurately classify defaulters. Other metrics are AUC score, ROC, precision, recall and accuracy scores.*
- ✓ *Train and test OLS Regression, random forests regression, XGB regression and MLP regression models to estimate the amount of loan payment that can be recovered. For regression, total (loan) payment data will serve as dependent variable.*
- ✓ *Based on the classification predictions of the best model, for those borrowers who were identified as most creditworthy borrowers estimate expected payment. The metric used will be the MSE.*
- ✓ *Under different threshold values, estimate expected profit for the bank.*
- ✓ *Compare model results and make conclusions regarding model performances."*

**Logistic Regression**

The first algorithm for classification is logistic regression. Logistic regression is a binary classification algorithm, which estimates probabilities based on logit function:

$$P(Y=1|X)=\frac{1}{1+e^{-z}},$$

$$where\ z=\beta_0+\beta_1X_1+\beta_2X_2\ldots\beta_nX_n, \qquad \beta_n-slopes, \qquad X_n-explanatory\ variables$$

$$P(Y=0|X)=1-\frac{1}{1+e^{-z}}=\frac{e^{-z}}{1+e^{-z}}$$

By calculating odds ratio and taking natural logarithm of the function, we linearize the logistic regression:

$$Odds\ Ratio=\frac{P(Y=1|X)}{P(Y=0|X)}=\frac{1}{1+e^{-z}}\div\frac{e^{-z}}{1+e^{-z}}=e^{z}$$

$$\ln(odds)=z=\beta_0+\beta_1X_1+\beta_2X_2\ldots\beta_nX_n$$

To run the regression, we need to assign probabilities to loan status (dependent variable). However, in our data, loan status is given by either 1 or 0. Therefore, Maximum Likelihood Estimator method is used iteratively to estimate the sigmoid curve that maximizes log-likelihood (product of probabilities). It should be also noted that regularization will be performed in the regression. This helps to avoid overfitting the train data. After regression is performed and the slopes of log-odds are obtained, we can identify the most important determinants of default and their impact on probability of default.

**Random Forests**

Random forests algorithm is an ensemble method that is based on decision trees algorithm. To make a prediction, the model randomly creates a bootstrapped sample from data and fits a decision tree for this sample. To fit a classification decision tree, information gain, Gini index or similar impurity criteria can be used. For the research, Gini index is selected as the base measure for random forests classification because of its faster calculation speed.

$$Gini\ Index = 1 - (probability\ of\ default)^2 - (probability\ of\ non\text{-}default)^2$$

Gini index shows the impurity, in other words, how well explanatory variables make splits. Lower the impurity measure, the better the explanatory variable split the data. To build a tree, weighted Gini index is calculated for randomly selected explanatory variables and the explanatory variable with the lowest Gini index is selected as the root node.

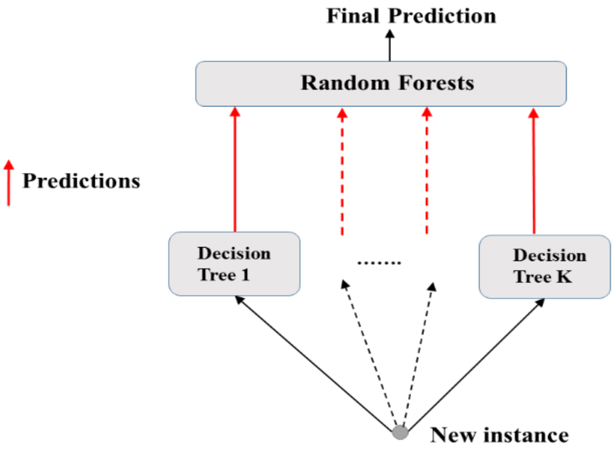


*Figure 2. Random Forests Algorithm. Source: DataCamp (2020)*

Next, the same process is repeated to build subsequent layers of decision nodes. At the point when splitting does not reduce the Gini index or when other early stopping algorithm is applied, the node will stop splitting. By repeating this procedure many times and building hundreds of decision trees, the algorithm makes final prediction based on predictions of all decision trees using majority voting scheme. For random forests regression, the procedure is similar. However, in this case MSE is used to make splits and prediction is made based on average of all predictions made by decision trees. The problem with both regression and classification random forests is that without preventive measures, trees may grow deep and overfit the data. As a result, accuracy score on test data can be low. Therefore, it is necessary to limit the depth of trees and specify other conditions for splitting.

**Extreme Gradient Boosting**

Generally, Extreme Gradient Boosting (XGB) is a modified version of basic decision trees, where decision is made based on many decision trees.

The difference between XGB and random forests is that random forests build trees independently of each other, while XGB builds trees sequentially. In other words, XGB makes some initial prediction and builds trees to improve this base case and each next tree is build based on errors of the previous tree and a new tree is trained to reduce the residuals of previous tree. At each step residuals are adjusted based on new tree's recommendations. To avoid overfitting, the trees recommendation is reduced by some factor, called the learning rate η. Industry standard for the learning rate is usually 0.3. This whole process is called 'boosting'. Boosting process continues until residuals are at their minimum or until specified number of iterations is reached.

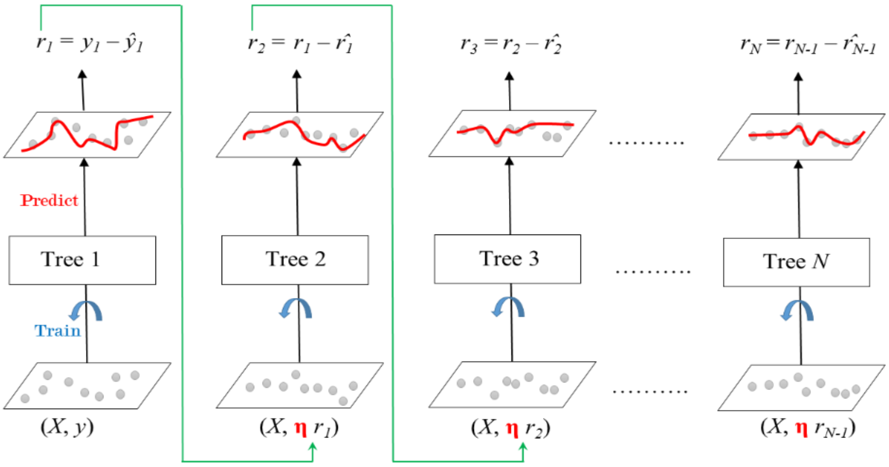


*Figure 3. XGB Algorithm. Source: DataCamp (2020)*

To make a prediction, XGB algorithm summarizes all trees' recommendations on how to minimize residuals. Similar to random forests, XGB hyperparameters can be tuned and cross validation allows to test the model's stability.

**Multilayer Perceptron**

Multilayer Perceptron is an artificial neural network algorithm which generally consists of 3 parts: the input layer, which represents explanatory variables, the hidden layer(s) where computations are performed and the output layer, where classification or regression results are displayed.

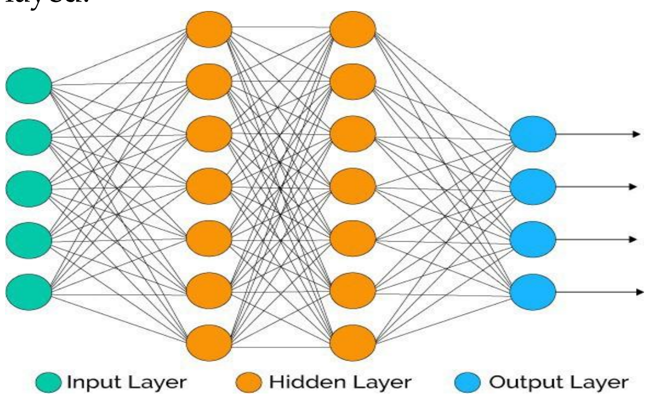


*Figure 4. MLP Algorithm. Source: Towards Data Science (2020)*

Each layer consists of several nodes and each of these nodes are connected to each node in the previous layer and the following layer. Starting with the input layer, each explanatory variable is treated as one node. These nodes are given weights and the weighted sum is obtained. For each node in the following layer, separate weights should be given and, therefore separate weighted sum is calculated.

To introduce non-linearity, some activation function is introduced. Otherwise, MLP algorithm will be reduced to ordinary least squares. There are several options available such as tanh, sigmoid, ReLU and others. For the research, ReLU activation function will be used due to its fast and easy implementation.

ReLU activation function: $\begin{cases} x = x, \ if \ x > 0 \\ x = 0, \ if \ x \leq 0 \end{cases}$. The value obtained after the activation function is applied will be used as the value of the node in the following layer. In addition, kernel regularization will be applied to avoid overfitting. Specifically, both L1 (Lasso) and L2 (Ridge) regularization will be applied to classification and regression models. The regularization strength is 0.01 for both regularizations, which is standard industry practice. Finally, batch normalization will be applied to reduce overfitting and to improve model performance by standardizing outputs of each layer. These steps will be repeated for each layer of neural network.

For MLP classification, there will be two nodes in the output layer, one node for each class (default and non-default). Softmax function will be applied to the output layer nodes to get probabilities of classes. Softmax function squeezes node values between 0 and 1 and converts them into probabilities of classes.

$$Softmax\ Function - \ Probability\ of\ Class\ j = \frac{e^{(z_j)}}{\sum_j^n e^{(z_j)}},$$

$$where\ z_j - value\ of\ node\ for\ class\ j$$

For MLP regression, there will be only one node in the output layer which shows the expected value. No activation is applied to the output layer of regression MLP, although ReLU will still be applied to hidden layers.

MLP fits the data by minimizing some loss function. For classification, categorical cross-entropy will be used. For regression, the weights will be adjusted to minimize MSE.

$$Averag\ Cross\ Entropy = -\frac{1}{N}[p_i * \ln(q_i) + (1 - p_i) * \ln(1 - q_i)],$$

$$where\ N - sample\ size,$$

$$p_i - true\ probability\ of\ class, \quad so\ it\ can\ be\ either\ 0\ or\ 1,$$

$$q_i - predicticted\ probability\ for\ class$$

$$Mean\ Squared\ Error = \frac{1}{N}[y_i - \hat{y}_i]^2,$$

$$where\ y_i - actual\ value\ and\ \hat{y}_i - predicted\ value$$

In both cases, algorithm optimizes weights iteratively using modified version of gradient descent optimizer. This optimizer is called adam, which is adaptive learning rate optimization algorithm. Adam is a powerful algorithm which finds optimal point by adjusting both learning rate and gradient. Although it is more complicated than other alternatives, adam combines momentum with learning rates and generally offers significant improvement in model optimization.

Adam algorithm:

$$m_j^t = \frac{\beta_1 m_j^{t-1} + (1 - \beta_1) g_{tj}}{1 - \beta_1^t}$$

$$v_j^t = \frac{\beta_2 v_j^{t-1} + (1 - \beta_2) g_{tj}^2}{1 - \beta_2^t}$$

$$w_j^t = w_j^{t-1} - \frac{\eta}{\sqrt{v_j^t} + \epsilon} m_j^t$$

$where\ \beta_1 = 0.9, \beta_2 = 0.999, \eta = 0.001, \epsilon = 10^{-8}$

$m_j^t - first\ moment\ estimate, which\ adjust\ the\ speed\ of\ gradient$

$v_j^t - second\ moment\ estimate, which\ adjust\ the\ speed\ of\ learning\ rate$

$g_{tj} - gradient\ of\ weight\ j\ at\ iteration\ t$

$w_j^t - weight\ of\ node\ j$

It should be noted that, a reader should understand that MLP is a neural network algorithm designed for classification and, therefore it is expected that the model will perform poorly in regression task. Despite this, MLP for regression will be performed to test this hypothesis.

**CONCLUSION**

The machine learning models used to predict loan defaults, can also be modified as credit scoring models, where probabilities are considered as scores. However, credit scoring models have lower accuracy than loan default models since the number of explanatory variables will not be available at the time of loan application. Credit scoring classification results show that logistic regression, random forests and XGB exhibit similar performance. However, XGB shows best performance in classifying 'default' loans, which is the primary criterion in credit scoring. With respect to regression, OLS and XGB regressor exhibit comparable performance, but OLS has lower MSE, which indicates better estimation of individual loan payments. MLP is the worst performer in both classification and regression. Based on this information, we can conclude that for credit scoring, where the number of parameters is small, benefits of more advanced machine learning models cannot be exploited. In such cases it is better to use simple linear models.

**REFERENCES:**

Berdiyorov, B. (2024). Using Machine Learning in Banks to Forecast Loan Defaults. International Conference "Banklarda Raqamli Transformatsiya: Ta'lim Bilan O'zaro Hamkorlik", TIFT University.

Odegua, R. (2020). Predicting Bank Loan Default with Extreme Gradient Boosting. Cornell University. Available from https://arxiv.org/abs/2002.02011 [Accessed 18 February 2020].

Lending Club Data (2025). Loan Default Dataset. Kaggle. Available from https://www.kaggle.com/wendykan/lending-club-loan-data/download

Kumar, U.A. (2018) Neural networks and statistical techniques: A review of applications. *Expert Systems with Applications*, 36(1), 2-17. Available from https://www.researchgate.net/publication/222546331_Kumar_UA_Neural_networks_and_statistical_techniques_A_review_of_applications_Expert_Systems_with_Applications_361_2-17

Zhou, L. and Wang, H. (2012). Loan Default Prediction on Large Imbalanced Data Using Random Forests, TELKOMNIKA Indonesian Journal of Electrical Engineering, 10(6), 1519-1525. Available from https://www.researchgate.net/publication/267864165_Loan_Default_Prediction_on_Large_Imbalanced_Data_Using_Random_Forests

Zurada, M. (2002). How Secure Are Good Loans: Validating Loan-Granting Decisions and Predicting Default Rates on Consumer Loans. *The Review of Business Information Systems*, 6(3), 65-84. Available from https://clutejournals.com/index.php/RBIS/article/view/4563/4654