STATISTICS IN TRANSITION new series, September 2024
Vol. 25, No. 3, pp. 49–67, https://doi.org/10.59170/stattrans-2024–027
Received – 31.08.2023; accepted – 30.05.2024

Comparing logistic regression and neural networks for predicting skewed credit score: a LIME-based explainability approach

Jane Wangui Wanjohi[1], **Berthine Nyunga Mpinda**[2], **Olushina Olawale Awe**[3]

Abstract

Over the past years, machine learning emerged as a powerful tool for credit scoring, producing high-quality results compared to traditional statistical methods. However, literature shows that statistical methods are still being used because they still perform and can be interpretable compared to neural network models, considered to be black boxes. This study compares the predictive power of logistic regression and multilayer perceptron algorithms on two credit-risk datasets by applying the Local Interpretable Model-Agnostic Explanations (LIME) explainability technique. Our results show that multilayer perceptron outperforms logistic regression in terms of balanced accuracy, Matthews Correlation Coefficient, and F1 score. Based on our findings from LIME, building models on imbalanced datasets results in biased predictions towards the majority class. Model developers in the field of finance could consider explanation methods such as LIME to extend the use of deep learning models to help them make well-informed decisions.

Key words: credit score, logistic regression, multilayer perceptron, explainability, LIME.

1. Introduction

The credit section is a critical function for financial institutions and banks as it contributes significantly to their revenue share. One of the ways they maximize their profits is by offering more credit. With increased competition and pressure to generate more revenue, financial institutions are searching for more effective ways to attract new creditworthy customers while minimizing losses. Credit approval puts the bank at risk of losing money while disapproval may lead to loss of customers, hence competition among lenders. Most financial institutions have suffered losses due to wrong decision-making in the past led by many of their customers defaulting on payments as discussed by Taghavi et al. (2015). This has created the need for mechanisms to identify and distinguish between eligible and non-eligible loan applicants (Bolton C. 2009). In the early years, lenders used personal relationships and subjective judgments to decide whether the applicant deserved a loan offer

[1]African Institute for Mathematical Sciences (AIMS), Cameroon. E-mail: jane.wanjohi@aims-cameroon.org. ORCID: https//orcid.org/0009-0004-3295-6426.
[2]African Institute for Mathematical Sciences (AIMS), Cameroon. E-mail: bmpinda@aimsammi.org. ORCID: https//orcid.org/0000-0003-2981-6436.
[3]State University of Campinas, Brazil. E-mail: oawe@unicamp.br. ORCID: https//orcid.org/0000-0002-0442-4519.

or not. The method was reliable and less time-consuming because of the smaller number of applicants. Prior information about the applicant is important in determining credit scores (Lahsasna et al. 2010).

A comparison study between Radial Basis Function (RBF) neural networks and LR was carried out by Taghavi et al. (2015) in Tose-Taavon Bank, Guilan. The study involved 376 cases of loan instalments with 7652 total credits within approximately five years. The cases were categorized into two groups based on credit risk: good customers were those with lower credit risk and bad customers with higher credit risk. Out of the total 376 cases, 302 were classified as good customers, while the remaining 74 cases belonged to bad customers. To evaluate the efficiency of the designed LR model, the study utilized Pearson correlation analysis to identify any significant relationship between the variables. The results indicated a significant negative correlation between the loan amount and the credit decision of bank customers, and a significant positive correlation between the number of installments and the credit decisions of bank customers, both at 99% confidence level. RBFs were trained with 320 samples and 50 samples were used for testing. RBFs showed a higher prediction accuracy of 88% than 82.7% of logistic regression. The study recommended efficient databases to store customers' information for easy access.

Dumitrescu et al. (2022), proposed a new credit scoring method called penalized logistic tree regression (PLTR), which combined decision trees and logistic regression to improve the accuracy of credit risk prediction. The paper also discussed the need for interpretability in the credit scoring industry, which showed why simpler models like logistic regression were still widely used despite their limitations compared to more complex machine learning methods. Several studies have shown that neural networks are more accurate, and logistic regression has performed better than neural networks on different datasets. This means there is a need for further investigation of the performance of neural networks and logistic regression in predicting credit scores. The application of deep learning methods in credit scoring has shown promising results as shown in Imtiaz and Brimicombe (2017) and Zhao et al. (2015) compared to traditional methods. Traditional methods like logistic regression have been widely used due to their simplicity and interpretability. Despite the effectiveness of neural networks, it is often difficult for stakeholders to understand and trust their decisions due to lack of transparency. This study aims to solve this problem by comparing the effectiveness of logistic regression and neural networks in predicting credit scores, using the LIME (Local Interpretable Model-agnostic Explanations) technique on binary and multiclass datasets to enhance the explainability of the neural network models. The goal is to determine which model provides a better balance between predictive accuracy and interpretability in the context of credit scoring.

This present study comparing Logistic Regression and Neural Networks for predicting credit scores using a LIME-based explainability approach offers several key contributions to predictive analytics and model interpretability. Firstly, it enhances understanding of how different models operate, especially in the context of finance where stakeholders require trust and clarity in automated decision-making processes. It carefully analyses the trade-offs

between the simplicity and transparency of logistic regression versus the accuracy and complexity of neural networks, aiding stakeholders in selecting the appropriate model based on their specific needs for accuracy and transparency. Furthermore, the study demonstrates the practical application of LIME, showcasing how local interpretable model-agnostic explanations can be effectively generated for complex models. This not only serves as a valuable guideline for other researchers and practitioners but also advances the field of explainable AI (XAI) by providing empirical insights into the effectiveness of explainability techniques across different models. The emphasis on explainability also promotes ethical AI practices, highlighting the importance of transparency in sensitive areas like credit scoring, which could influence policy and regulatory approaches.

Additionally, by making model decisions accessible to both technical and non-technical stakeholders, this study bridges a crucial gap, fostering communication and trust among model developers, policymakers, loan officers, and customers. Overall, these contributions are instrumental in advancing machine learning applications in finance, encouraging the responsible use of complex models while upholding high standards of accountability and transparency. Adding to this introduction, the rest of the work is organized as follows. In Section 2, we present a description of the mathematics behind the methods used in this study. Section 3 presents the analyses and results of our study, Section 4 gives the explainability with LIME and Section 5 gives the conclusion.

2. Methodology

The goal of this section is to present the mathematical description of the methods and models used for the validity of our results. The flowchart in Figure 1 illustrates the various processes performed in this study.

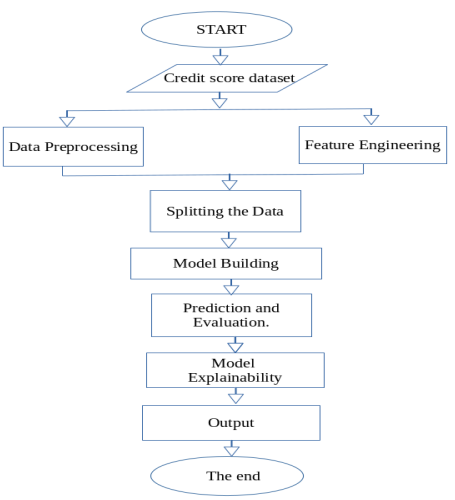


Figure 1: Credit Scoring Flowchart.

2.1. Data Preparation

2.1.1 Missing Value Imputation using MICE

Multiple Imputation by Chained Equations (MICE) is used as a data imputation technique to handle missing values in this study. The method assumes that values are missing at random implying that other columns can be used to determine a missing observation (Little and Rubin 2019). This is achieved by investigating acceptable estimates that best approximate the missing value using methods such as regression as discussed by Wulff and Jeppesen (2017). MICE creates multiple imputed datasets, where each fills in missing observations with valid data.

Given $X = (X_1, X_2, \ldots, X_k)$, a set of k features and each feature $X_j = (X_j^{observ}, X_j^{mis})$ where X_j^{observ} and X_j^{mis} are present and missing values respectively. The data imputation challenge is to get the conditional multivariate density $P(X)$ of X. Let t denote the number of iterations. With the assumption that data are missing at random, we repeat the Gibbs sampler iteration in Scheuren(2005);

$$\begin{aligned} X_1 &\sim P(X_1|X_2^t, X_3^t, \ldots, X_k^t) \\ X_2 &\sim P(X_2|X_1^t, X_3^t, \ldots, X_k^t) \\ &\vdots \\ X_k &\sim P(X_1|X_2^t, X_3^t, \ldots, X_{k-1}^t) \end{aligned}$$

A study done by Van and Oudshoorn (2000) demonstrated that when dealing with missing data in a multivariate normal distribution, iterating linear regression models like $X_1 = X_2^t\theta_{12} + \ldots + X_k^t\theta_{1k} + \varepsilon_1$ with $epsilon \sim N(0, \sigma)$ and estimates θ, can be a powerful tool for imputation as adopted in this present study.

2.1.2 Feature Selection with RFECV

The Recursive Feature Elimination with Cross-Validation (RFECV) method is applied for feature selection in this work. The method involves deleting features based on the importance it has on the model (Mustaqim et al. 2021). The final subset of selected features is chosen based on the performance of the model trained on the selected features using an independent validation set or through a nested cross-validation procedure. Suppose X is the input data matrix and y is the target vector. The algorithm is as follows:

1. Initialize a model M using all available features in X and evaluate its performance using cross-validation.

2. Rank the features based on their importance according to M.

3. Remove the least important feature from X, and evaluate the performance of M using cross-validation. If the performance metric improves, keep the feature removed. If not, add the feature back to X.

4. Repeat steps 3-4 until the desired number of features is reached or the model performance can no longer be improved. RFECV helps reduce the dimension of the dataset and that improves the performance of the model (Misra and Yadav, 2020).

2.1.3 Data Standardization

Machine learning offers distinct advantages in addressing some stringent assumptions associated with traditional statistical methods like Maximum Likelihood Estimation (MLE). Many machine learning models like the ones adopted in this study do not require assumptions about the data's distribution and can handle complex, nonlinear relationships automatically. This flexibility allows them to adapt more naturally to the actual structure of the data. Machine learning techniques such as logistic regression (borrowed from statistical methods) and neural networks are particularly adept at managing non-linearities and interactions without explicit modeling. Moreover, these methods often demonstrate robustness against outliers and noisy data, a common challenge in real-world datasets. In this study, we standardized the data before analysis to mitigate the possible effect of outliers. The mean is subtracted from each feature to standardize the data and then divided by the standard deviation. Standardization was used in our study to ensure all the inputs were on the same scale to avoid creating biased models. Given a dataset with n observations and p features, X_j ,$j = 1,2,\ldots p$, the mean and standard deviation of feature j is shown in Equations 1 and 2 respectively.

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

$$\sigma_j = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu_j)^2 \tag{2}$$

where μ_j and σ_j are the mean and standard deviation of feature j respectively, x_i is the i-th observation in feature j, and n is the total observations. Standardization of an observation i of feature j using z-score normalization is expressed as:

$$\mathbf{X}_{\text{standardized}} = \frac{x_i - \mu_j}{\sigma_j} \tag{3}$$

Standardization is an important technique in machine learning that helps improve the robustness and generalizability of models.

2.1.4 Data Balancing with SMOTE

Synthetic Minority Oversampling (SMOTE) was applied to address class imbalance. The method was first introduced by Chawla et al. (2002). It creates synthetic samples in the minority class for fair distribution between the classes. This method works closely as K-Nearest Neighbours (KNN). Let X be the input matrix with minority class instances, k be the number of nearest neighbours in consideration and m be the synthetic samples to generate. Let x_i be a minority data point x_{ij} an i^{th} observation of j^{th} feature and k random

selected neighbours. The new feature vector $X_{i,new}$ is given as:

$$X_{j,new} = X_j + (X_j - X_r) \times \alpha, \tag{4}$$

where X_j is the original feature vector, X_r is the feature vector selected at random and α is a random number between 0 and 1. SMOTE oversampling helps overcome overfitting observed in random sampling. Applying this technique enhances the accuracy of classifiers.

2.2. Classification Algorithms

Credit score prediction is a classification task in classification algorithms that classifies a customer as a defaulter or a non-defaulter. In supervised learning, the machine learns from labelled data automatically and improves its prediction capability with experience. The target label depends on the input features in the data and these inputs are meant to give accurate predictions of unseen data. Given a set of features $x_i \in X$ the model seeks to find a function f that maps the features to the output $y_i \in Y$, $f : X \to Y$. This function is used to make predictions of new data after it has learned from a set of labelled data. The model seeks also to get a function that gives a minimal difference between Y and the models' prediction $f(X)$, for $x \in X$. This study focuses on two supervised learning classification algorithms; logistic regression and multilayer perceptron.

2.2.1 Logistic Regression

Logistic Regression is a supervised classification algorithm applied when predicting a dependent categorical variable. The algorithm captures the probability of an outcome (e.g., 0 or 1) as a function of one or more input variables. The output of the logistic regression model is a probability value between 0 and 1. The algorithm is an extension of linear regression where the sigmoid function transforms the linearity (Imtiaz and Brimicombe 2017). There are various types of logistic regression. This study utilizes binary and multinomial logistic regression. The approximated value of y as a linear function of x is given as:

$$g_\beta(x) = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p = \beta_0 + \sum_{j=0}^{p} \beta_j x = \beta^{'} x, \tag{5}$$

where $g_\beta(x)$ is a linear combination of the input variables $x_1, x_2, \ldots, x_p$ and their associated weights $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ and $x_0 = 1$. As we saw earlier in supervised learning the main goal is to find a function that captures the relation between the input and output data. The sigmoid function transforms $g_\beta(x)$ in such a way that it takes values between 0 and 1 in the form:

$$h(\beta^{'} x) = \frac{1}{1 + e^{-\beta' x}}, \tag{6}$$

$$h(z) = \frac{1}{1 + e^{-z}}, \tag{7}$$

where $z = \beta^{'} x$ and $h(z)$ is the sigmoid or logistic function. From the notion of statistics, the regression coefficients (i.e. the parameters) provide information about each indepen-

dent variable's influence on the variations in the response variable (Hossain 2022). The conditional probabilities of the labels (0 & 1) for a given observation i are defined by:

$$p(y_i|x;\beta) = (g_\beta(x))^{y_i}(1-g_\beta(x))^{1-y_i}. \tag{8}$$

The likelihood function is used to find the values of parameters that maximize the likelihood of the observed data to train logistic regression (Hand and Henley 1997, Sewpaul et al. 2023), Mahmood 2024). The observations are taken to be independent. This function is given by:

$$L(\beta_0,\beta_1,\beta_2,\ldots,\beta_p) = \prod_{i=1}^{p}((g_\beta(x))^{y_i}(1-g_\beta(x))^{1-y_i}). \tag{9}$$

where p is the number of observations in the training data, y_i is the binary outcome for the i-th observation (0 or 1), and $g_\beta(x)$ is the linear combination of input variables with their associated weights for the i-th observation (Taghavi et al. 2015). An optimization algorithm such as gradient descent is then used to estimate the values of $\beta^{'}$. Gradient descent iteratively updates the parameters and finds the values that minimize the loss function (Zhao et al. 2015). The gradient of the log-likelihood function with respect to the parameters is given by:

$$\nabla L(\beta_0,\beta_1,\beta_2,\ldots,\beta_p) = \sum_{i=1}^{n}(g_\beta(x)-y)x_i. \tag{10}$$

The loss function is given by:

$$\mathscr{L}(\beta_0,\beta_1,\beta_2,\ldots,\beta_p) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i\log(g_\beta(x)) + (1-y_i)\log(1-g_\beta(x))\right]. \tag{11}$$

For prediction, a trained logistic regression model takes in a new instance and gives a probability output. A threshold such as 0.5 is set to make a binary prediction. The probability of the classes present adds up to one.

2.2.2 Multilayer Perceptron

Multilayer Perceptron (MLP) is commonly used in credit scoring. Unlike a percepton, MLP contains more than one hidden node or layer. MLP comprises three types of layers; the input, hidden and the output layer. Each layer contains interconnected nodes that transmit signals. The behaviour of hidden neurons is influenced by the input units and the weights connecting them, while the output neurons' behaviour is determined by the activities of the hidden neurons and the weights connecting them to the output neurons (Rodrigues et al. 2020). The main intention in neural networks is to get the ultimate parameters that best predict an output. The first step is forward propagation where each attribute is fit in the input layer. These inputs are assigned to random parameters called weights and passed to the first hidden layer. The hidden layer does some processes, like applying an activation function to each neuron to determine the output, passing this output to the next hidden layer for the same process, and then passing it to the output layer. The output layer gives the

predicted outcome, as shown in Equation 12:

$$\widehat{y_i} = h(\sum_{i=1}^{n} w_i x_i + b), \tag{12}$$

where $\widehat{y_i}$, h, w_i, x_i and b are the output, activation function, weights, inputs and the bias term respectively. The loss function is then used to compare the predicted output with the actual output. The cross-entropy loss function is used for classification purposes as shown in Zhang et al. (2023):

$$crossentropy = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p} y_{i,j}\log(\widehat{y}_{i,j}), \tag{13}$$

where n is the number of inputs, p represents the number of different classes present in Y, y_{ij} is the actual value and $\widehat{y}_{ij}$ is the predicted value. An optimizer is then introduced to the network. Optimizers are computational techniques used to adjust certain parameters of a neural network, such as the weights and bias term, to minimise the loss function and achieve precise outcomes. The learning rate, a hyperparameter, determines the magnitude of the step taken at each iteration as the algorithm moves towards a minimum of the loss function. Gradient descent is the most commonly used optimizer in neural networks (Agarwal et al., 2021). Equation 14 shows how the value of a parameter w_{ij} is updated, by subtracting the product of the learning rate $\eta, 10^{-}6 < \eta < 1$ and the partial derivative of the loss function $\mathscr{L}$ with respect to w_{ij} from its current value. This process is repeated for each parameter during training to minimize the loss function and improve the neural network's performance. The learning rate controls how much the parameters of a multilayer perceptron are updated during training.

$$w_{ij+1} = w_{ij} - \eta\frac{\partial\mathscr{L}}{\partial w_{ij}}, \tag{14}$$

$$b_{ij+1} = b_{ij} - \eta\frac{\partial\mathscr{L}}{\partial b_{ij}}, \tag{15}$$

where, w_{ij+1}, w_{ij}, η, and $\mathscr{L}$ are the new weights of observation i feature j, initial weight of observation i feature j, the learning rate and the loss function respectively. b_{ij+1} in Equation 15 is the updated bias term. The collective process of adjusting the parameters using an optimizer, computing the gradient of the loss function, and propagating the error backwards through the network is known as backpropagation as discussed in Zhang et al. (2023). After a multilayer perceptron is trained, it can be used for the prediction of unseen data and the output is determined by the learned parameters.

2.3. Performance Evaluation Metrics

Evaluation metrics are intended to estimate the model's ability to generalize unseen data in this study. Below are the metrics used to evaluate binary classification tasks in this work. A confusion matrix calculates evaluation metrics such as accuracy, balanced

accuracy, precision, and F1 score.

Table 1: Representation of a Confusion Matrix.

Class	**Positive Prediction**	**Negative Prediction**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

The accuracy(AC) performance metric gives the proportion of the correct predictions out of all the predictions made. The accuracy of a classification model is given as:

$$AC = \frac{TP+TN}{TP+TN+FP+FN}. \tag{16}$$

The accuracy metric lies between, 0 and 1, where values close to 1 show a good model performance. The accuracy metric is a commonly used evaluation metric. However, the metric is sensitive to imbalanced data. In cases of imbalanced data, balanced accuracy is a more efficient evaluation metric.

Balanced Accuracy (Bal AC) metric gives the average of the true positive rate (sensitivity) and the true negative rate (specificity). It ranges from 0 to 1. The metric is expressed as:

$$BalAC = \frac{Sensitivity + Specificity}{2}. \tag{17}$$

With

$$Sensitivity = \frac{TP}{TP+FN}$$

and

$$Specificity = \frac{TN}{TN+FP}$$

.

Additionally, this metric guarantees that all classes present are considered which gives a better view of the model's performance.

Matthews Correlation Coefficient (MCC) is the association between the actual and the predicted classes. This metric is commonly used when dealing with imbalanced datasets because it considers true positives, false positives, and false negatives of the model's predictions. MCC is expressed as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}. \tag{18}$$

The value of MCC ranges from -1 to 1, with values close to 1 showing a good model performance. Additionally, an MCC score of 1 indicates a perfect prediction, 0 shows a random prediction, and -1 indicates a wrong prediction.

2.4. Machine Learning Explainability

The success stories in solving diverse problems have led to a rapid expansion of the field, often referred to as a "golden age" because its end cannot be seen (Tran et al. 2022). However, there are several challenges that need to be addressed for further improvement in the performance and applicability of machine learning. Interpretability is one of the major challenges that researchers are working hard to solve so that the models are easy to understand and explain how they function. Some of the main reasons for machine learning explainability are: to validate the previous studies made on these algorithms, to increase the trust of end users of these models, and also the explanation methods may bring in the validation techniques (Molnar 2020). In this study, a Local Interpretable Model-agnostic Explanations (LIME) model explainability technique is applied. The aim of this method is to reduce the difference between artificial intelligence and humans by providing an explanation of how each feature is influencing a classifier. The method's main focus is to create a local approximation of our complex model (the trained model) for a particular discussed by Ribeiro et al. (2016).

Given a set of predictors $x \in X$, our trained model is denoted as f, and $g \in G$ where g is a simple model that comes from G, a family of interpretable models such as linear regression. An explainer $\xi(x)$ is defined as shown below:

$$\xi(x) = argmin_{g \in G} \mathscr{L}(f, g, \pi_x) + \Omega(g), \tag{19}$$

where π_x denotes the local neighbours of an instance x and $\Omega(g)$ represents a measure of the complexity that $g \in G$ does not explain. Equation 19 is an optimization task whose aim is to get a good approximation $\mathscr{L}$ and a minimum complexity $\Omega(g)$. Basically, the main idea is to look for a simple model g that approximates the trained model f in the local function $\mathscr{L}$ and keep its simple nature. To accomplish this, a new dataset is generated randomly with similar features as the original set. Predictions are made using the new data and the complex model f. The first loss term $\mathscr{L}$ is minimized by getting the highest accuracy on the new dataset using a simple linear model. Ribeiro et al. (2016), get the loss term by getting the difference in the sum of squared distances between the labels predicted by model f and the predictions of model g. π_x is also included to weigh the loss according to how close the data point is.

$$\mathscr{L}(f, g, \pi_x) = \sum_{y,y'} \pi_x(y)(f(y) - g(y'))^2. \tag{20}$$

The loss term $\Omega(g)$, ensures that the model keeps its simplicity nature. Ribeiro et al. (2016) state that in LIME, a sparse linear model is used to maintain simplicity. This model takes care of the second loss function since the model aims at producing as many zero weights as it can. This can also be achieved by using regularization techniques that help get a simple model with relevant features (Molnar 2020).

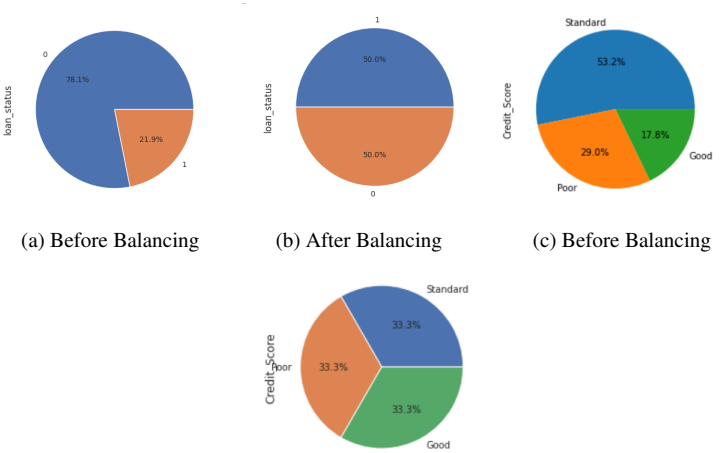
3. Data Analyses and Results

This section aims to provide a description of our datasets, an evaluation of the model's performance, and an explanation of the model's prediction. We used Python 3 software to

implement the various methods applied. Python libraries used include Pandas, Matplotlib, Seaborn, Scikit-learn as indicated by Pedregosa et al. (2011) and TensorFlow.

3.1. Data Description and Preprocessing

We used two types of datasets, a binary and multiclass case dependent variable. Both datasets contained categorical and numerical variables. The binary case dataset contained 32580 observations with a total of 12 columns while the multiclass case had 100,000 observations and 28 columns. In the binary case data, the dependent variable, 'loan status', takes a value of 0 or 1, where 0 indicates that the customer is a non-defaulter, while 1 is a defaulter. Conversely, in the multiclass case, the dependent variable, 'credit score', consists of three categories: good, standard, and poor credit scores. The simulated dataset was obtained using the make classification inbuilt function in scikit-learn library in Python. The binary case dataset contained 4011 missing values, but no missing values in the multiclass case. Various methods were employed, such as imputation of missing values, feature selection, standardization, data splitting, and balancing of classes in the training set to ensure that the data were consistent, complete, and appropriate. Good quality data and relevant features improve the accuracy and generalization of the model (Laborda and Ryoo 2021).



(a) Before Balancing (b) After Balancing (c) Before Balancing

(d) After Balancing

Figure 2: Distribution of Classes.

Figure 2a shows the distribution of customers' loan status, with approximately 78% non-defaulter customers and 22% defaulters. After preprocessing, the dataset was reduced to 32414 rows and 8 features for the binary case and 100000 rows and 16 variables for the multiclass case, using recursive feature elimination with cross-validation method. Figure 2c shows the proportion of customers' credit scores, with approximately 53% of the customers having a standard credit score, 29% having a poor one and 18% having a good score.

3.2. Model Assessment and Discussion

The performance evaluation metrics such as accuracy, balanced accuracy, Matthews Correlation Coefficient (MCC), precision and AUC-ROC score were used to examine the performance of the models used in the study. To assess the model's performance, we divided our data where 80% of the data trained the models and 20% for evaluation. The parameters used to train the MLP are shown in Table 2.

Table 2: Multilayer Perceptron Hyperparameters.

Parameters	**Levels in Binary Classification**	**Levels in Multiclass Classification**
Hidden Layers	2(8, 6 nodes in each layer)	2(10, 8 nodes in each layer)
Output Layer	2 nodes	3 nodes
Activation Functions	relu, relu, sigmoid	relu, relu, softmax
Dropout	0.1	0.1
Loss Function	binary cross entropy	categorical cross entropy
Optimizer	Adam	Adam
Iterations	50	50

3.2.1 Binary Case

Table 3 gives a view of how the models performed based on different evaluation metrics.

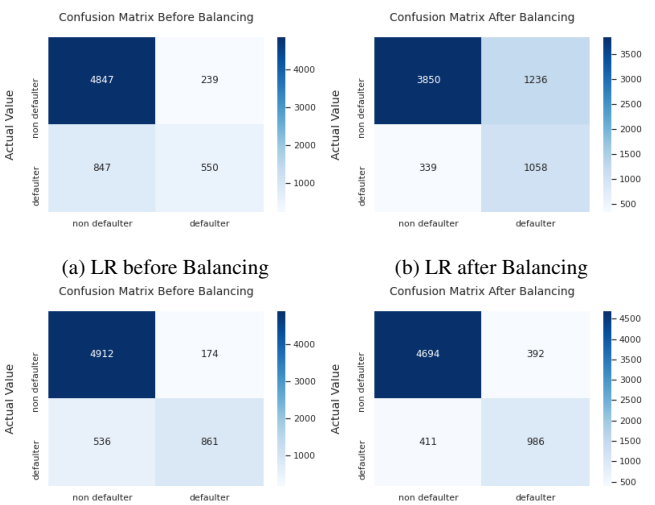
Table 3: Performance of LR and MLP for Binary Classification.

Evaluation Metrics	**Empirical Data**				**Simulated Data**			
	Imbalanced Data		Balanced Data		Imbalanced Data		Balanced Data	
	LR	MLP	LR	MLP	LR	MLP	LR	MLP
Accuracy	0.832	0.888	0.755	0.878	0.910	0.939	0.870	0.928
Balanced Accuracy	0.673	0.759	0.754	0.812	0.820	0.871	0.860	0.902
Precision	0.820	0.861	0.820	0.880	0.890	0.900	0.890	0.940
F1 Score	0.810	0.870	0.770	0.880	0.900	0.940	0.880	0.940
MCC	0.436	0.624	0.437	0.648	0.701	0.813	0.657	0.856
AUC-ROC score	0.827	0.759	0.828	0.812	0.924	0.871	0.925	0.902

From Table 3, we observe that the accuracy of all created models decreases after balancing the data. Given the sensitivity of accuracy on imbalanced data, we consider balanced accuracy metric. MLP exhibits a higher performance for both imbalanced and balanced data than logistic regression for both accuracy and balanced accuracy metrics. This highlights MLP's ability to predict the credit payment ability of customers accurately. The models demonstrate higher MCC scores after data balancing. MLP has higher MCC scores compared to LR. This shows MLP's ability to predict both non-default and default customers more accurately than LR.

The precision scores of all the models are found to be high on both balanced and imbalanced data. Upon comparing the performance of LR and MLP, we observe that MLP exhibited higher precision scores than LR. This indicates that MLP is more capable of accurately classifying true positive instances out of the total positives compared to LR, which means it is better at capturing fewer false non-default customers. The F1 scores of LR and MLP are reasonably high, indicating that both models are able to capture the underlying patterns in the dataset. Based on F1 score MLP performs best. Both models demonstrated a good performance based on the AUC-ROC score, as all models achieved scores higher than 0.5. This suggests that the models were able to effectively distinguish between non-default and default loan applicants. Notably, LR outperforms MLP by achieving a higher AUC-ROC score. This means that LR has a higher ability to distinguish loan applicants than MLP. We observe that the results of all the metrics appear similar in both the empirical and simulated data.

The confusion matrices in Figure 3 provide a summary of the number of correct and incorrect predictions made by the two models we have created on imbalanced and balanced data in our binary classification. The non-default class, which is the majority class, shows a higher count of instances that are correctly predicted on imbalanced data. However, after balancing the data, a more balanced distribution is achieved across all classes. This indicates that for imbalanced data, the models are more likely to predict the non-default class, but after balancing the data, the models become better at predicting the minority class as well. In addition, MLP demonstrates a greater ability to accurately predict the true classes and a lower number of misclassifications shown in the off-diagonal elements of the confusion matrices in comparison to LR.



(a) LR before Balancing (b) LR after Balancing

(c) MLP before Balancing (d) MLP after Balancing

Figure 3: Binary Confusion Matrix.

3.2.2 Multiclass Case

This section focuses on the model performance of our second dataset which has a dependent variable with three classes. The models created are meant to predict whether a customer has a good, standard or poor credit score. A good credit score means that a customer is likely to pay a given loan on time; a standard credit score, the customer pays the loan on time on average; and a poor credit score means that the customer is likely to default on a loan. A visual of the model's performance is shown in Table 4 for both imbalanced and balanced data. The simulated data showed similar performance scores for both imbalanced and balanced data.

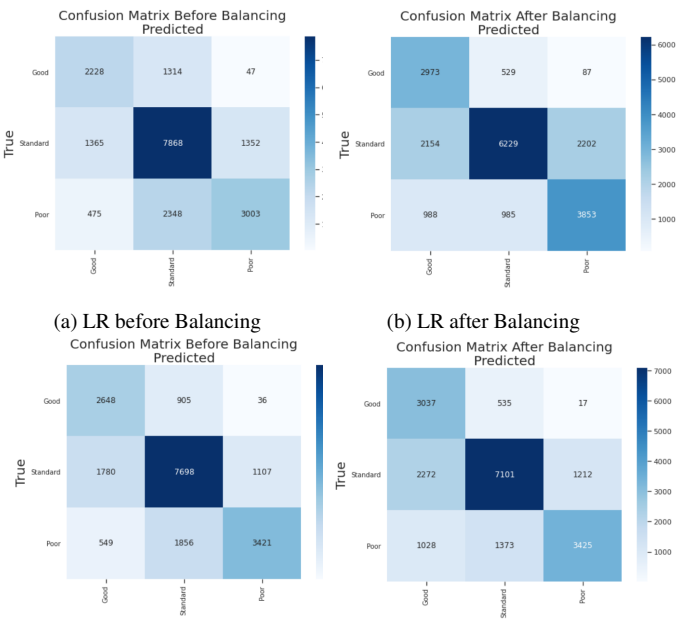
Table 4: Performance Metrics of LR and MLP for Multiclass Classification.

Evaluation Metrics	**Empirical Data**				**Simulated Data**	
	Imbalanced Data		Balanced Data			
	LR	MLP	LR	MLP	LR	MLP
Accuracy	0.645	0.680	0.65	0.678	0.629	0.687
Balanced Accuracy	0.600	0.709	0.685	0.718	0.525	0.621
Precision	0.660	0.710	0.690	0.720	0.620	0.700
F1 Score	0.650	0.670	0.650	0.680	0.610	0.680
MCC	0.422	0.485	0.469	0.515	0.345	0.473
AUC-ROC score	0.806	0.855	0.803	0.849	0.770	0.840

An analysis of the performance of our models with respect to the scores presented in Table 4 revealed that MLP exhibited a higher accuracy and balanced accuracy on both balanced and imbalanced data compared to LR. This is also observed in the simulated data. This means that MLP model was better at predicting the probability of a good, standard or poor credit score than LR. MLP has a higher precision score, 71% for imbalanced and 72% for balanced data compared to LR. MLP model also showed higher MCC scores of 0.485 and 0.515 for imbalanced and balanced data, respectively, in comparison to logistic regression, which showed MCC scores of 0.422 and 0.469 for imbalanced and balanced data, respectively. This indicates that MLP has a higher capacity to accurately predict the three classes of customers compared to LR. This means that MLP can capture fewer false positives. The MCC values can vary depending on factors such as class distribution, the separation of classes, interclass correlations, and the complexity of the model. It is important to consider these factors and evaluate MCC values on a case-by-case basis, as the interpretation of MCC values depends on the specific problem, dataset, and model performance in both binary and multiclass classification scenarios (Chicco et al. 2020). All the models perform well on AUC-ROC scores because they have scores greater than 0.5. This indicates that both models can distinguish between loan applicants with good, standard and poor credit scores. Considerably, MLP has higher AUC-ROC scores than LR. This shows that MLP has a higher ability to distinguish loan applicants than LR. Moreover, on the simulated data, MLP performs better than LR for all the performance metrics.

Figure 4 presents the confusion matrices of the multiclass classification models. Before balancing, as shown in Figures 4a and 4c, the standard credit score class had the highest

count owing to its majority class status, whereas the good credit score class had the lowest count as a result of being the minority class. Following the balancing of our data, in Figures 4b and 4d, the count of majority classes reduced therefore achieving a fair distribution across all classes. Our analysis of the confusion matrices presented in Figures 4b and 4d shows that the MLP model outperformed logistic regression in terms of correctly predicting class elements for the balanced data. This observation is consistent with the higher balanced accuracy of the MLP model, as reported in Table 4.



(a) LR before Balancing (b) LR after Balancing

(c) MLP before Balancing (d) MLP after Balancing

Figure 4: Multiclass Confusion Matrix.

4. Explainability with LIME

In order to obtain explanations of a model's prediction we use the LIME package in Python. We compile a list of the attributes used to train the model. We then define class labels (i.e. non-defaulter and defaulter for binary classification, and good, standard and poor for the multiclass classification), and then we create a function that provides the probabilities of each feature, fed it as an array. Passing all these components to the LIME explainer object, we input an observation into the explainer, which yields a prediction and offers insights into how each feature contributes to the classes present. Figures 5, 6, 7, and 8 display a visual representation of the prediction explanations for two instances provided by LIME. The results of a binary class prediction instance for imbalanced and balanced datasets are illustrated in Figures 5 and 6, respectively.

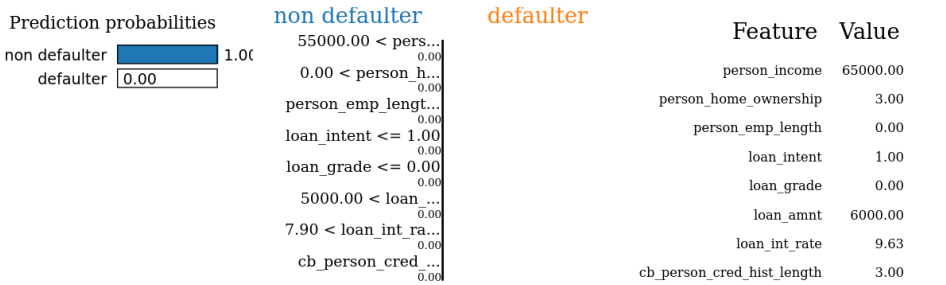


Figure 5: LIME explanations for a given observation on Binary Imbalanced data.

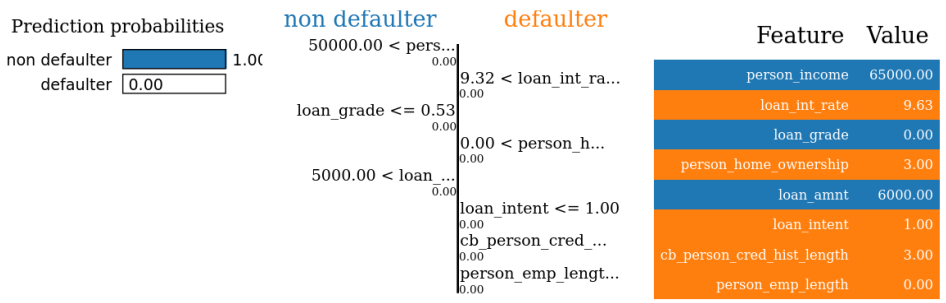


Figure 6: LIME explanations for a given observation on Binary Balanced data.

Figure 5 displays the prediction of a non-default customer, where all features were relevant in contributing to the non-default class prediction. In the scenario of balanced data in Figure 6, we observe a similar prediction of a non-default class on the same instance. However, in this case only three features significantly contribute to the non-default class: the applicants' income, loan grade, and loan amount. In both cases, the models give 100% assurance that the given loan applicant is a non-defaulter meaning that the given customer was likely to pay the loan on time. Figures 7 and 8 present the LIME explanation for a given instance on both imbalanced and balanced multiclass data, respectively.

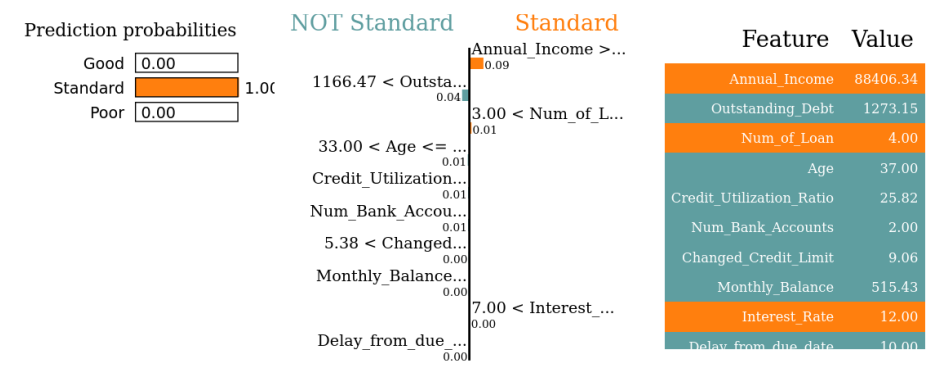


Figure 7: LIME generated explanations for a given observation on Multiclass Imbalanced data.

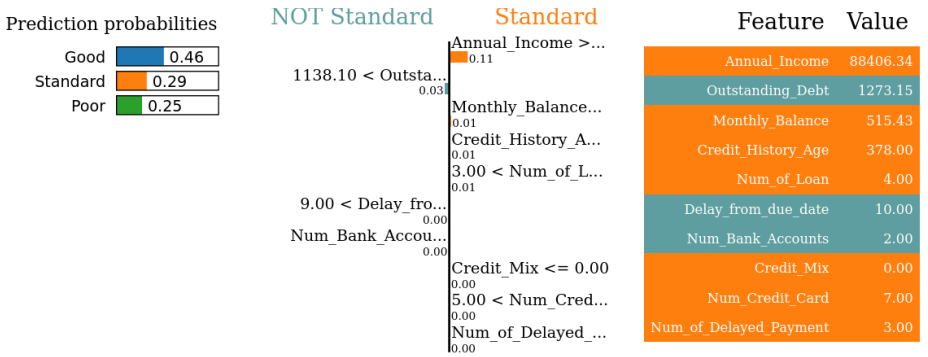


Figure 8: LIME generated explanations for a given observation on Multiclass Balanced data.

In Figure 7, we observe that the model predicts a standard credit score for the given loan applicant, where most features contribute to the decision. Additionally, the model shows 100% confidence in its prediction, suggesting a high likelihood that the customer will not pay the loan on time entirely. However, the prediction for the same loan applicant differs when it comes to balanced data, as shown in Figure 8. In this scenario, the model predicted with 46% assurance that the given applicant had a good credit score, indicating a likelihood of timely loan repayment. This information proves to be valuable to financial regulators in their decision-making process regarding loan offers.

5. Conclusion

In this study, we have compared the predictive ability of Logistic Regression (LR) and a Multilayer Perceptron (MLP) using two types of datasets, with an advanced model explainability technique - Local Interpretable Model-Agnostic Explanations (LIME). The findings show that all models performed better after the data were balanced. MLP had higher scores than LR in terms of balanced accuracy, Matthews correlation coefficient, and F1 score. From our findings, this study recommends that lending companies with small amounts of data use a logistic regression model but for companies with vast amounts of data a multilayer perceptron will ease their credit offer processes. The study also highlights the importance of using explainable artificial intelligence. With the LIME explanation approach, we were able to see how each feature influences the predicted class of a model for a given instance. We also found out that, models developed on imbalanced data are likely to show biased results, which may cost the lenders in the future. From what we were able to interpret, the LIME framework will enable developers to clarify to end-users the rationale behind a specific decision.

Disclosure of Competing Interests

The authors declare that they have no competing interests.

References

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R. and Hinton, G. E., (2021). Statistical regression modeling with R. *Advances in Neural Information Processing Systems*, 34, No. 1, pp. 4699–4711.

Bolton, C., (2009). Logistic regression and its application in credit scoring.*University of Pretoria (South Africa)*.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P., (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, No. 1, pp. 321–357.

Chicco, D., Jurman, G., (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, 21, pp. 1–13.

Dumitrescu, E., Hué, S., Hurlin, C. and Tokpavi, S., (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*, 297, pp. 1178–1192.

Hand, D. J., Henley, W. E., (1997). Statistical classification methods in consumer credit scoring: a review.*Journal of the Royal Statistical Society: Series A (Statistics in Society)*, Vol. 160(3), pp. 523–541.

Hossain, M. M., (2022). Statistical regression modeling with R. *Oxford University Press*, Vol. 5, No. 1, pp. 63–72.

Imtiaz, S., Brimicombe, A. J., (2017). A Better Comparison Summary of Credit Scoring Classification. *International Journal of Advanced Computer Science and Applications*, 8(7).

Laborda, J., Ryoo, S., (2021). Feature selection in a credit scoring model. Mathematics. *Mathematics*, 16, 9(7), p. 746.

Lahsasna, A., Ainon, R. N. and Teh, Y. W., (2010). Credit Scoring Models Using Soft Computing Methods: A Survey. *Int. Arab J. Inf. Technol.*, 7(2), pp. 115–123.

Little, R. J. Rubin, D. B., (2019). Statistical analysis with missing data, Vol. 793. John Wiley & Sons.

Mahmood, E. A., (2024). Robust Estimation of Multiple Logistic Regression Model.

Misra, P., Yadav, A. S., (2020). Improving the classification accuracy using recursive feature elimination with cross-validation. *Int. J. Emerg. Technol*, 11(3), pp. 659–665.

Molnar, C., (2020). Interpretable machine learning. *Lulu. com*.

Mustaqim, A. Z., Adi, S., Pristyanto, Y. and Astuti, Y., (2021). The effect of recursive feature elimination with cross-validation (RFECV) feature selection algorithm toward classifier performance on credit card fraud detection. *In 2021 International conference on artificial intelligence and computer science technology (ICAICST)* , pp. 270–275, IEEE.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., (2011). Scikit-learn: Machine Learning in Python. *the Journal of machine Learning research*, 12, pp. 2825–2830.

Ribeiro, M. T., Singh, S. and Guestrin, C., (2016). Why should i trust you?" Explaining the predictions of any classifier. *In Proceedings of the 22nd ACM SIGKDD international conference on knowledgediscovery and data mining*, pp. 1135–1144.

Rodrigues, P. C., Awe, O. O., Pimentel, J. S. and Mahmoudvand, R., (2020). Modelling the behaviour of currency exchange rates with singular spectrum analysis and artificial neural networks. *Stats*, 3(2), pp. 137–157.

Scheuren, F., (2005). Multiple imputation: How it began and continues. *The American Statistician*, Vol. 59(4), pp. 315–319.

Sewpaul, R., Awe, O. O., Dogbey, D. M., Sekgala, M. D. and Dukhi, N., (2023). Classification of Obesity among South African Female Adolescents: Comparative Analysis of Logistic Regression and Random Forest Algorithms. *International Journal of Environmental Research and Public Health*, 21(1), No. 1, p. 2.

Taghavi Takyar, S. M., Aghajan Nashtaei, R. and Chirani, E., (2015). The Comparison of Credit Risk between Artificial Neural Network and Logistic Regression Models in Tose-Taavon Bank in Guilan. *International Journal of Applied Operational Research-An Open Access Journal*, 5(1), pp. 63–72.

Tran, K. L., Le, H. A., Nguyen, T. H. and Nguyen, D. T., (2022). Explainable machine learning for financial distress prediction: Evidence from Vietnam. *Data*, 7(11), p. 160.

Van Buuren, S., Oudshoorn, C. G. M., (2000). Multivariate imputation by chained equations: Mice v1. 0 user's manual.

Wulff, J. N., Jeppesen, L. E., (2017). Multiple imputation by chained equations in praxis: guidelines and review *Electronic Journal of Business Research Methods*, 15, pp. 41–56.

Zhang, A., Lipton, Z.C., Li, M. and Smola, A. J., (2023). Dive into deep learning *Cambridge University Press*.

Zhao, Z., Xu, S., Kang, B. H., Kabir, M. M. J., Liu, Y. and Wasinger, R., (2015). Investigation and improvement of multi-layer perceptron neural networks for credit scoring. *Expert Systems with Applications*, 42(7), pp. 3508–3516.