A picture containing text, screenshot, font, logo

Description automatically generated

**MDSAA**

Master Degree Program in

**Data Science and Advanced Analytics**

**Machine Learning Operations**

**Group B:**

Carolina Confraria | 20220711  
Pedro Peças | 20220586   
 Rodrigo Silva| 20221360  
Tiago Figueiredo | 20220495

**NOVA Information Management School**

**Instituto Superior de Estatística e Gestão de Informação**

Universidade Nova de Lisboa

June, 2023

**Table of Contents**

[1.Introduction 3](#_Toc138443031)

[2.Data Description and Data Exploration 3](#_Toc138443032)

[3.Data Preprocessing 5](#_Toc138443033)

[3.1. Removing Irrelevant Column 5](#_Toc138443034)

[3.2. Handling Missing Values 5](#_Toc138443035)

[3.3. Removing Outliers 5](#_Toc138443036)

[3.4. Scaling Features 5](#_Toc138443037)

[3.5. Data Drift Test 5](#_Toc138443038)

[3.6.Feature selection 6](#_Toc138443039)

[4.Modelling and results 6](#_Toc138443040)

[5.Conclusions 7](#_Toc138443041)

[6.Further Suggestions and Limitations 8](#_Toc138443042)

[7.Appendix 9](#_Toc138443043)

# **1.Introduction**

In today's market, banks hold a significant position as gatekeepers of finance, influencing investment decisions and determining credit access. Accurately assessing creditworthiness is crucial for informed loan approvals. This report outlines a project that simulates the real-world deployment of machine learning models for credit scoring. The goal is to enhance credit scoring by predicting the probability of financial distress within the next two years. By improving credit scoring models, we aim to provide banks with more accurate risk assessments, enabling better-informed lending decisions.

The report explores various stages of the machine learning operations (MLOps) pipeline, including data preprocessing, model training, deployment, and monitoring. Simulating these processes provides practical insights into challenges and considerations of deploying machine learning models in credit scoring. A comprehensive drift test evaluation is conducted to assess model performance and accuracy over time, ensuring ongoing reliability and effectiveness in credit assessment models.

The project culminates in the development of LightGBM, an innovative model with enhanced predictive abilities in identifying customers likely to experience financial difficulties in the next two years. The integration of theoretical knowledge with real-world implementation has been instrumental in this pursuit. The aim is to make a significant impact on credit scoring, offering valuable insights for future research and practical applications in the industry. For further details, please refer to the following GitHub link: XXX.

# **2.Data Description and Data Exploration**

The code uses the read\_data function to read a CSV file and load the data into a pandas DataFrame object. The CSV file path is provided as input, and the function returns the DataFrame containing the data.

We employed Kendro and MLflow for the development of our project. Kedro plays a crucial role in our pipeline design (Fig. 01) by providing robust data management and model training capabilities. With Kedro, we can easily package and register trained models, streamlining the deployment process. Meanwhile, MLflow's tracking feature enables us to log and monitor training runs, capturing important details such as hyperparameters, metrics, and relevant outputs. This tracking functionality empowers us to analyse and compare results effectively, enabling informed decision-making throughout our machine learning operations.

During our data testing process, we aimed to evaluate both the quality of the data and the proper functioning of our code. These tests involved checking for missing values, validating the presence of positive values, and ensuring the consistency of data types. Additionally, we executed specific testing functions to confirm the accuracy and expected behaviour of the functions within the give\_me\_some\_credit.nodes module. These comprehensive tests allowed us to confirm that the data was appropriately read, accurately pre-processed, and that our models and data drift analysis yielded the anticipated outcomes. However, it is important to note that two tests, namely the test for missing values and data type consistency, did not pass successfully. In the forthcoming topics, namely Data Exploration and Data Cleaning, it will be provided detailed explanations of the results obtained and discuss the steps taken to address the identified issues.

The dataset for this project consists of 11 variables, including the target variable 'SeriousDlqin2yrs,' which indicates whether a person experienced a 90-day past due delinquency or worse. The remaining variables provide valuable insights into the borrower's creditworthiness and financial behaviour.

The 'RevolvingUtilizationOfUnsecuredLines' variable represents the percentage of the total balance on credit cards and personal lines of credit (excluding real estate and instalment debt) divided by the sum of credit limits. The 'age' variable indicates the borrower's age in years. Other variables include 'NumberOfTime30-59DaysPastDueNotWorse,' which represents the number of times the borrower has been 30-59 days past due but no worse in the last 2 years. The 'DebtRatio' variable indicates the percentage of monthly debt payments, alimony, and living costs divided by monthly gross income.

The 'MonthlyIncome' variable denotes the borrower's monthly income, while 'NumberOfOpenCreditLinesAndLoans' represents the number of open loans (e.g., car loans, mortgages) and lines of credit (e.g., credit cards) the borrower has. The 'NumberOfTimes90DaysLate' variable indicates the number of times the borrower has been 90 days or more past due. 'NumberRealEstateLoansOrLines' represents the number of mortgage and real estate loans, including home equity lines of credit.

Besides this, the dataset also includes 'NumberOfTime60-89DaysPastDueNotWorse,' which signifies the number of times the borrower has been 60-89 days past due but no worse in the last 2 years. The 'NumberOfDependents' variable indicates the number of dependents in the borrower's family, excluding themselves (e.g., spouse, children).These 11 variables provide crucial information for credit scoring and risk assessment, enabling better-informed lending decisions.

Upon conducting an initial statistical analysis of our variables, we have gained insights into their characteristics:

* "SeriousDlqin2yrs" variable has an average of 0.06, indicating a low occurrence of serious delinquency.
* "RevolvingUtilizationOfUnsecuredLines" variable has an average of 6.048, suggesting a relatively high utilization rate.
* The average age of our customers is 52.3 years, indicating a middle-aged customer base.
* "NumberOfTime30-59DaysPastDueNotWorse" variable has a mean value of 0.421, indicating a low incidence of late payments.
* "DebtRatio" variable shows that customers carry a relatively high amount of debt compared to their income.
* "NumberOfOpenCreditLinesAndLoans" variable indicates customers have an average of 8 or 9 open credit cards and loans.
* The average value of 0.757 for the "NumberOfDependents" variable suggests a relatively low occurrence of customers with dependents.

Bar charts (Fig.02- Fig.11*)* were used to examine the distribution of variables, where was possible to conclude that none of them displays a flawless normal distribution. However, the variable 'Age' showcases a distribution that closely resembles a normal distribution. Furthermore, the target variable 'SeriousDlqin2yrs' exhibits imbalanced data, with only 6.68% of our dataset pertaining to the 1 class in this binary variable. This could potentially represent a challenge in our predictions, requiring careful attention.

In the dataset, there are only two variables, 'MonthlyIncome' and 'NumberOfDependents', that contain NaN values. Nevertheless, it is important to note that the 'NumberOfDependents' variable has a significant proportion of missing values, accounting for 19.2% of the data. This indicates a substantial amount of missing information. Our team conducted a thorough analysis and determined that this high number of missing values is not specifically associated with the category having a smaller number of observations in the target variable.

Finally, we analyse outliers thought box plots (Fig. 12-Fig.21), noticing a high presence in several variables.

# **3.Data Preprocessing**

## **3.1. Removing Irrelevant Column**

The first column of the DataFrame, which represents the customer ID, is removed using the line data = data.iloc[:, 1:]. This column is not relevant for the analysis and modelling tasks, so it is eliminated.

## **3.2. Handling Missing Values**

The code handles missing values in two different ways:

For the 'MonthlyIncome' feature, the missing values are imputed using the k-nearest neighbours (KNN**)** imputation method. The column is prepared for imputation by reshaping it into a 2D array. Then, the KNNImputer from scikit-learn is applied with n\_neighbors=5 and weights='uniform' parameters to impute the missing values. The imputed values are then replaced in the original DataFrame.

For the 'NumberOfDependents' feature, the missing values are filled with the mode (most frequent value) of the column. The mode is obtained using the mode() function, and the missing values are replaced with the mode value using the fillna() method.

## **3.3. Removing Outliers**

We tried two different approaches to handling with the outliers: Interquartile Range Method (IQR) and manually.

The IQR method delete around 49% (which is very high percentage) of our data so we decide to apply the manual approach from the variables: 'DebtRatio', 'MonthlyIncome', 'NumberOfTime30-59DaysPastDueNotWorse', 'NumberOfTime60-89DaysPastDueNotWorse', 'NumberOfTimes90DaysLate', and 'RevolvingUtilizationOfUnsecuredLines'. To be able to do that, it was preformed specific threshold values for each variable based on boxplots (Fig.03-Fig.12). Any data points exceeding these thresholds were considered outliers and subsequently removed from the dataset. The threshold values were determined based on analysing in detail the variables and their boxplots.

## **3.4. Scaling Features**

The code applies feature scaling to a subset of numerical features using the StandardScaler from scikit-learn. The selected numerical features are: RevolvingUtilizationOfUnsecuredLines; age; NumberOfTime30-59DaysPastDueNotWorse; DebtRatio; MonthlyIncome; NumberOfOpenCreditLinesAndLoans; NumberOfTimes90DaysLate; NumberRealEstateLoansOrLines; NumberOfTime60-89DaysPastDueNotWorse

The fit\_transform() method is used to fit the scaler on the selected features and transform their values to have zero mean and unit variance. The scaled values are then updated in the original DataFrame.

## **3.5. Data Drift Test**

In order to access the presence of data drift we conducted a thorough analysis to identify any changes in data distribution between a reference dataset (data\_reference) and an analysis dataset (data\_analysis) set. The data drift analysis involved several important phases:

Defining Thresholds for Data Drift Test: To quantify and measure data drift, we established threshold values using the ConstantThreshold class from the nannyml library. The lower threshold was set at 0.3, and the upper threshold was set at 0.7. These thresholds serve as benchmarks for detecting potential data drift during the analysis.

Calculating Univariate Data Drift: To evaluate the univariate data drift, we utilized the UnivariateDriftCalculator object from the nannyml library. This calculator takes into account column names, categorical columns, chunk size, and the defined thresholds. First, we fitted the drift calculator on the reference data using the fit () method. Then, we applied the calculate () method to the analysis data, which provided us with a comprehensive DataFrame detailing the data drift results.

Generating Data Drift Report: To facilitate visual representation and interpretation of the data drift analysis, we employed the Report class from the evidently library. By leveraging the DataDriftPreset metric present, with a categorical statistical test set to Kolmogorov-Smirnov test (KS) and a threshold of 0.05, we generated a detailed data drift report specifically for the numerical columns.

The resulting data drift report was saved as an HTML file and presented in Fig. 22 in the appendix. Based on the analysis and examination of the data distributions, we have concluded that there is no drift observed in any of the variables. This indicates that the data used for the analysis remains consistent and stable over time.

## **3.6.Feature selection**

It was conducted an analysis of the correlation between variables through a correlation matrix (Fig.23), concluding that the variables exhibit a low degree of correlation among each other. Consequently, based solely on the information provided by the correlation matrix we decide to do not eliminate or transform any of the variables.

For feature selection, we utilized the SelectKBest method, which leverages mutual information to evaluate the relevance of features. Through this process, we identified the top eight most significant features.

Prior to the modelling phase, it is essential to split the dataset into training and test sets using the 'train\_test\_data\_split' function. This critical step allows for model performance evaluation and effective hyperparameter tuning.

# **4.Modelling and results**

During this phase, it was evaluated the performance of two models, Random Forest and LightGBM, both with and without feature selection and Scaling. The selection of Random Forest and LightGBM models was driven by the unbalanced nature of our dataset, as both models have proven effectiveness in handling such data, making them well-suited for our analysis.

To facilitate the modelling process, we have implemented a function called "model\_train." This function encompasses several important steps, including testing the two models with and without feature selection, performing cross-validation, tuning the models, selecting the best model, and finally providing the Shap graphic for interpretation.

To fine-tune our models and discover the optimal hyperparameter combinations, it was employed a Grid Search with cross-validation. For the Random Forest model, we evaluated the number of estimators (50, 100, and 200) and the maximum depth of each tree (3, 5, and 7). In the case of the LightGBM model, in addition to the number of estimators and maximum depth, we also explored the impact of learning rate on model performance. We considered learning rate options of 0.01, 0.1, and 0.3, which control the speed at which the model learns from each boosting iteration.

The best prediction model found is with LightGBM algorithm with Feature Selection and Scaling. With the following parameters and results:

* Number of estimators: 100
* Learning rate: 0.1
* Max Depth: 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Model | 0 | 1 | Macro average | Weighted average |
| Accuracy | 0.94 |  |  |  |  |
| F1\_score | 0.92 | 0.97 | 0.27 | 0.62 | 0.92 |
| Precision |  | 0.94 | 0.58 | 0.76 | 0.92 |
| Recall |  | 0.99 | 0.18 | 0.59 | 0.94 |
| Support |  | 27972 | 1973 | 29945 | 29945 |

Fig.05- Evaluation Metrics (values rounded with 2 decimal places)

The LightGBM classifier was tested using different variations, including feature selection and scaling. The results obtained from these models were very similar.

The classification model achieved an accuracy value indicating that it correctly classified approximately 93.73% of the instances in the dataset. The F1 score, which combines precision and recall, was measured at 0.92. This score considers both false positives and false negatives, providing an overall evaluation of the model's performance.

For Class 0, the model demonstrated a high precision of 0.94 and the recall value for suggest a high proportion of true positives correctly identified out of all actual positive instances. Also, the F1-Score (0.97) demonstrates a good balance between precision and recall. However, for Class 1, the precision was lower at 0.58 and the recall 0.18 indicating a higher number of false positives for this class and that the model had difficulty correctly identifying positive instances for this class. The F1-score (0.27) also confirms relatively low performance in terms of precision and recall.

In addition, for a comprehensive understanding of our best model, our team generated a SHAP graphic (Fig. 24) revealing the importance values of all features, from which we determined that the variable NumberOfTime90DaysPastDueNotWorse holds the highest significance in our model. Therefore, we analyse the cross-validation score, and train score thought the Figure 25 to despite overfitting.

# **5.Conclusions**

In today's market economies, banks serve as essential intermediaries in the financial system, wielding significant influence over investment decisions and determining access to credit. To facilitate informed lending decisions, credit scoring algorithms have become indispensable tools for banks to assess the creditworthiness of individuals and businesses seeking loans.

This report highlights a project focused on enhancing credit scoring models through the application of machine learning techniques. By simulating real-world processes and employing advanced models like LightGBM, the project aims to provide more accurate risk assessments for banks.

The report delves into various stages of the project, starting with data preprocessing, where techniques like imputation and mode filling are employed to handle missing values and outliers are removed. The dataset, consisting of 11 variables, is carefully explored to uncover valuable insights relevant to credit scoring.

Moreover, the report addresses specific challenges encountered during the project, such as imbalanced data and the need for feature selection. To tackle these challenges, the team employs techniques like resampling methods for imbalanced data and feature selection algorithms to identify the most relevant variables for the credit scoring task.

In the subsequent stages, the project focuses on model training, deployment, and monitoring. LightGBM, a powerful gradient boosting framework, is selected as the optimal model due to its superior performance. The chosen model undergoes feature scaling and selection, leading to an impressive accuracy rate of approximately 93.73%.

In our dataset, we face class imbalance, with the minority class (1) accounting for only 6% of the data. This imbalance negatively affects model performance, particularly in terms of recall. The model exhibits a tendency to prioritize predicting the majority class, resulting in a low recall value for the minority class (classification=1).

To ensure the sustainability and adaptability of the credit scoring model, the report also addresses the concept of data drift and suggests potential approaches to monitor and mitigate its impact, the dataset does not exhibit any data drift, which is beneficial as it enables us to proceed with our analysis and modelling without concerns regarding the stability of the data.

Overall, this project showcases the potential of machine learning in revolutionizing credit scoring, enabling banks to make more precise and informed lending decisions while minimizing risks.

# **6.Further Suggestions and Limitations**

To address the issue of unbalanced data, several techniques can be employed. One approach is resampling, where the dataset can be balanced by either oversampling the minority class or under sampling the majority class.

Another technique is data augmentation, which involves creating synthetic samples for the minority class by applying transformations or introducing random variations to existing instances. This increases the diversity of the minority class and provides more training data for the model to learn from. Albeit this method seems plausible, due to the unbalanced data, the synthetic samples created might be biased.

Having that in mind, the most obvious and probably most adequate solution is to manually insert new real data to balance the data.

Additionally, expanding the range of models used in the analysis can provide valuable insights. Including a diverse set of models, such as ensemble methods, support vector machines, or gradient boosting algorithms, can help capture different patterns and improve the overall performance and robustness of the modelling process.

# **7.Appendix**



*Figure 01- Kedro visualization*

*Uma imagem com quadrado, Retângulo, diagrama, píxel

Descrição gerada automaticamente*

*Figure 02- Age Variable Histogram*

*Uma imagem com Retângulo, captura de ecrã, quadrado, design

Descrição gerada automaticamente*

*Figure 03- DebtRatio Variable Histogram*

*Uma imagem com Retângulo, captura de ecrã, quadrado, design

Descrição gerada automaticamente*

*Figure 04- MontlyIncome Variable Histogram*

*Uma imagem com Retângulo, captura de ecrã, quadrado, file

Descrição gerada automaticamente*

*Figure 05- NumberOfDependents Variable Histogram*

*Uma imagem com Retângulo, quadrado, captura de ecrã, diagrama

Descrição gerada automaticamente*

*Figure 06- NumberOfOpenCreditLinesAndLoans Variable Histogram*

*Uma imagem com Retângulo, captura de ecrã, quadrado, design

Descrição gerada automaticamente*

*Figure 06- NumberOfTime30-59DaysPastDueNotWorse Variable Histogram*

*Uma imagem com Retângulo, captura de ecrã, quadrado, design

Descrição gerada automaticamente*

*Figure 07- NumberOfTime60-89DaysPastDueNotWorse Variable Histogram*

*Uma imagem com Retângulo, captura de ecrã, quadrado, design

Descrição gerada automaticamente*

*Figure 08- NumberOfTime90DaysPastDueNotWorse Variable Histogram*

*Uma imagem com Retângulo, captura de ecrã, quadrado, moldura de fotografia

Descrição gerada automaticamente*

*Figure 09- NumberRealStateLoanssOrLines Variable Histogram*

*Uma imagem com Retângulo, captura de ecrã, quadrado, design

Descrição gerada automaticamente*

*Figure 10- RevolvingUtilizationOfUnsecuredLines*

*Uma imagem com captura de ecrã, Retângulo, quadrado, design

Descrição gerada automaticamente*

*Figure 11- SeriousDlqin2yrs Variable Histogram*

*Uma imagem com diagrama, captura de ecrã, Retângulo, texto

Descrição gerada automaticamente*

*Figure 12- Boxplot of the variable Age*

*Uma imagem com texto, captura de ecrã, Retângulo, file

Descrição gerada automaticamente*

*Figure 13- Boxplot of the variable Debt Ratio*

*Uma imagem com texto, captura de ecrã, Retângulo, diagrama

Descrição gerada automaticamente*

*Figure 14- Boxplot of the variable MontlyIncome*

*Uma imagem com texto, captura de ecrã, file, diagrama

Descrição gerada automaticamente*

*Figure 15- Boxplot of the Variable NumberOfDependents*

*Uma imagem com texto, captura de ecrã, file, diagrama

Descrição gerada automaticamente*

*Figure 16- Boxplot of the variable NumberOfopenCreditLinesAndLoans*

*Uma imagem com texto, captura de ecrã, Retângulo, file

Descrição gerada automaticamente*

*Figure 17- Boxplot of the variable NumberOfTime30-59DaysPastDueNotWorse*

*Uma imagem com texto, captura de ecrã, Retângulo, file

Descrição gerada automaticamente*

*Figure 18- Boxplot of the variable NumberOfTime60-89DaysPastDueNotWorse*

*Uma imagem com texto, captura de ecrã, Retângulo, diagrama

Descrição gerada automaticamente*

*Figure 19- Boxplot of the variable NumberOfTime90DaysPastDueNotWorse*

*Uma imagem com texto, captura de ecrã, file, diagrama

Descrição gerada automaticamente*

*Figure 20- Boxplot of the variable NumberRealLoansOrLines*

*Uma imagem com texto, captura de ecrã, Tipo de letra, Retângulo

Descrição gerada automaticamente*

*Figure 21- Boxplot of the variable RevolvingUtilizationOfUnsecuredLines*

*Uma imagem com texto, número, captura de ecrã, file

Descrição gerada automaticamenteFigure 22- Data Drift Results*

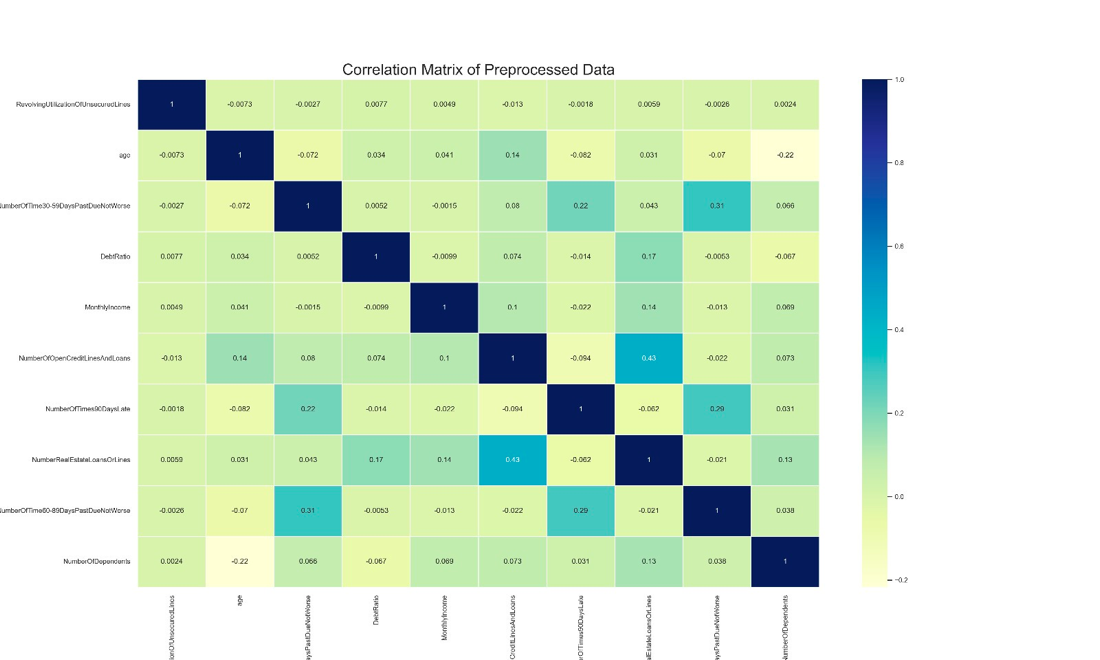


Figure 23- Correlation matrix

Uma imagem com texto, file, Tipo de letra, diagrama

Descrição gerada automaticamente

*Figure 24- SHAP values feature importance*

Uma imagem com captura de ecrã, file, Gráfico, texto

Descrição gerada automaticamente

*Figure 25-Analyse overfitting on the best model.*