PRAGUE UNIVERSITY OF ECONOMICS AND BUSINESS

FACULTY OF ACCOUNTING AND FINANCE

Department of Banking and Insurance

Application of Machine Learning Algorithms within Credit Risk Modelling

Master's thesis

Author: Bc. Petr Nguyen

Study program: Banking and Insurance | Data Engineering

Supervisor: prof. PhDr. Petr Teplý, Ph.D.

Year of defense: 2023

Declaration of Authorship

I, as an author, hereby declare that I wrote and compiled the Master's thesis *"Application of Machine Learning Algorithms within Credit Risk Modelling"* independently, using only the resources and literature listed in bibliography.

May 25, 2023, Prague

Petr Nguyen

Abstract

The abstract should concisely summarize the contents of a thesis. Since potential readers should be able to make their decision on the personal relevance based on the abstract, the abstract should clearly tell the reader what information he can expect to find in the thesis. The most essential issue is the problem statement and the actual contribution of described work. The authors should always keep in mind that the abstract is the most frequently read part of a thesis. It should contain at least 70 and at most 120 words (200 when you are writing a thesis). Do not cite anyone in the abstract.

Keywords: machine learning, data science, credit risk, probability of default, loans, mortgages

Abstrakt

Nutnou součástí práce je anotace, která shrnuje význam práce a výsledky v ní dosažené. Anotace práce by neměla být delší než 200 slov a píše se v jazyce práce (tj. česky, slovensky či anglicky) a v překladu (tj. u anglicky psané práce česky či slovensky, u česky či slovensky psané práce anglicky). Anotace práce by neměla být delší než 200 slov a píše se v jazyce práce (tj. česky, slovensky či anglicky) a v překladu (tj. u anglicky psané práce česky či slovensky, u česky či slovensky psané práce anglicky). V abstraktu by se nemělo citovat.

Klíčová slova: machine learning, data science, kreditní riziko, pravděpodobnost defaultu, úvery, hypotéky

Acknowledgments

I, as an author, would like to express my deepest gratitudes and thanks to my supervisor prof. PhDr. Petr Teplý, Ph.D.for his help and significant advices throughout my thesis. Last but not least, I would like to also thank to my family for an enormous support during my studies.

Contents

List of Tables

List of Figures

Acronyms

ML Machine Learning

PD Probability of Default

AUC Area Under the Curve

LR Logistic Regression

RF Random Forest

GB Gradient Boosting

MLP Multi-Layer Perceptron

DT Decision Tree

Chapter 1

Introduction

TBD

This document serves two purposes. First, it is a template and example for a master's thesis. Second, the text in all sections contains some useful information on structuring and writing your thesis.

The introduction should consist of three parts (as paragraphs, not to be structured into multiple headings): The first part deals with the background of the work and describes the field of research. It should also elaborate on the general problem statement and the relevance. The second part should describe the focus of the thesis, typically the paragraph starts with a phrase like "The objective of this thesis is ...." The last part should describe the structure of the thesis, for instance in the following manner. The thesis is structured as follows: Chapter 2 cites some formal requirements of the faculty and the frequently asked questions about the template, Chapter 3 gives some hints on basic formatting features and covers also acronyms, figures, boxes and tables. Chapter 4 gives a recommendation on the usage of hyphens in English language in LaTeX and explains how to use the itemize and quote environments and shows a few enumerate-based environments. Chapter 5 presents a checklist of common mistakes to avoid. **??** contains numerous hints. Chapter 6 summarizes our findings.

Chapter 2

Credit Risk Modelling

TBD

2.1 Formal requirements of master's thesis

According to Dean's Provision no. 18/2017:

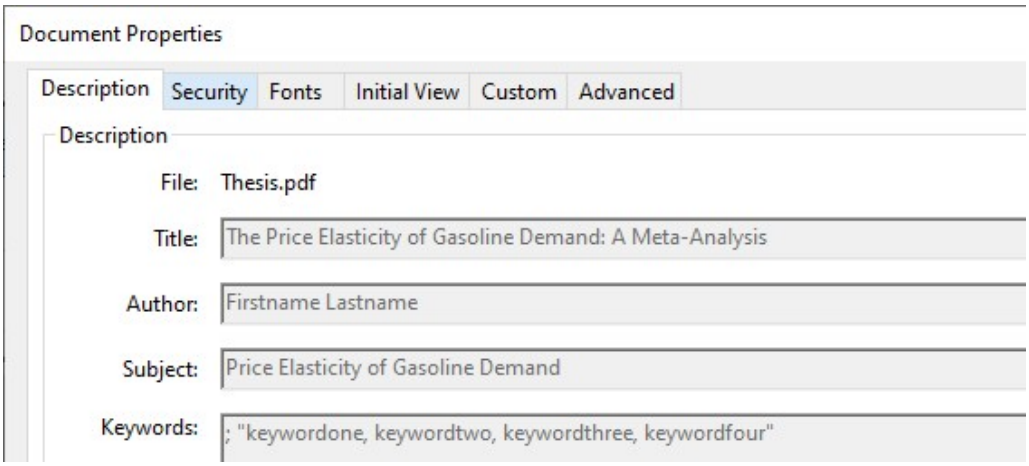
- The minimum extent of master's thesis is 60 standard pages (108 thousand characters including spaces) of the text itself, i.e. without an abstract and appendices and a list of literature. In case the master's thesis is written in English, its minimum extent is 50 standard pages (90 thousand characters including spaces) without an abstract and appendices and a list of literature. When writing a standard text document, the minimum requirement is 60 characters per line and 30 lines per page, i.e. 1,800 characters per page (the so-called standard page). Font size, page layout, margins, and line spacing need to be customized.
- Generally, a standard form of the page of the final thesis applies the fonts of 12 points, the gaps between the paragraphs are recommended to be of the size of 6 points. Notes and footnotes can be written in a 10-point font. The text is aligned on both sides (aligned to a block). Electronic version of the thesis will be entered by a student/applicant for a state examination through the SIS website interface in the archive format of PDF/A version 1.3 or higher. Further details are stipulated by the rector's provision.

- The master's thesis is submitted in the accreditation language of the respective follow-up Master's study program.

Note that due to GDPR, the thesis cannot include any personal information (phone, e-mail) or signatures (neither of the author nor of the supervisor).

2.2 Template adjustments and meta-data

Read README.txt to get a jist of how the template works and how to adjust styles. You can change the properties of your pdf file, such as title, author, keywords, or publisher



in **Thesis.xmpdata** file. The file is editable in any text editor.

Chapter 3

Machine Learning

TBD

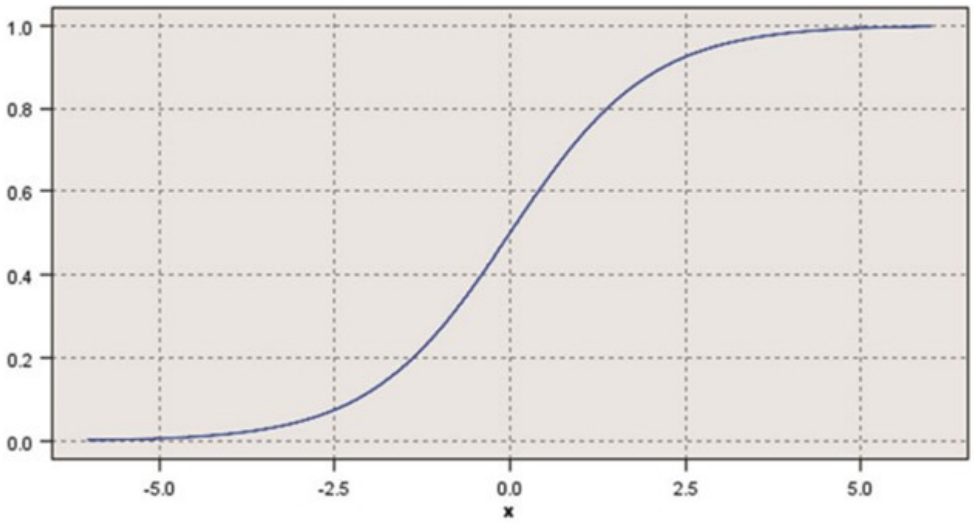
3.1 Terminology

3.2 Algorithms

3.2.1 Logistic Regression

Despite the algorithm's name, it is actually not a regression but rather a classification model. In contrast, a linear regression's target variable is continuous whereas regarding a logistic regression, the target variable is binary or dichotomous. For the probability estimation it is using a logistic, or so-called sigmoid function, which maps any real value within the range of 0 to 1 and takes a S-shaped curve as can be seen in Figure 3.1.

Figure 3.1: Logistic function



Source: (Wendler & Gröttrup 2021)

The linear form of the logistic regression with n features can be written as:

$$\ln\left(\frac{P}{1-P}\right) = \beta_0 + \sum_{i=1}^{n} \beta_i X_i \tag{3.1}$$

where P is the probability of the occurred event, conditional on the set of given features. Let us denote $Y = 1$ as an observed target instance where the event occurred (e.g., defaulted), then:

$$P = \Pr(Y = 1 \mid X_1, X_2, \ldots, X_n) \tag{3.2}$$

Therefore, the term within the natural logarithm are the odds or more particularly, the ratio of the probability of the event with respect to the probability of non-event, both conditional on the same set of given features.

$$\begin{aligned} \frac{P}{1-P} &= \frac{\Pr(Y = 1 \mid X_1, X_2, \ldots, X_n)}{1 - \Pr(Y = 1 \mid X_1, X_2, \ldots, X_n)} \\ &= \frac{\Pr(Y = 1 \mid X_1, X_2, \ldots, X_n)}{\Pr(Y = 0 \mid X_1, X_2, \ldots, X_n)} \end{aligned} \tag{3.3}$$

Referring to the previous equations, solving for P, henceforth we get a final equation for computing the probability of occurred event with usage of logistic regression:

$$P = \frac{1}{1 + e^{-\left(\beta_0 + \sum_{i=1}^{n} \beta_i X_i\right)}} \tag{3.4}$$

3.2.2 Decision Tree

3.2.3 Naive Bayes

Naive Bayes is a classification and probabilistic machine learning algorithm which is based on the Bayes theorem:

$$\Pr(C = c \mid E) = \frac{\Pr(C = c) \times \Pr(E \mid C = c)}{\Pr(E)} \tag{3.5}$$

where:

- $\Pr(C = c \mid E)$ is the posterior probability which is the probability that the target variable C takes on the class of interest c after taking the evidence E.
- $\Pr(C = c)$ is the prior probability of the class c is the probability we would assign to the class c before seeing any evidence E.
- $\Pr(E \mid C = c)$ is the probability of seeing the evidence E conditional on the given class c.
- $\Pr(E)$ is the probability of the evidence E.

With regards to the binary classification, we can substitute Y as a target variable instead C, and set of features X which will refer to the set of evidence E. Assuming that $Y = 1$ refers to the occurrence of given event (e.g., default), henceforth the probability of default using the NaĂŻve Bayes algorithm can be mathematically expressed as:

$$\Pr(Y = 1 \mid X) = \frac{\Pr(Y = 1) \times \Pr(X \mid Y = 1)}{\Pr(X)} \tag{3.6}$$

One of the assumptions of this algorithm is the conditional probabilistic independence among the features. Therefore, instead of computing the probability of all features together, conditional on the class event, for each feature X we will compute its probability, conditional on the class event. Hence:

$$\Pr(Y = 1 \mid X) = \prod_{i=1}^{n} \Pr(X_i \mid Y = 1) \tag{3.7}$$

With regards to the conditional independence, we can also derived the probability of evidence E or set of features X respectively, as a sum of the probability of given set of features, conditional on class one (event), and of the probability of given set of features, conditional on one class two (non-event). Therefore:

$$\Pr(X) = \Pr(X \mid Y = 1) \times \Pr(Y = 1) + \Pr(X \mid Y = 0) \times \Pr(Y = 0) \tag{3.8}$$

Therefore:

$$\begin{aligned} \Pr(X) = & \prod_{i=1}^{n} \Pr(X_i \mid Y = 1) \times \Pr(Y = 1) + \\ & \prod_{i=1}^{n} \Pr(X_i \mid Y = 0) \times \Pr(Y = 0) \end{aligned} \tag{3.9}$$

Finally, we can derive the final formula for NaĂŻve Bayes the posterior probability as:

$$\frac{\prod_{i=1}^{n} \Pr(X_i \mid Y = 1)}{\prod_{i=1}^{n} \Pr(X_i \mid Y = 1) \Pr(Y = 1) + \prod_{i=1}^{n} \Pr(X_i \mid Y = 0) \Pr(Y = 0)} \tag{3.10}$$

3.2.4 K-Nearest Neighbors

3.2.5 Random Forest

3.2.6 Gradient Boosting

3.2.7 Support Vector Machine

3.2.8 Neural Networks

3.3 Evaluation Metrics

This section focusdx on particular measures through which it is possible to determine to a predictive power of model in terms of its performance. The are

many ways, how to evaluate the model's performance, therefore, only the most common ones will be further described. Note, since default prediction regards classification model, therefore regression's evaluation metrics will be omitted.

3.3.1 Confusion Matrix

Confusion matrix is a table which summarizes the classification model's performance with respect to the actual classes and predicted classes. It is a square $n \times n$ matrix, where n determines number of classes within the target variable. Let us denote the confusion matrix as $C(f)$ for classification algorithm f. Its elements can be denoted as $c_{i,j}$ where i and j refer to the row and column indices, respectively, or more particularly, i refers to the actual class and j to the class predicted by the classifier f. Each element of the confusion matrix refers to the number of instances corresponding to actual class i and predicted class j. For instance, the element $c_2, 1$ would refer to the number of instances which have the actual class 2 but have been classified as class 1. Mathematically, the confusion matrix can be written as following:

$$C = c_{i,j} = \sum_{l=1}^{m} [(y_l = i) \wedge (f(x_l) = j)] \tag{3.11}$$

Or either in matrix form as:

$$C_{i \times j} = \begin{bmatrix} c_{1,1} & c_{2,1} & \cdots & c_{1,j} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ c_{i,1} & c_{i,2} & \cdots & c_{i,j} \end{bmatrix} \tag{3.12}$$

From the given matrix, the diagonal elements represent the numbers of correctly classified instances, whereas the non-diagonal elements represent the numbers of misclassified instances. Further, let us consider a binary classification - hence, the confusion matrix will have a form of 2×2.

$$C_{2 \times 2} = \begin{bmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{bmatrix} \tag{3.13}$$

We can this rewrite confusion matrix as:

$$C_{2\times 2} = \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \tag{3.14}$$

where:

- TP is the True Positive which refers to the number of instances which correspond to the actual class $True$ and indeed have been correctly classified as class $True$.
- FP is the False Positive which refers to the number of instances which correspond to the actual class $True$, but have been incorrectly classified as class $False$. In the statistics and hypothesis–testing terms, it can be also called as Type 1 Error.
- FN is the False Negative which refers to the number of instances which correspond to the actual class $False$, but have been incorrectly classified as class $True$. In the statistics and hypothesis–testing terms, it can be also called as Type 2 Error.
- TN is the True Negative which refers to the number of instances which correspond to the actual class $False$ and indeed have been correctly classified as class $False$.

3.3.2 Accuracy

$$Accuracy = \frac{TP + FN}{TP + TN + FP + FN} \tag{3.15}$$

3.3.3 Recall

$$Precision = \frac{TP}{TP + FN} \tag{3.16}$$

3.3.4 Precision

$$Precision = \frac{TP}{TP + FP} \tag{3.17}$$

3.3.5 F1 Score

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{3.18}$$

3.3.6 AUC

In order to derive Area Under the Curve (AUC), first we need to define Receiver Operating Characteristics (ROC) curve. ROC curve is two-dimensional visualization of the model performance as a probability curve in terms of True Positive Rate (TPR) and False Positive Rate (FPR) based on varying the given threshold.

Briefly, it can be construct as following: First, we need to sort the instances by the predicted probability and based on the given probability, we set a threshold - what will be above the threshold will be classified as $True$ instance and what is below the threshold will be classified as $False$ instance. Based on these classified instances, the confusion matrix can be constructed and via which we can compute the TPR and FPR values. Thus, if the probability is 1, the threshold will be 1 as well and hence:

- TPR will be 0 because there is no probability which is higher than 1 and hence, everything will be classified as $False$ which will result into TP of 0, and subsequently into TPR of 0 as well.
- FPR will be 0, too â€“ since everything will be classified as $False$, therefore FP will be 0 which implies FPR to be 0, too.

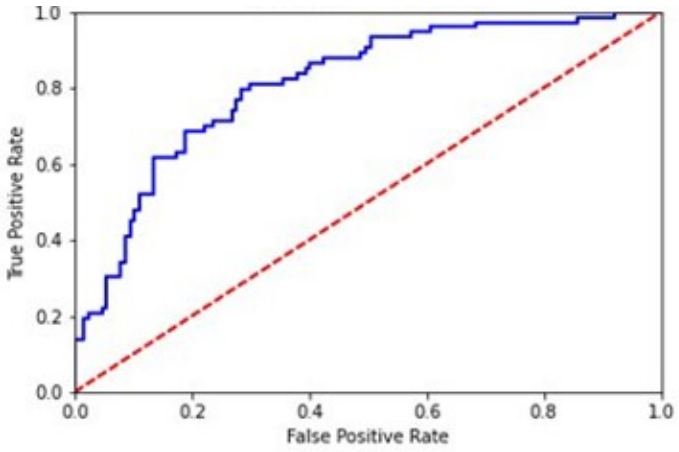
On the other hand, if the probability is 0, the threshold will be 0 as well and hence:

- TPR will be 1 because there is no probability which is lower than 0 and hence, everything will be classified as $True$ which will result into FN of 0, and subsequently into TPR of 1.
- FPR will be 1, too â€“ since everything will be classified as $True$, therefore TN will be 0 which implies FPR to be 1.

Thus, based on each threshold, the TPR and FPR will be to coordinates for single point within the graph and based on such points, we can construct the

ROC curve. Such visualization on the following Figure 3.2. Note the diagonal line represents a random model which randomly and correctly predicts the $True$ and $False$ classes in such way, that FPR and TPR are the same. Logically, a decent model should perform better than the random model, thus it the ROC curve should be above the diagonal line. Intuitively, the best possible theoretical model would have TPR of 1 and FPR of 0, meaning that all the $True$ actual classes should be predicted as $True$ and all the $False$ actual classes should not be classified as $True$ Within the ROC curve, the given curve reaches the left top corner which corresponds to the coordinates of TPR and FPR.

Figure 3.2: ROC Curve



Source: Author's results in Python.

AUC is basically the representation of ROC curve as a single number as it aggregates the performance on all possible thresholds. AUC can be interpreted as the probability that the randomly chosen actual $True$ instance is ranked higher than the randomly chosen actual $False$ instance. Since ROC curve is a probability curve, thus it is considering distribution curve of TP and distribution class of TN, separated by particular threshold â€“ hence, TP would have probability scores above the given thresholds, whereas TN would have probability scores below the threshold. If these curves do not overlap, meaning the model can perfectly distinguish between the $True$ and $False$ values, therefore the AUC would be 1 and the ROC curve would reach the left top corner. However, this idealistic situation does not occur in the practice at all, but rather the two distributions are overlapping since the misclassification of the classes takes the place. The bigger overlap, the lower AUC is. If the distributions are completely overlapping, it implies the AUC of 0.5, meaning that the model cannot distinguish between the $True$ and $False$ classes, which is

the worst scenario. On the other hand, if the distributions are totally opposite (meaning that the TP instances would have probability scores below the given threshold, whereas the TN instances would have probability scores above the given threshold), the AUC would be 0 since the model is predicting the $True$ actual classes instead of $False$ and vice versa.

As the AUC is an area present underneath the ROC curve, mathematically, it can be computed with the definite integral where x is the given threshold:

$$AUC = \int_0^1 TPR\left(FPR^{-1}(x)\right) dx \tag{3.19}$$

3.3.7 Kolmogorov-Smirnov Distance

$$KS = \max_{x \in \mathbb{R}} \left| \hat{F}_n(x) - F(x) \right| \tag{3.20}$$

3.3.8 Somer's D

$$SD = \frac{\tau(X,Y)}{\sqrt{\tau(X,X)\,\tau(Y,Y)}} \tag{3.21}$$

3.3.9 Matthews Correlation Coefficient

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{3.22}$$

3.3.10 Brier Score Loss

$$BSL = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.23}$$

3.3.11 Jaccard Score

$$JC = \frac{|y \cap \hat{y}|}{|y \cup \hat{y}|} \tag{3.24}$$

3.3.12 Zero-One Loss

$$ZOL = \frac{1}{n}\sum_{i}^{n} \delta_{y_{i=1} \neq \hat{y}_i} \text{ , where } \delta_{y_i \neq \hat{y}_i} = \begin{cases} 1, & \text{if } x < 0. \\ 0, & \text{otherwise.} \end{cases} \tag{3.25}$$

3.4 Hyperparameter Tuning

TBD

3.4.1 Grid Search

3.4.2 Random Search

3.4.3 Bayesian Optimization

3.5 Imbalanced Class Distribution

TBD

3.5.1 Random Oversampling

3.5.2 SMOTE Oversampling

3.5.3 ADASYN Oversampling

TBD

3.6 Optimal Binning

3.6.1 Weight of Evidence Encoding

TBD

Chapter 4

Application of Machine Learning Algorithms

4.1 Repository and Environment Structure

Figure 4.1: Repository Structure

```
|--- data
|  |--- interim_dat.csv
|  |--- preprocessed_dat.csv
|  |--- raw_data.csv
|
|--- flask_app
|  |--- inputs
|  |  |--- inputs_flask_app_dict.pkl
|  |
|  |--- templates
|  |  |--- index.html
|  |  |--- results.html
|  |
|  |--- app.py
|
|--- models
|  |--- feature_preprocessing
|  |--- feature_selection
|  |--- model_selection
|  |--- objects_FINAL
|
|--- plots
|--- Master_Thesis.ipynb
|--- README.md
|--- requirements.yml
```

Source: Author's results in Python.

4.2 Data Exploration

This section is focused on exploration of the analyzed loan dataset, particularly on dataset description, distribution analysis and association analysis, in order to infer potential valuable insights and hypotheses which can be used in the preprocessing or modelling part.

4.2.1 Dataset Description

The analyzed dataset pertains to the HMEQ dataset which contains loan application information and default status of 5,960 US home equity loans. Such dataset was acquired from Credit Risk Analytics.

As can be seen in Table 4.1, the dataset contains 12 columns, 11 features and 1 target variable `BAD` indicating whether the loan was in default (`1`) or not (`0`). Amongst the 11 features, there are 9 numeric features and 2 categorical features, namely `REASON` which contains 2 categories - Debt consolidation (`DebtCon`) and Home improvement (`HomeImp`), and `JOB` which fontains following categories - Administration (`Offce`), Sales, Manager (`Mgr`), Professional Executive (`ProfExe`), Self-employed (`Self`), and Other.

Table 4.1: Dataset columns

Columns	Description	Data type
BAD	Default status	Boolean
LOAN	Requested loan amount	numeric
MORTDUE	Loan amount due on existing mortgage	numeric
VALUE	Value of current underlying collateral property	numeric
REASON	Reason of loan application	categorical
JOB	Job occupancy category	categorical
YOJ	Years of employment at present job	numeric
DEROG	Number of derogatory public reports	numeric
DELINQ	Number of delinquent credit lines	numeric
CLAGE	Age of the oldest credit line in months	numeric
NINQ	Number of recent credit inquiries	numeric
CLNO	Number of credit lines	numeric
DEBTINC	Debt-to-income ratio	numeric

Source: `http://www.creditriskanalytics.net/datasets-private2.html`

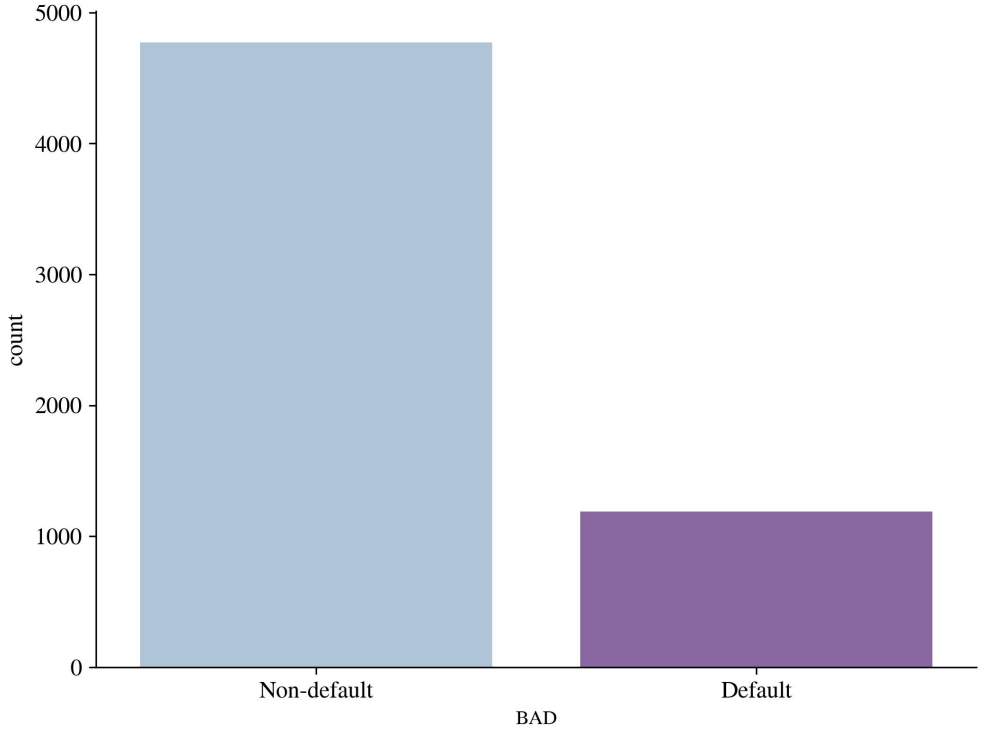
4.2.2 Distribution Analysis

In this subsection, we inspect the distribution of our variables, including the target variable and the features. Such distribution inspection may help us to identify potential outliers, missing values, and other potential issues with the dataset.

Default Distribution

Regarding the the target variable distribution, from the Figure 4.2 we can observe that the default status distribution is heavily imbalanced, as most of the loans have not defaulted yet. Particularly, 80.05% of the observations have been labelled as non-default (4,771 observations) and 19.95% observations labelled as default (1,189 observations). This may cause problems in the modelling part, as the model may be biased towards the majority class, i.e., the non-default class. Such imbalanced class issue will be further treated in Subsection 4.3.1.

Figure 4.2: Default status distribution



Source: Author's results in Python.

Numeric Features' Distribution

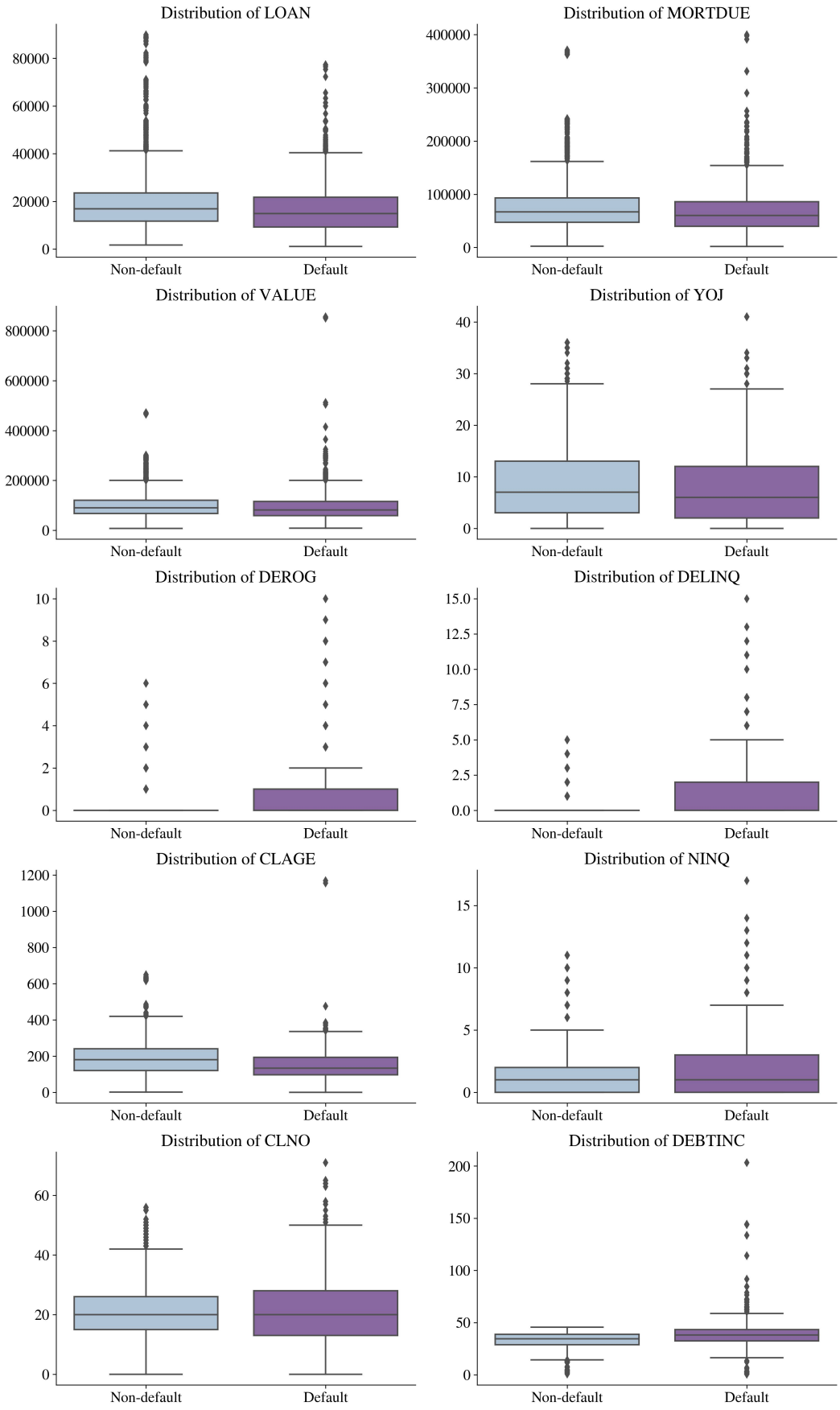
With regards to the numeric features, most of the features are positively skewed containing outliers as can be seen in Figure 4.3, which depicts conditional distribution of numeric features on the default status visualized using boxplots.

All the outliers appear to be valid, i.e., they have not occurred due to data entry errors. This can be attributed to the fact that all the numeric features are non–negative, thus it is not possible to have negative number of years at present job, negative number of delinquent credit lines, etc. Furthermore, the maximum values of given features are not unrealistically high, which further confirms the validity of the outliers. Nevertheless, the outliers need to be treated since they can biased model's weights or coefficients (in case of logistic regression or neural networks), jeopardize distance calculation (in case of KNN) or in general, they can affect the position and orientation of the decision boundary – all the reasonings can lead to the model's overfitting and inaccurate and biased

predictions. The outliers' treatment is further described in Subsection 4.3.2.

With respect to the target variable, we can observe some differences in the distribution shapes of `DEROG` and `DELINQ`, which are less skewed and have lower dispersion for non–default cases compared to the default cases. Since both features indicates a negative information about the delinquency, it is expected that the higher value is, the more likely the loan will end in default. Referring to the feature `DEBTINC`, it does not have any extreme values for non–default cases, but it has some extreme values for default cases. We can infer that if the debt–to–income ratio is too high, hence applicant's income is not sufficient to cover his debt, the loan is more likely to end in default. The assocation between the default status and the numeric features is further investigated in Section 4.2.3.

Figure 4.3: Conditional distribution of numeric features



Source: Author's results in Python.

Due to the fact that the boxplots do not capture the missing values occurred in given features, it is also important to inspect the numbers and proportions of missing values in each feature, conditional on the default status. As can be seen in Table 4.2, n_0 refers to the number of missing values in given feature for non-default cases, n_1 refers to the number of missing values in given feature for default cases. N_0 and N_1 refer to the total number non–default cases and default cases respectively, therefore n_0/N_0 refers to the proportion of missing values in given feature for non-default cases, and n_1/N_1 refers to the proportion of missing values in given feature for default cases.

Pertaining to the feature `DEBTINC`, we can observe a significant difference in the number of missing values between the default and non-default cases. Out of all defaulted loans, 66.11 % had missing debt–to–income ratio, whereas only 10.08 % out of all non-defaulted loans had missing debt–to–income ratio. Therefore, there could be a strong association between the missing debt–to–income ratio and the default.

Similarly, the table depicts a significant difference with respect to `VALUE` as 0.15 % had missing collateral property value out of all non–defaulted loan, and 8.92 % defaulted loans had missing collateral property value. It can be inferred that loan applicants who withhold information on their collateral property value or debt-to-income ratio are more likely to default on their loans. This may be due to negative information that they are trying to conceal, such as an excessively high debt or low income, or a low collateral property value. Such associations are further investigated in Section 4.2.3.

Table 4.2: Numeric features NA's table

Feature	n_0	n_1	n_0/N_0	n_1/N_1
LOAN	0	0	0 %	0 %
MORTDUE	412	106	8.64 %	8.92 %
VALUE	7	105	0.15 %	8.83 %
YOJ	450	65	9.43 %	5.47 %
DEROG	621	87	13.02 %	7.32 %
DELINQ	508	72	10.65 %	6.06 %
CLAGE	230	78	4.82 %	6.56 %
NINQ	435	75	9.12 %	6.31 %
CLNO	169	53	3.54 %	4.46 %
DEBTINC	481	786	10.08 %	66.11 %

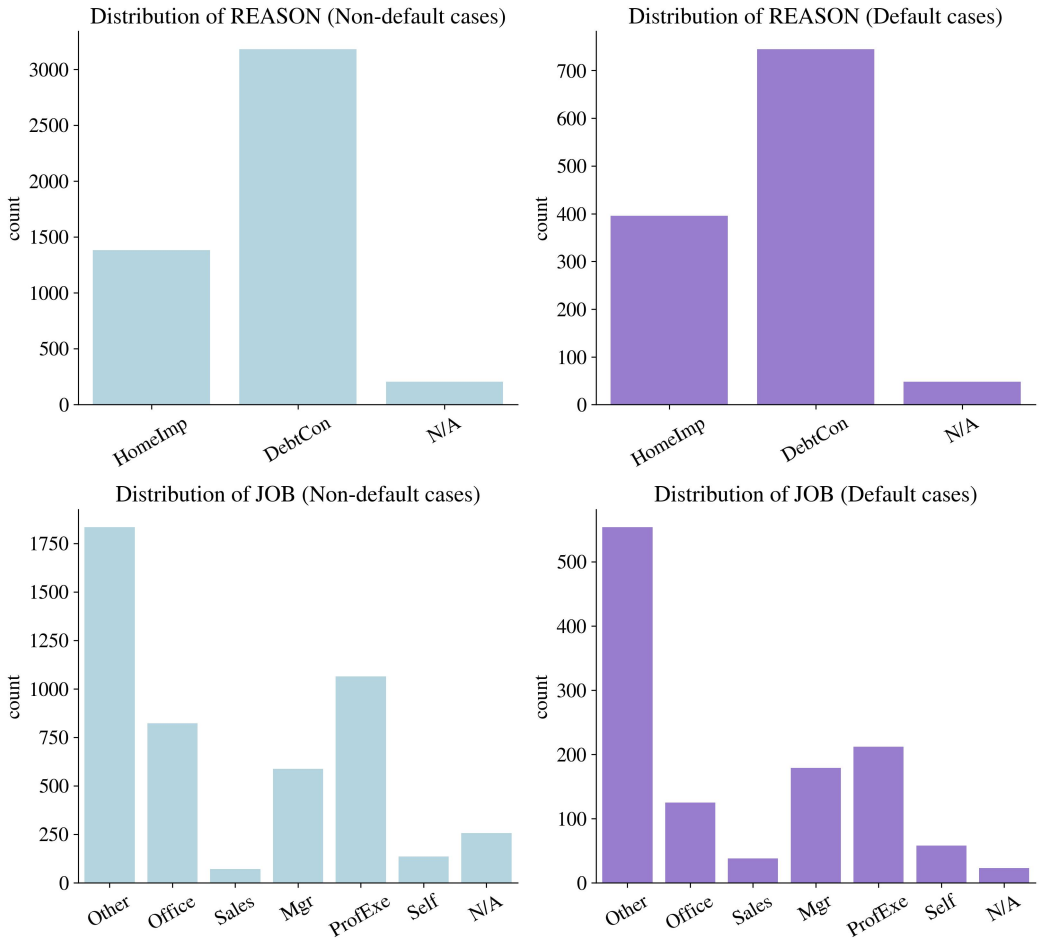
Source: Author's results in Python

Categorical Features' Distribution

Pertaining to the categorical features, such dataset has 2 nominal features - `REASON` and `JOB`. The following Figure 4.4 depicts the conditional distribution of categorical features on the default status visualized using barplots. We can observe that most of the loan applicants applied for a loan because of the debt consolidation and/or their job occupancies were labelled as `Other`.

Regarding the default status, there does not seem to be a significant difference between the default and non-default cases in terms of relative distribution of `REASON`. However, we can observe a slight difference between the default and non-default cases in terms of relative distribution of `JOB` feature. Particularly, in the categories `Office`, `ProfExe` and `N/A` there is relatively higher proportion of non–default cases than default cases. Therefore, there could be a moderate association between the `JOB` feature and the default status – such association is further investigated in Section 4.2.3.

Figure 4.4: Conditional distribution of categorical features



Source: Author's results in Python.

4.2.3 Association Analysis

In this subsection, we focus on the association analysis in order to inspect potential relationships between the variables. First, we inspect the association between the default status and the features and aftewards, we check for the association amongst the features themselves.

Association between default status and numeric features

In order to measure an association between the target variable and the numeric features, we use Point–Biserial correlation which denotes the value of Pearson's product moment correlation between the continuous variable and dichotomous variable (Kornbrot 2014). Such coefficient ranges from -1 to +1.

$$r_{pb,X} = \frac{\mu\left(X|Y=1\right) - \mu\left(X|Y=0\right)}{\sigma_X} \sqrt{\frac{N\left(Y=1\right) \times N\left(Y=0\right)}{N\left(N-1\right)}} \tag{4.1}$$

where $\mu\left(X|Y=1\right)$ and $\mu\left(X|Y=0\right)$ are the means of given numeric feature X conditional on the default status and non–default status, respectively, σ_X is the standard deviation of given numeric feature X, $N\left(Y=1\right)$ and $N\left(Y=0\right)$ are the numbers of observations with default status and non–default status, respectively, and N is the total number of observation within given feature X.

In the following Table 4.3, we can observe the computed Point-Biserial coefficient for each numeric feature with respect to the default status, including its statistical significance. As can be seen, features such as `DEROG`, `DELINQ` and `DEBTINC` are moderately and positively association with default status on 1% statistical significance level. Thus, we can confirm the findings from Section 4.2.2 regarding the positive associations of the features `DEROG`, `DELINQ` and `DEBTINC` with the default status. It can be deemed and inferred that such features may be important predictors within modelling.

Table 4.3: Point–Biserial Correlation table

Feature	**Coefficient**	**Significance**
LOAN	-0.075	***
MORTDUE	-0.048	***
VALUE	-0.030	**
YOJ	-0.060	***
DEROG	0.276	***
DELINQ	0.354	***
CLAGE	-0.170	***
NINQ	0.175	***
CLNO	-0.004	
DEBTINC	0.200	***

*: $p < 0.10$, **: $p < 0.05$, ***: $p < 0.01$

Source: Author's results in Python.

ok

ok

Association between default status and categorical features

For measuring a relationship's strength between the dichotomous and categorical variables, we use Cramer's V which ranges from 0 to 1 and is defined as:

$$CV_X = \sqrt{\frac{\chi^2}{N\,(k-1)}} \tag{4.2}$$

As expected based on findings in Section 4.2.2, according to the following Table 4.4, the association between the default status and `REASON` is really weak as the Cramer's V is close to zero. On the other hand, we can observe slightly stronger association of `JOB` with default status as some of its categories, namely `Office`, `ProfExe` and `N/A`, showed a higher proportion of non–default cases compared to the default cases. Note both `REASON`'s and `JOB`'s associations with default status are statistically significant on 1% significance level. Although statistical significance is important, it does not guarantee that a feature is a strong predictor of the target variable. Ultimately, the performance metrics of the model are what determine the usefulness of a feature in predicting the target variable.

Table 4.4: Cramer's V Association table

Feature	Coefficient	Significance
REASON	0.038	***
JOB	0.120	***

*: $p < 0.10$, **: $p < 0.05$, ***: $p < 0.01$

Source: Author's results in Python.

Association between default status and missing values

Since the loan data contains missing values, it is deemed appropriate to check whether the missing values are associated with the default status as well. One way how to deal with the measurement, is to encode the feature into a binary variable where 1 indicates a missing value occurrence and 0 otherwise.

For measuring a strength of association between the two binary variables, we use Phi coefficien which is defined as:

$$\phi_X = \sqrt{\frac{\chi^2}{n}} \tag{4.3}$$

In line with the finding regarding the `DEBTINC` and `VALUE` in Section 4.2.2, as shown in Table 4.5, we can observe a strong and statistically significant association between the missing debt–to–income and default status, respectively a moderate and statistically significant association between the missing collateral property value and the default status. Hence, we can expect such indicators will be the main drivers in a default prediction.

Table 4.5: Phi Correlation Coefficient table

Feature	**Coefficient**	**Significance**
LOAN	0.000	
MORTDUE	0.003	
VALUE	0.254	***
REASON	0.004	
JOB	0.064	***
YOJ	0.056	***
DEROG	0.070	***
DELINQ	0.061	***
CLAGE	0.030	**
NINQ	0.039	***
CLNO	0.018	
DEBTINC	0.547	***

*: $p < 0.10$, **: $p < 0.05$, ***: $p < 0.01$

Source: Author's results in Python.

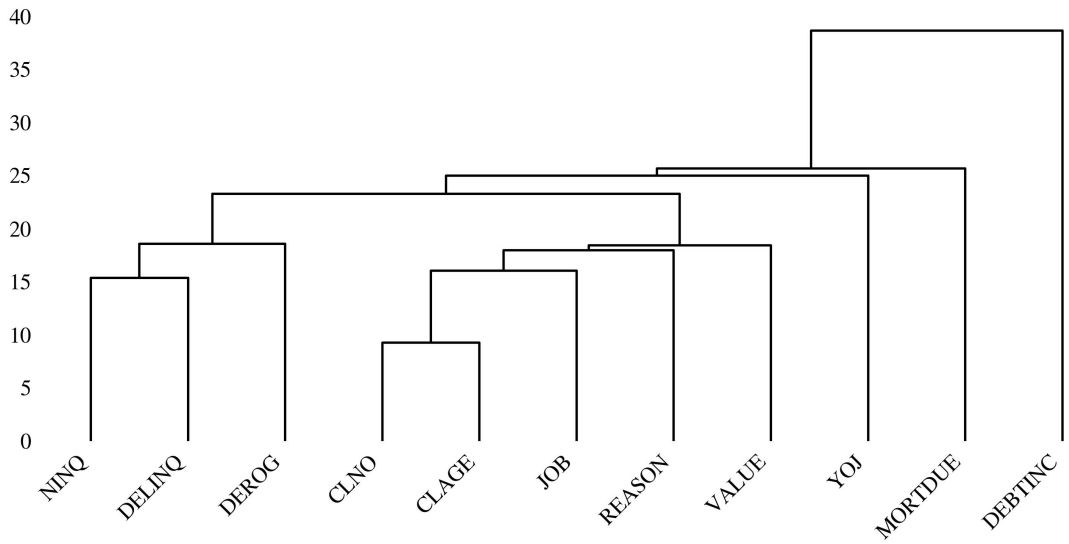
Missing Values Association

Furthermore, it is needed to inspect how the missing values are related to default status, but also how they are associated between themselves. One way, how to indetify any patterns of missing data in dataset is using dendrogram which hierarchically clusters the variables based on the missing values' occurrences. Hence it groups the variables into clusters based on the similarity of their missing value patterns ,i.e., variables with similar patterns of missingness are clustered together, while variables with dissimilar patterns of missingness are placed in separate clusters. The dendrogram is constructed by iteratively

merging the two closest clusters until all variables are in the same cluster. The distance between the clusters at each step of the merging process is shown on the y-axis of the dendrogram, and the order in which the variables are merged is shown on the x–axis.

The following Figure 4.5 depicts the hierarchical clustering of the dataset's variables, excluding the default status the requested loan amount `LOAN` since these variables do not contain any missing values. As can be seen, features `CLNO` and `CLAGE` are the most similar in terms of missing values' occurrences Thus, most of the loan applicants, when submitting their loan application, they oftenly omit their number of credit lines (`CLNO`) as well as the age of their recent credit line (`CLAGE`).

Figure 4.5: Nullity dendrogram



Source: Author's results in Python.

Multicolinearity Analysis

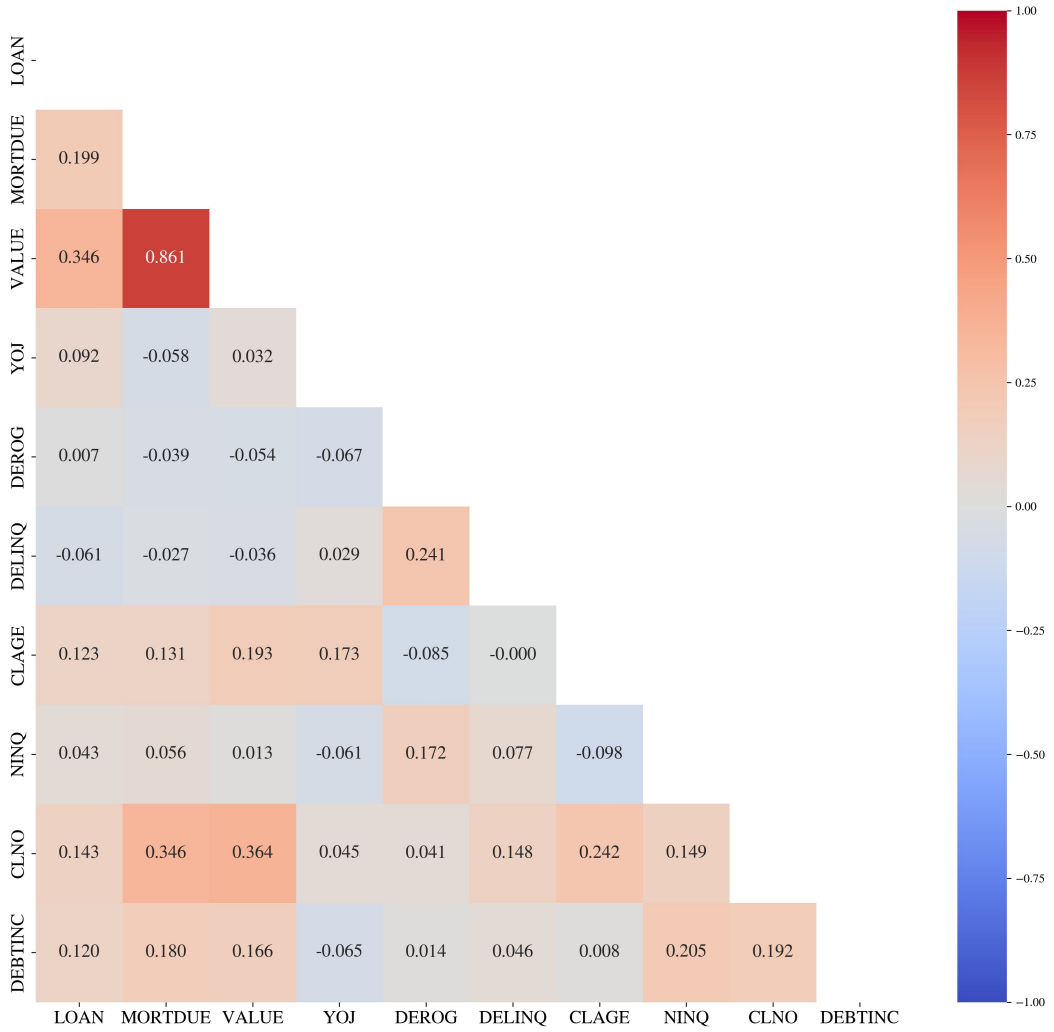
ADD contigency table for REASON and JOB.

To measure the correlation between the numeric features, one would use Pearson correlation coefficient. However, such measure is very sensitive to outliers and assumes normal distribution of variables as well as the linear relationship between the variables. Therefore, we use Spearman correlation coefficient instead, which is a nonparametric measure of the association between two continuous variables. It is defined as:

$$\rho_{spearman} = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n\left(n^2 - 1\right)} \tag{4.4}$$

In the Figure 4.6, we can observe a very strong correlation between the `MORTDUE` and `VALUE` features. Such multicolinearity can cause problem in predictions na model's overfitting. Therefore, a feature selection is recommended - such selection is further described in Subsection 4.4.2.

Figure 4.6: Spearman Correlation Matrix



Source: Author's results in Python.

4.3 Data Preprocessing

In this section, the process of preprocessing data is described as the crucial step in the machine learning modelling. Particularly, the following is described:

- Splitting the data
- oversampling,
- discretization,
- Weight–of–Evidence encoding.

4.3.1 Data Split and ADASYN Oversampling

In order to ensure the appropriate model training and unbiased evaluation, it is needed to split our data into sets where each sample has its own purpose. We split our data into following sets in the ratio of 70:15:15:

- Training set - such set is used for training the model, feature selection and hyperparameter tuning.
- Validation set - such set is used for selection of the best final model.
- Test set - such set is used for the final evaluation of the best final model.

It is important to split the data into separate sets so we could independently train and tuned the several models as well as choose the best features, select the best model and evaluate it on the unseen data as we ensure an avoidance of overfitting as well as the data leakage.

As already, since the default status distribution highly imbalanced, it is needed to treat such issue, otherwise the model would be biased towards the non–defaults. Hence, the data is split into such sets using stratified split, which ensures that the default status distribution remains the same across all the sets. Thus, when splitting the data with a stratification, the default distribution is each set is preserved, i.e., 80 % of non–defaults and 20 % of defaults in each set. With a stratification, it is possible to train the model on the same population in which it is being evaluated leading to more accurate predictions (Igareta 2021).

Despite the stratification, it does not have to be sufficient when dealing with imbalanced class. One approach is to use undersampling which removes the instances with majority class. However, this is not deemed appropriate since our data sample size is relatively small and we would lose a lot of information by removing such instances. Other approach is to use random oversampling, which randomly duplicates the instances of minority class in such way that the

target distribution is then balanced. Although, this is not deemed appropriate as well since as it could lead to the overfitting as the model would just memorize the duplicated instances and would not be able to generalize well on the unseen data.

Instead, ADASYN (Adaptive Synthetic Sampling) is used, which is an oversampling technique that generates synthetic instances of the minority class based on the nearest neighbors of the minority class instances. Compared to SMOTE (Synthetic Minority Oversampling Technique), ADASYN is more effective as it uses density distribution of each minority instance which generates more synthetic samples in regions where the density of the minority class is lower and fewer synthetic instances in regions where the density is higher. In other words, ADASYN focuses on difficult–to–learn minority instances, i.e., instances having lower density, therefore it generates more synthetic instances for such instances (He *et al.* 2008). Thereby, it makes easier for the machine learning model to learn the decision boundary between the minority and majority classes and boost the model's performance.

It is important to emphasize, that the oversampling has to be performed on the training set only after the split and not before the split or on validation or test sets. Otherwise it would lead to data leakage and biased evaluation of the model. In the following Table 4.6, we can observe some split information regarding the individual sets. As can be seen, either test, validation and training sets (before oversampling) have the same default distribution which is desired thanks to the stratification. The table also depicts the information about the training set after performed ADASYN oversampling, where the default distribution is balanced.

Table 4.6: WoE distribution

Set	**# instances**	**% defaults**	**% non–defaults**
Training	4,171	19.95 %	80.05 %
Training (oversampled)	6,437	48.13 %	51.87 %
Validation	895	20.00 %	80.00 %
Test	894	20.00 %	80.00 %

Source: Author's results in Python.

In Python, the data are split and oversampled using a custom function `data_split()` which first splits the data into training, validation and test sets

using `train_test_split()` function with a stratification from `scikit-klearn` module. Then, the training set is oversampled using `ADASYN()` class with 5 nearest neighbors and Euclidean distance from `imblearn` module.

However, the `imblearn`'s `ADASYN()` class cannot handle either missing values or categorical variables. Therefore, the approach in order to deal with such obstacle is following:

1. Separate the categorical and numeric features;
2. Impute the missing values with arbitrary values:
   - Categorical features: string `'N/A'`;
   - Numeric features: number `99999999999999` - such value is chosen since it is highly unlikely to be present in the dataset.
3. Convert the categorical features into dummy variables;
4. Join the numeric features with the dummy variables;
5. Perform the oversampling on the joined dataset;
6. Convert the dummy variables back into categorical features;
7. Retrieve back the missing values:
   - Categorical features: replace string `'N/A'` with `np.nan`;
   - Numeric features: for each feature X if its value exceeds the original maximum value, then replace it with `np.nan`;[1]

4.3.2 Optimal Binning and WoE Encoding

One of the most common approach of feature preprocessing are for instance - dummy encoding, standardization, logarithmic transformation, normalization, etc. However, these approaches are not always suitable for the given dataset. For instance, the dummy encoding is not suitable for the categorical features with a large number of categories, since it would lead to the curse of dimensionality. The standardization is not suitable for the features with a large number of outliers, since it would lead to the loss of information. The logarithmic transformation is not suitable for the features with a large number of zeros, since it would lead to the loss of information. The normalization is not suitable for

[1]The theory behind this ADASYN will be described later.

the features with a large number of outliers, since it would lead to the loss of information and so on.

Instead, a discretization or so called binning is used as it can captures outliers within bins, which is not possible with the standardization or normalization. It also captures missing values, so there is no need to remove or impute missing values or removing outliers thanks to such approach.

In Python, `BinningProcess` from `optbinning` module is used for an optimal binning of both numeric features into interval bins and categorical features into category–group bins, with respect to the target variable. Such bins as categories are then encoded into numeric values using Weight-of-Evidence[2] which is calculated as:

$$WoE_{X,b} = \ln\left(\frac{\Pr\left(X = b \mid Y = 0\right)}{\Pr\left(X = b \mid Y = 1\right)}\right) \tag{4.5}$$

The following Figure 4.7 depicts Weight–of–Evidence bins distribution per each feature. It can be observed that the binning captures either linear, non–linear, monotonic or non–monotonic relationship between the default status and the numeric features in terms of WoE. With respect to the `DELINQ` future, we can observe a monotonic relationship as the higher number of delinquent credit lines is, the lower WoE cofficient is, i.e., the larger distribution of defaults with respect to the non-defaults in given bin. Thus the higher number of delinquent credit lines, the higher likelihood of defaulting in terms of WoE.

A non–linear relationship can be observed with respect to the `YOJ` feature, where the WoE coefficient is positive for applicants who just have recently started working at their new job (i.e., number of years at the present job is less than 1) and for applicants who are working at their current job relatively for a long time (i.e., number of years at th present job is higher than 19). Thus, applicants who are working for a longer time, have stable income and therefore are more creditworthy and less likely to default. Regarding the applicants who just have recently started working at their new job, it is possible that they are less likely to default since the `YOJ` feature does not capture the applicant's total number of years of work experience, but only the number of years at the present job. Thus, in the given dataset, the applicants who just have recently started working at their new job, have a relatively higher total number of years

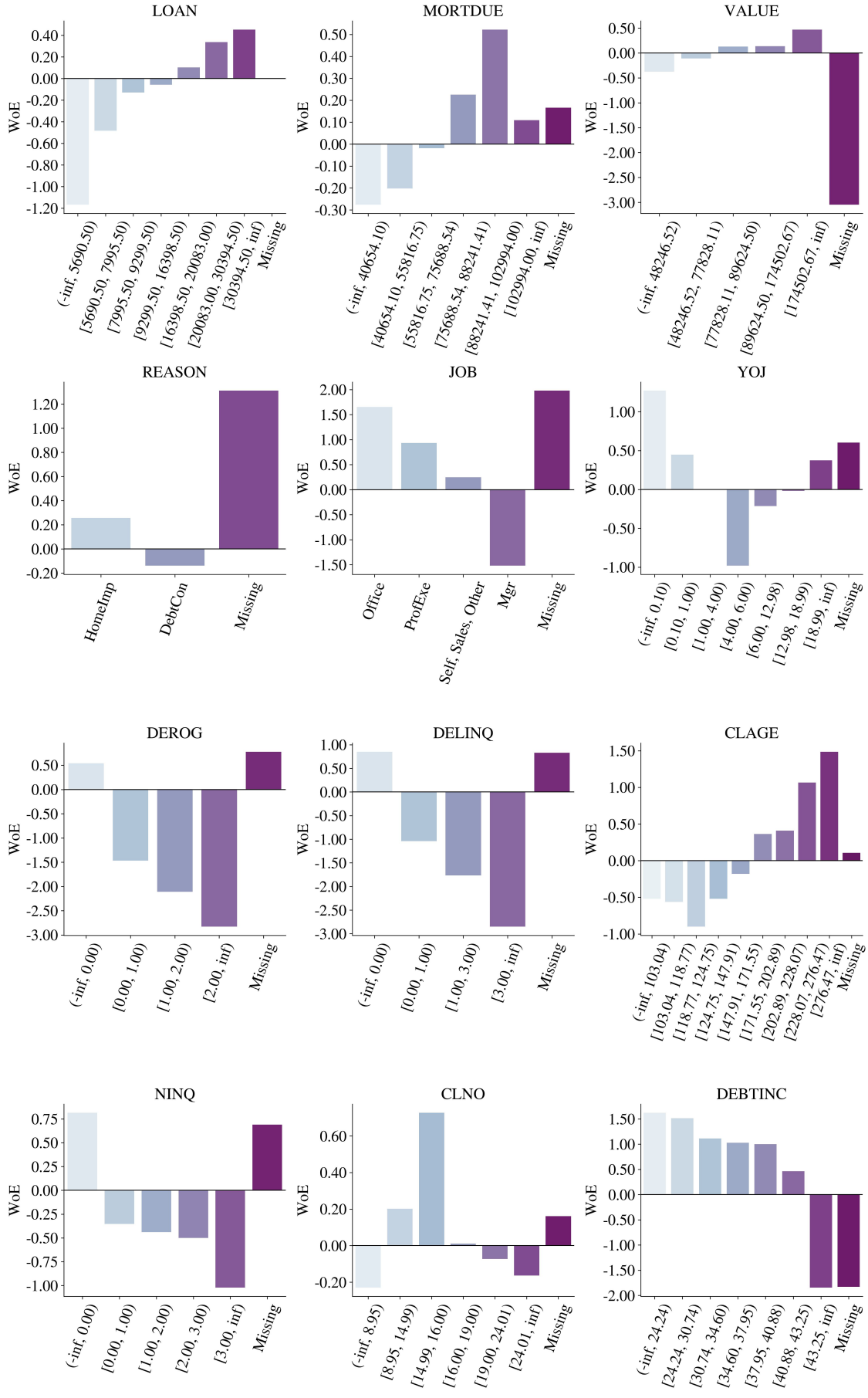
[2]The theory behind this optimal binnging will be described later.

of work experience, thus they are more creditworthy and less likely to default. On the other hand, applicants who are working at their present job between 1 and 19 years, the WoE coefficient is negative.This relationship seems to be complex and can be influenced by other factors which are not present in the dataset, such as the applicant's age, total number of years of work experience, education etc.

Both numeric and categorical features contain a separate bin capturing missing values which can be useful indicator when training a model. This can be apperant in the `DEBTINC` feature, where the bin capturing missing values has the most negative WoE coefficient, indicating that there is a larger distribution of defaulteres compared to non-defaulters. Such finding was already raised in Section 4.2.3 in terms of strong and statistically significant association between the default status and the missing values in `DEBTINC`.

Nevertheless, we can also observe some incosistencies in terms of distributions from exploratory analysis and the WoE coefficient distributions. One example pertains to the category `Mgr` in `JOB` feature - the WoE coefficient is substantially negative, which would indicate that there amongst the managers, there is a larger distribution of defaulters compared to non–defaulters. However, this was not observed in the distribution analysis of categorical features (Section 4.2.2), where the relative distributions conditional on the default status does not differ too much in terms of `Mgr` category. This is a result of ADASYN oversampling, which synthetically replicates minority instances, especially those instances which are hard–to–learn. Hence defaulted loans which the managers have applied for are difficult to learn, therefore ADASYN balances default distribution by generating default instances having `JOB` equal to `Mgr`, which results in an increase of number of instances having `JOB` equal to `Mgr`, i.e., in a larger distribution of defaulters in `Mgr` bin, therefore in a negative WoE coefficient.

Figure 4.7: WoE Bins Distribution



Source: Author's results in Python.

4.4 Modelling

Once the data are finally preprocessed, the next step regards the modelling part which includes hyperparameter tuning, feature selection, model selection and model building.

In Python, 8 different machine learning models from `Scikit-learn` module are used for the default status prediction, which are:

- Logistic Regression - `LogisticRegression from sklearn.linear_model`,
- Decision Tree - `DecisionTreeClassifier from sklearn.tree`,
- Gaussian Naive Bayes - `GaussianNB from sklearn.naive_bayes`,
- K-Nearest Neighbors - `KNeighborsClassifier from sklearn.neighbors`,
- Random Forest - `RandomForestClassifier from sklearn.ensemble`,
- Gradient Boosting - `GradientBoostingClassifier from sklearn.ensemble`,
- Support Vector Machine - `SVC from sklearn.svm`,
- Neural Network - `MLPClassifier from sklearn.neural_network`.

4.4.1 Hyperparameter Bayesian Optimization

Instead of using models with default hyperparameters, it is deemed desired to choose optimal hyperparameter values in order to boost model's performance. One would use Grid Search or Random Search which are commonly used in hyperparameter tuning. Nonetheless, these methods are computationally expensive and do not guarantee to find the global optimum. Also they do not consider any information from previous iterations but rather checks all the possible hyperparameters combinations or randomly chooses hyperparameters combinations, respectively.

Therefore, Bayesian Optimization is used for hyperparameter tuning instead of Grid Search or Random Search, as a probabilistic model using Gaussian Process is used to approximate the objective function, which we want to target. Such hyperparameter tuning approach uses Bayesian inference to update prior distribution over hyperparameters' values based on the result of previous iterations and uses resulting posterior distribution to sample guide the selection of the next set of hyperparameters to evaluate.[3].

[3]Theory behind Bayesian Optimization will be described later

In Python, a custom function `bayesian_optimization()` is used which further uses `BayesSearchCV` class from `Scikit-optimize` module with 10-fold stratified cross-validation, 50 iterations while maximizing F1 score. Therefore, each model, the Bayesian Optimization 50 times looks for the best hyperparameters' values while maximizing F1 score where within each iteration, 10-fold stratified cross-validation is used to evaluate the model's F1 score.

The hyperparameter space is defined for each model as follows:

Logistic Regression

Table 4.7: Logistic Regression - Hyperparameter Space

Hyperparameter	**Space**
Intercept	True, False
C factor	$<1 \times 10^{-6}$, 5>
Penalty	L1, L2, Elastic Net, None
Solver	lbfgs, liblinear, newton-cg, sag, saga
Class weight	None, balanced
L1 ratio	<0, 1>
Intercept scaling	True, False

Source: Author's results in Python.

Decision Tree

Table 4.8: Decision Tree - Hyperparameter Space

Hyperparameter	**Space**
Criterion	Gini, Entropy
Max depth	<1, 10>
Max features	<1, `len(X.columns)`>

Source: Author's results in Python.

Gaussian Naive Bayes

Table 4.9: Gaussian Naive Bayes - Hyperparameter Space

Hyperparameter	Space
Variance smoothing	$<1 \times 10^{-9}$, $1 \times 10^{-6}>$

Source: Author's results in Python.

K-Nearest Neighbors

Table 4.10: K–Nearest Neighbors - Hyperparameter Space

Hyperparameter	Space
# neighbors	<5, 20>
Weights	Uniform, Distance
Algorithm	Ball Tree, KD Tree, Brute, Auto
Metric	Euclidean, Manhattan, Cityblock, Minkowski

Source: Author's results in Python.

Random Forest

Table 4.11: Random Forest - Hyperparameter Space

Hyperparameter	Space
# estimators	<100, 1000>
Criterion	Gini, Entropy, Log Loss
Max depth	<1, 10>
Max features	<1, `len(X.columns)`>
Class weight	None, balanced, subsample balanced
Bootstrap	True, False
CCP alpha	$<1 \times 10^{-12}$, 0.5>

Source: Author's results in Python.

Gradient Boosting

Table 4.12: Gradient Boosting - Hyperparameter Space

Hyperparameter	**Space**
# estimators	<100, 1000>
Criterion	Friedman MSE, Squared Error
Max depth	<1, 10>
Max features	<1, `len(X.columns)`>
Loss	Log Loss, Exponential
Learning rate	<0.0001, 0.2>

Source: Author's results in Python.

Support Vector Machine

Table 4.13: Support Vector Machine - Hyperparameter Space

Hyperparameter	**Space**
C factor	$<1 \times 10^{-6}$, 5>
Kernel	Linear, Poly, RBF, Sigmoid
Degree	<1, 10>
Gamma	scale, auto
Shrinking	True, False
Decision function shape	OVR, OVO
tol	$<1 \times 10^{-9}$, 1×10^{-3}>
Class weight	balanced, None

Source: Author's results in Python.

Neural Network

Table 4.14: Multi Layer Percepton - Hyperparameter Space

Hyperparameter	**Space**
Hidden layer size	<5, 500>
Activation function	Identity, Logistic, Tanh, ReLU
Solver	Adam, SGD, LBFGS
Learning rate	Constant, Adaptive, Invscaling

Source: Author's results in Python.

4.4.2 Feature Selection

In subsection, the process of selecting optimal features is described. Instead of using all the features in dataset, only the most relevant are chosen and the noisy ones are eliminated.

For such case, the Forward Sequential Feature Selection is used. The approach is following - for each model:

1. tune the model with Bayesian Optimization;
2. use such tuned model as an input estimator within Forward Sequential Feature Selection;
3. get the best features for each model.

Thus, when having n input models, it returns n subsets of optimal features, one per each model.

In Python, a custom function `SFS_feature_selection()` is used which further uses another custom function defined previously `bayesian_optimization()` for tuning the model, and `SequentialFeatureSelector` class from `Scikit-learn` for feature selection. `SequentialFeatureSelector` class is predetermined with the `forward` direction, 10–fold stratified cross validation while maximizing F1 score.

Instead of choosing a fixed number of features, `SequentialFeatureSelector` class also allows to choose a variable number of features due to the early stopping (`tol` paramater) - The `SequentialFeatureSelector` stops adding new features into optimal subset if the objective score is not incremented by at least `tol` between two consecutive feature additions(Scikit-learn n.d.). The `tol` parameter is set to the number close to zero indicating that the feature selection should stop adding new features if the F1 score is not increasing between two consecutive feature additions.

The custom function `SFS_feature_selection()` iterively prints the process of the feature selection for in order to keep the track of such process as it is depicted in Figure 4.8. Particularly, it prints which step model and which step is currently being executed (whether it is Bayesian Optimization or Feature Selection), the execution time and the selected features for each model.

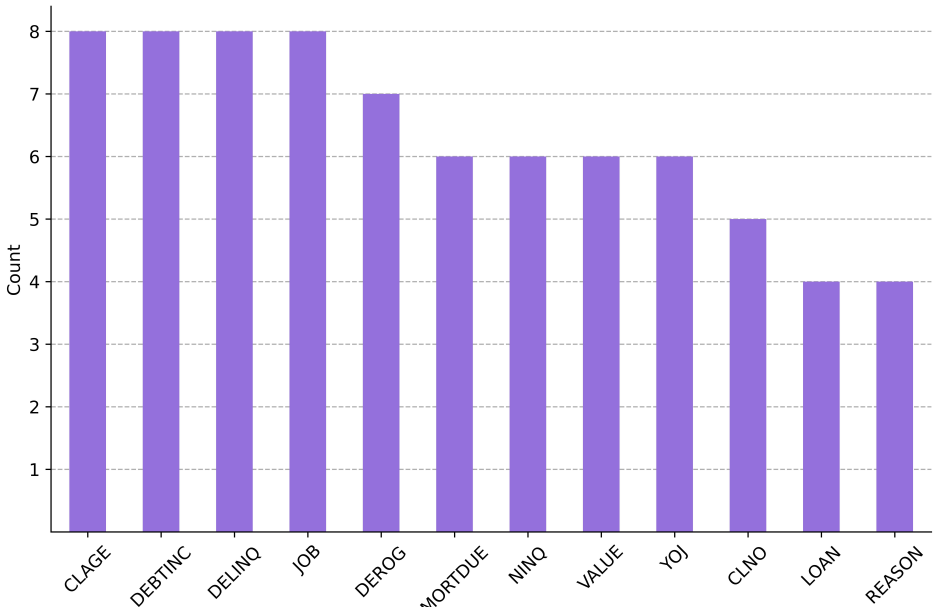
Figure 4.8: Feature Selection Print Statement

```
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------ 3/8 -------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
---------------------------------- FEATURE SELECTION WITH GAUSSIAN NAIVE BAYES ----------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------

    1/4 ... Starting Bayesian Optimization on the whole set of features
    2/4 ... Bayesian Optimization finished
    3/4 ... Starting Forward Sequential Feature Selection
    4/4 ... Forward Sequential Feature Selection with finished

    Execution time: 23 seconds

    6 features selected: MORTDUE, JOB, DELINQ, CLAGE, NINQ, DEBTINC

------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------



------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------ 4/8 -------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------- FEATURE SELECTION WITH K NEIGHBORS CLASSIFIER -------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------

    1/4 ... Starting Bayesian Optimization on the whole set of features
    2/4 ... Bayesian Optimization finished
    3/4 ... Starting Forward Sequential Feature Selection
    4/4 ... Forward Sequential Feature Selection with finished

    Execution time: 78 seconds

    11 features selected: LOAN, MORTDUE, VALUE, REASON, JOB, YOJ, DEROG, DELINQ, CLAGE, CLNO, DEBTINC

------------------------------------------------------------------------------------------------------------------------
```

Source: Author's results in Python.

The following Figure 4.9 depicts the reccurrence of the selected features. As can be seen, features such as `CLAGE`, `DEBTINC`, `JOB` and `REASON` were selected by each model. On the other hand, features such as `LOAN` and `REASON` were selected by only 4 times. Therefore, we can expect that such features which were selected every time will have high importance in predictions. *the figure will be adjusted, it has been incorrectly exported.*

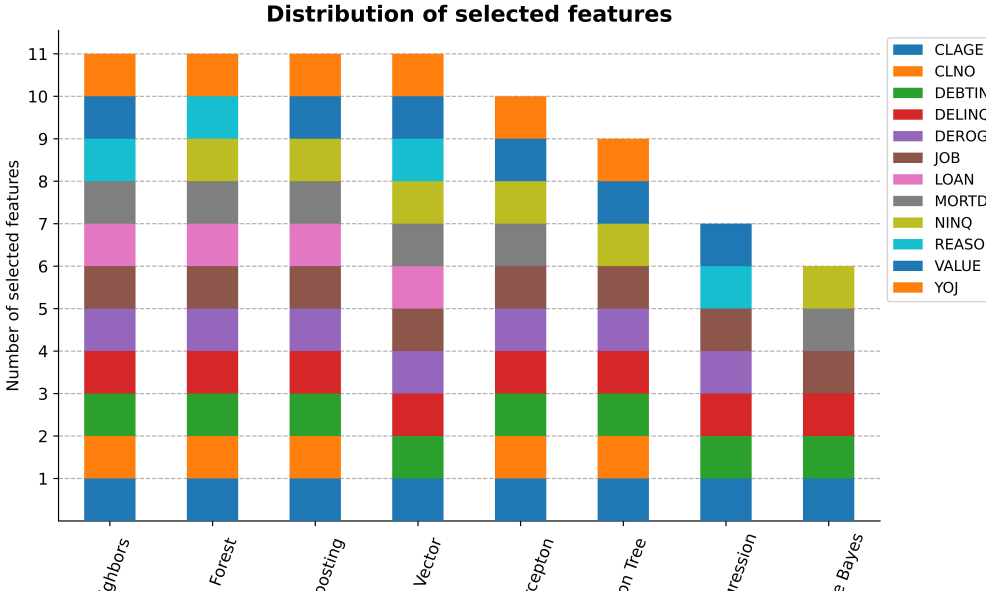
Figure 4.9: Reccurrence of Selected Features



Source: Author's results in Python.

According to Figure 4.10, models such as K–Nearest Neighbors, Random Forest, Gradient Boosting and Support Vector Machine chose all the features from the dataset. On the other hand, Gaussian Naive Bayes chose only 6. It seems to be that most of the features are important as each model has selected a higher amount of features. *the figure will be adjusted, it has been incorrectly exported.*

Figure 4.10: Distribution of Selected Features per Model



Source: Author's results in Python.

4.4.3 Model Selection

In combination with the pre–selected subsets of features, the next step regards the selection of the final model. The process is following:

1. tune each model on each subset of optimal features (selected within Feature Selection) with Bayesian Optimization on the training set.
2. evaluate tuned model on the validation set.

Thus, when having n input models and m subsets of selected features, we get $n \times m$ tuned models. $m \leq n$ because we exclude duplicated subset of selected features which can occur when more than one model choose the same subset(s) of features. Since there are 8 input models and 8 unique subsets of selected features, the total number of tuned models is 64.

Further, for each model, we evalute it on the validation set by computing following metrics scores and loss functions:

- F1 score,
- Precision,
- Recall,

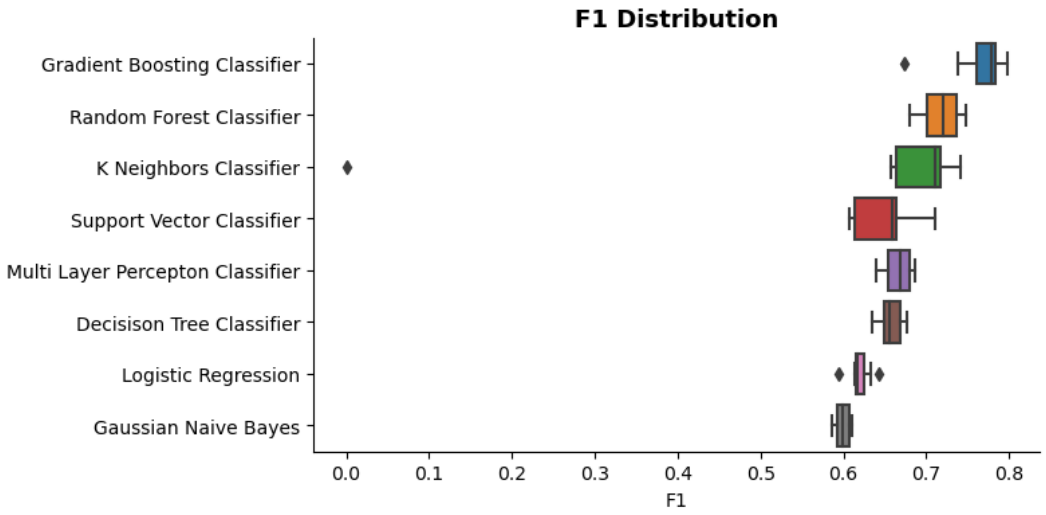
- Accuracy,
- AUC,
- Somers' D,
- Kolmogorov Smirnov Distance,
- Matthews Correlation Coefficient,
- Jaccard Score,
- Brier Score Loss,
- Zero-One Loss.

Pertaining to the class–based metrics which require predicted classes (and not predicted probability scores), such as F1 score, Precision, Recall, Accuracy, Matthews Correlation Coefficient, Jaccard Score and Zero-One Loss, by default they use a default classifiation threshold of 0.5. Meaning if the probability score is higher than 0.5, it will be encoded as class 1 and otherwise. Note that 0.5 clasification threshold does not have to be valid for real life use cases. Therefore, it is deemed desired to calculate an optimal threshold instead of using the default one. One way how to achieve this is to use Younden index which is derived from ROC curve

$$J = \text{TPR} - \text{FPR} \tag{4.6}$$

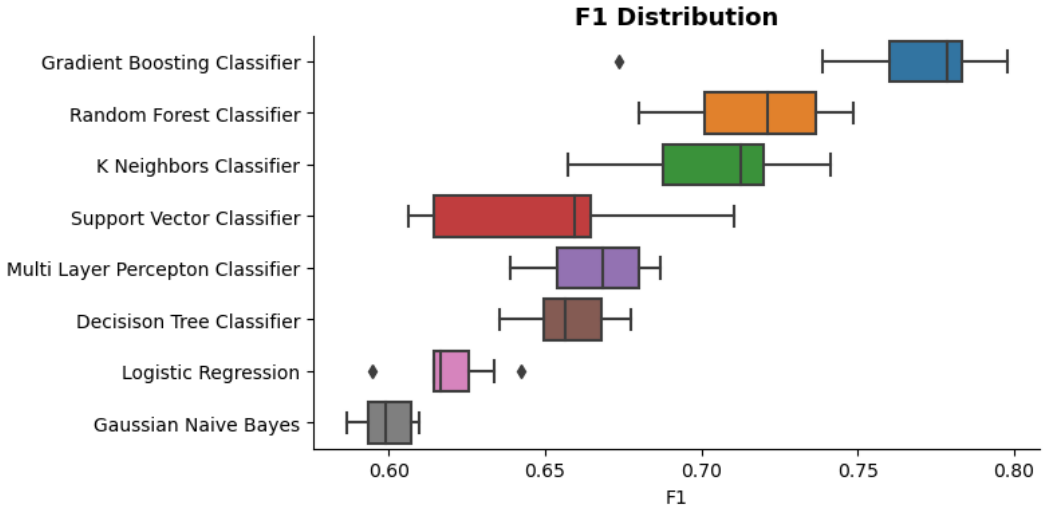
$$T_{opt} = \text{argmax}_{t \in [0,1]}(J) \tag{4.7}$$

Figure 4.11: F1 score distribution



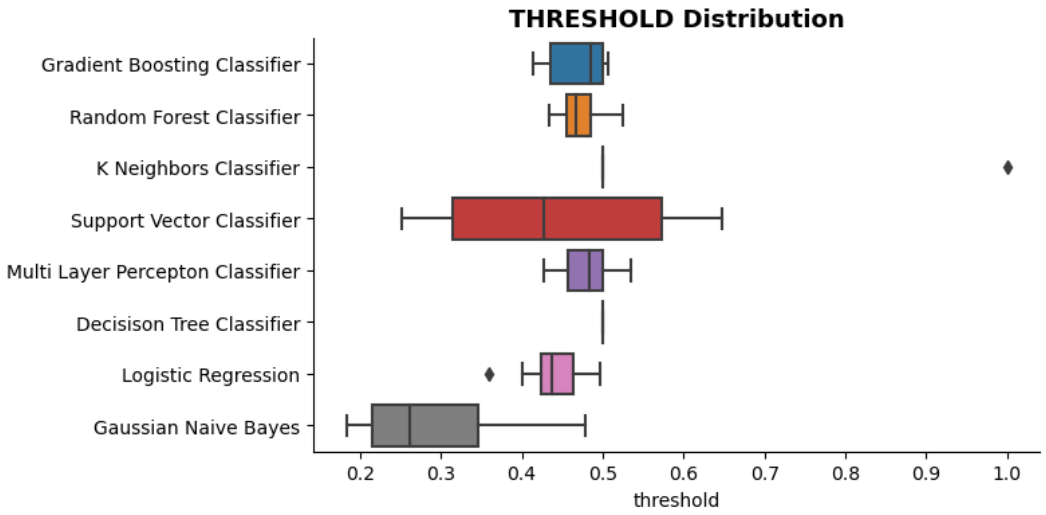
Source: Author's results in Python.

Figure 4.12: F1 score distribution - without outliers



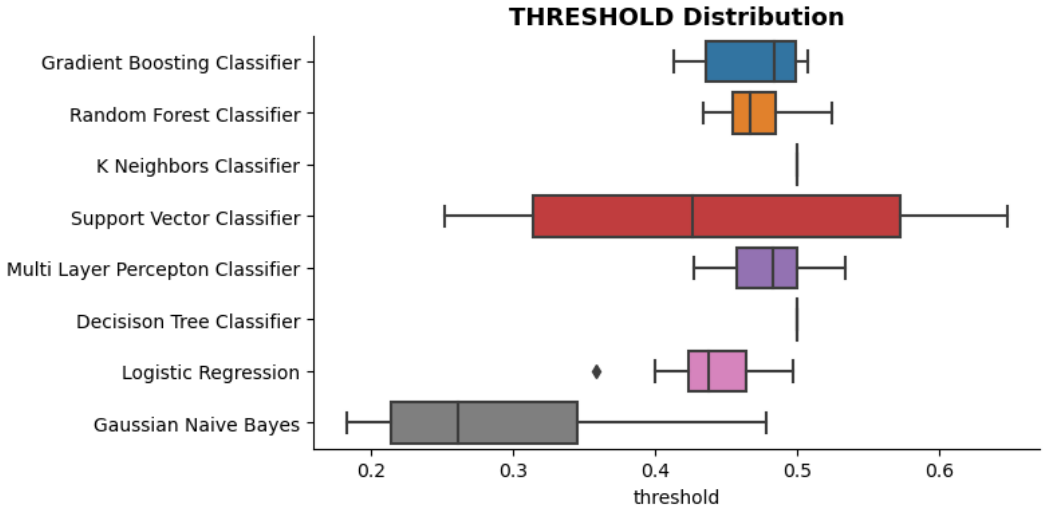
Source: Author's results in Python.

Figure 4.13: Classification Threshold distribution



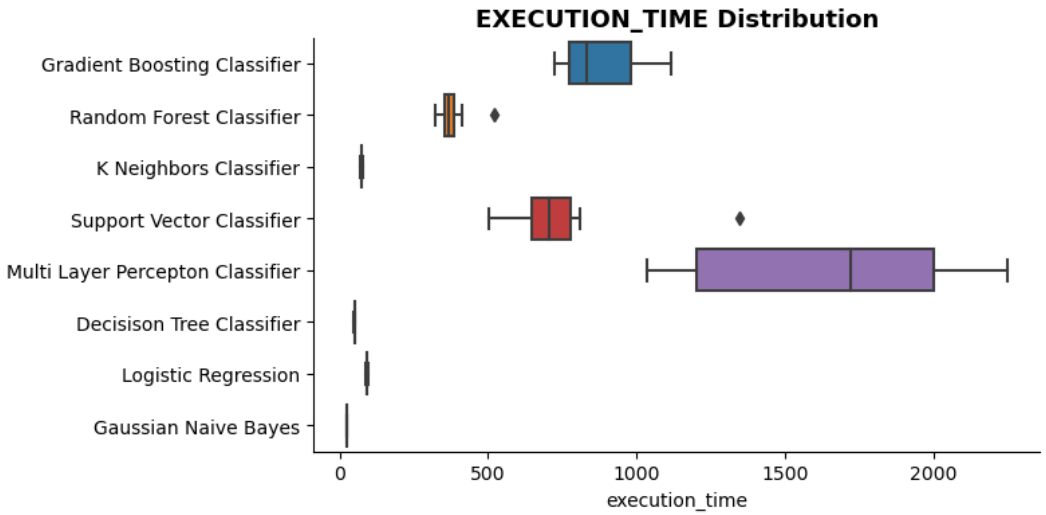
Source: Author's results in Python.

Figure 4.14: Classification Threshold distribution - without outliers



Source: Author's results in Python.

Figure 4.15: Execution time distribution



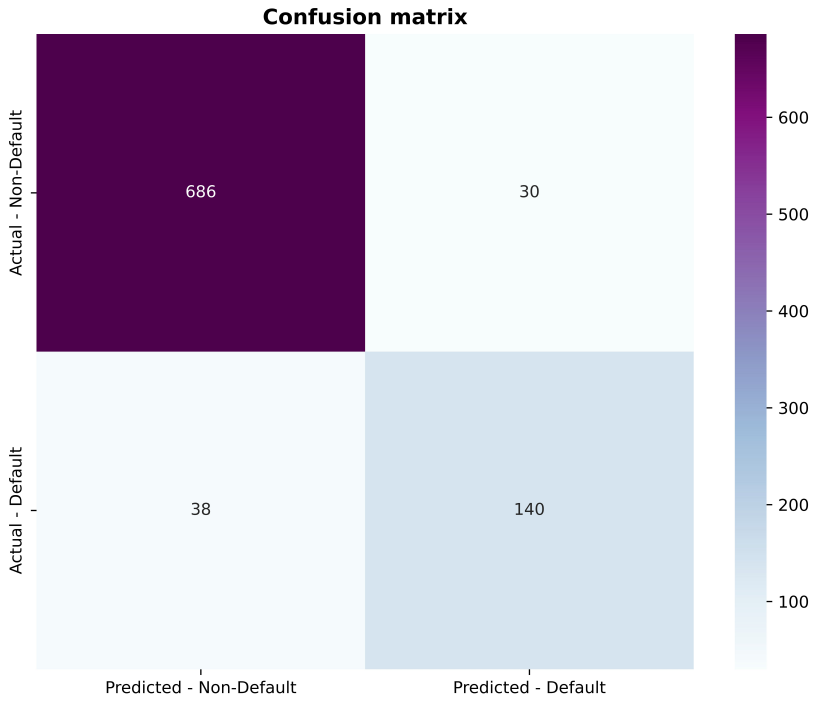
Source: Author's results in Python.

4.4.4 Model Building

4.5 Model Evaluation

4.5.1 Confusion Matrix

Figure 4.16: Confusion Matrix



Source: Author's results in Python.

4.5.2 Metrics Scores

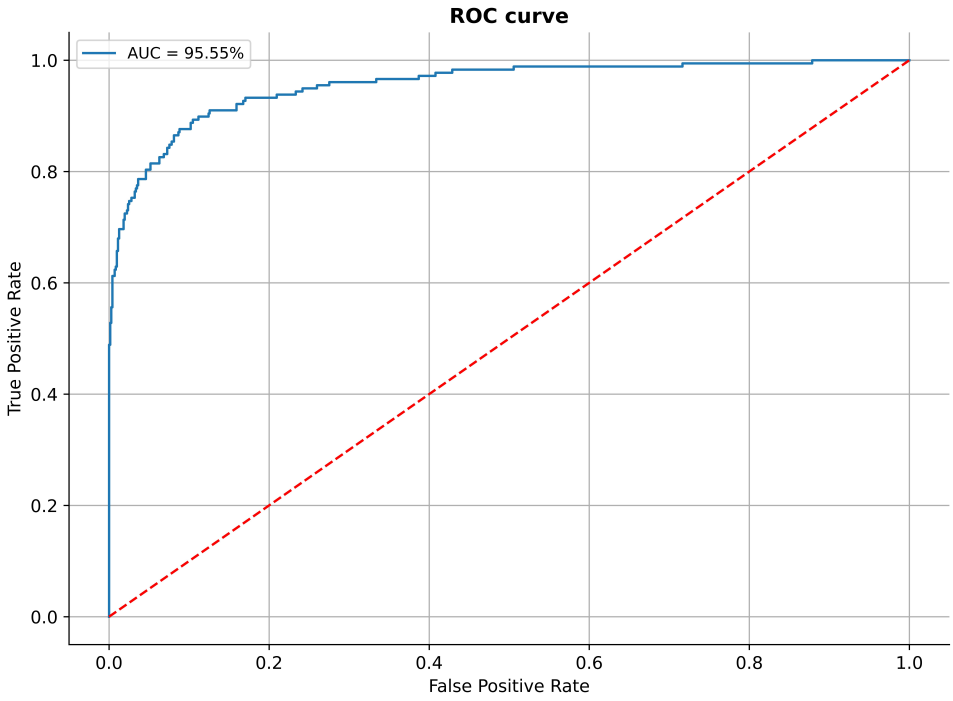
Table 4.15: Metrics Evaluation

Metric	Value
F1	0.8046
Precision	0.8235
Recall	0.7865
Accuracy	0.9239
AUC	0.9555
Somers D	0.9111
KS	0.7885
MCC	0.7577
Jaccard Score	0.6731
Brier Score Loss	0.0617
Zero-One Loss	0.0761

Source: Author's results in Python.

4.5.3 ROC Curve

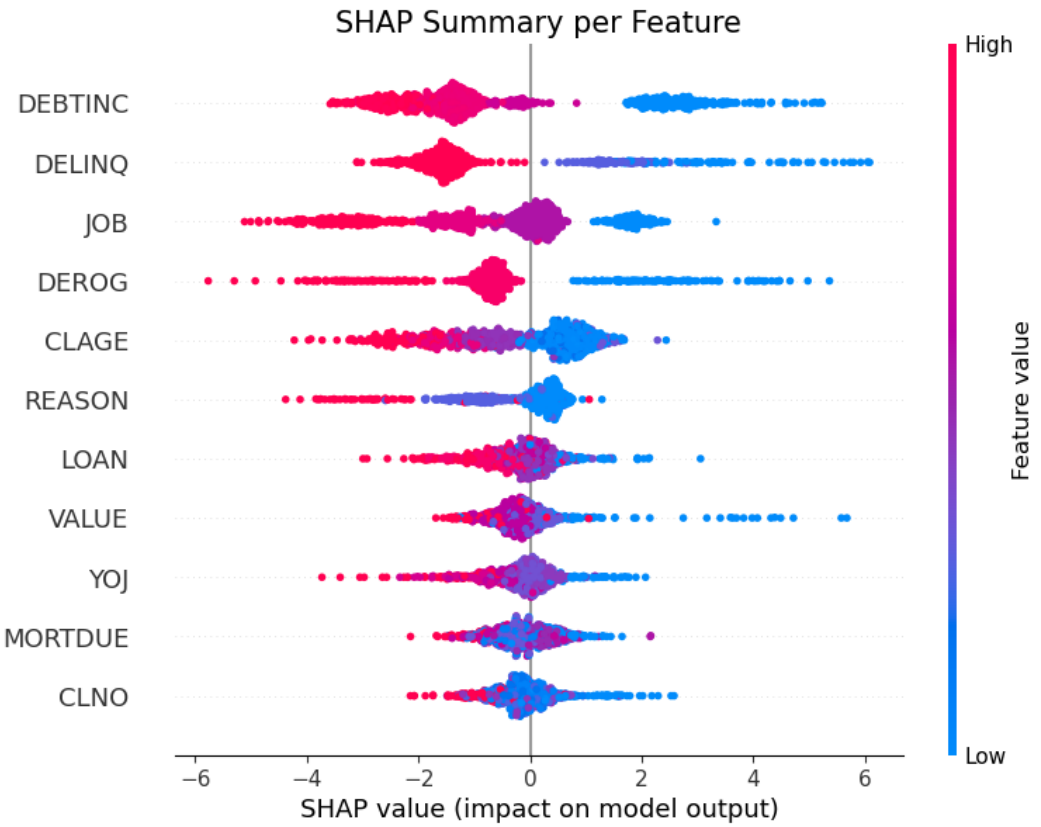
Figure 4.17: ROC Curve



Source: Author's results in Python.

4.5.4 SHAP Values

Figure 4.18: SHAP Summary Plot



Source: Author's results in Python.

4.6 Machine Learning Deployment

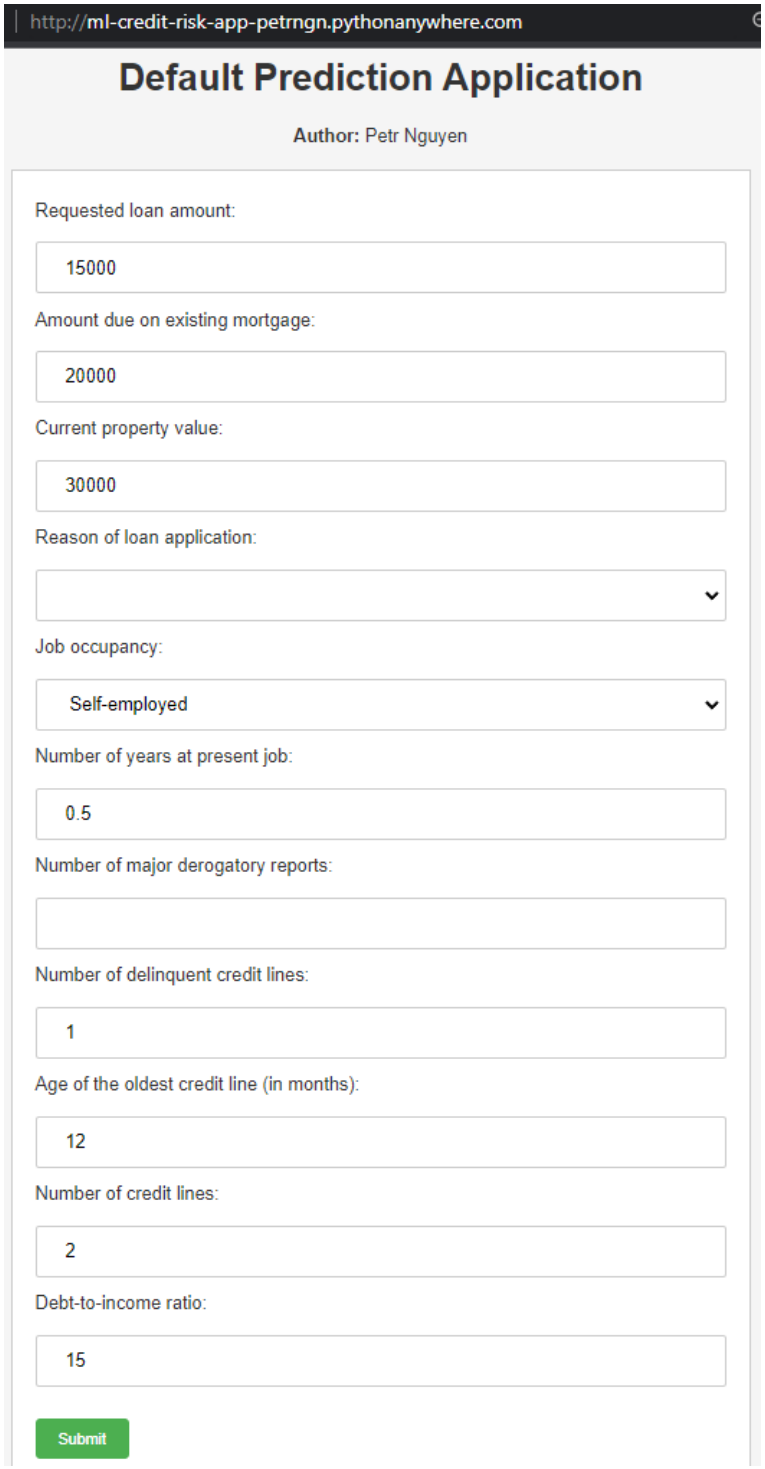
4.6.1 Final Model Building

4.6.2 Flask and HTML Web Application

Once the final model is trained or built, it can be deployed into a production and monitored.

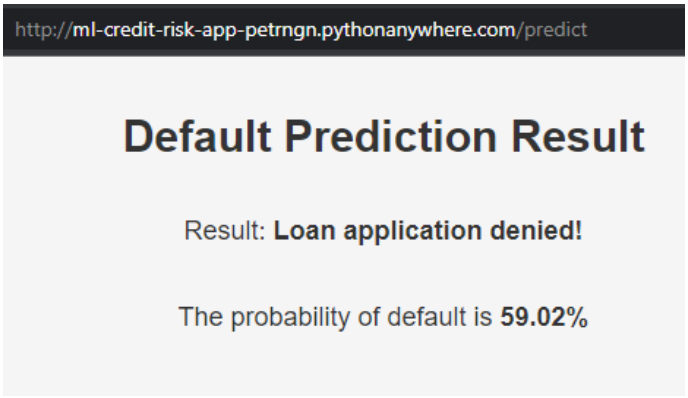
For machine learning deployment in this case, the model is deployed into a web application using Flask and HTML.

Figure 4.19: Flask Web Application Form



Source: Author's results in Python.

Figure 4.20: Flask Web Application - Prediction Result



Source: Author's results in Python.

Chapter 5

Title of Chapter Five

5.1 Frequently made mistakes

The following checklist should help in avoiding some frequently made mistakes, if any of the following propositions apply for your thesis, there is a problem:

- You have citations in your abstract.
- The introduction does not cover the three parts as described in Chapter 1.
- The introduction contains subheadings.
- You described different aspects than promised in the title.
- You copied some parts of the text from other work without proper referencing and citing.
- You used automatic translation tools to produce text by translating it from another language.
- Your thesis contains many typos and grammatical errors. (Use an electronic spell checker. Please!)
- You used color in your figures and refer to the "blue" line (assume that your readers use a monochrome printer).
- You mainly used websites and other unrefereed material as your sources or you used Wikipedia as your source.

- You refer to something in your conclusion which you have not mentioned before.
- Some forenames in the references are abbreviated, some not.
- Some references miss a publishing date.

5.2 Useful Hints

If you write in English, you might find the following hint useful: The indefinite article a is used as an before a vowel sound—for example an apple, an hour, an unusual thing, an **MNC!** (MNC!) (because the acronym is pronounced Em-En-See). Before a consonant sound represented by a vowel letter a is usual—for example a one, a unique thing, a historic chance. Few more tips to follow:

- Don't give orders—don't write in the imperative mood—unless you are training to be a teacher.
- Avoid the use of questions. You may know the answer: does your reader? It's much safer to tell her, or him.
- Do not become entangled in the problems of 'sexist' language. It is much easier to write in the plural. "Students should check their work" is good English. "A student should check—" is also good English, but now the problems begin: "—her work?" "—his work?" Which? You can write "his or her," but that seems clumsy. Stick to the plural.
- If you must refer to yourself, use the third person such as "The present writer would recommend that . . . " may be useful.
- Use the full forms of words and phrases, not contractions like "he's," "don't," etc. Keep the apostrophe to indicate possession—and use it correctly. Academics really sneer at students who use the "Greengrocer's apostrophe."
- Do not despise short, workmanlike, and effective plain English words. If they mean what you want to say. Accurately.

- Avoid the use of humor in academic writing—unless you are very sure of yourself.
- Even when you are not being funny, avoid the use of irony or sarcasm.
- Paragraphs in academic English should contain more than one sentence. (Short paragraphs look as if you are writing for a tabloid newspaper—or a simple Template!) I guess that the average academic book runs to two or three paragraphs per page. Look at the books in your subject, and get a feel for how long your own paragraphs should be when you are imitating the academic style.
- Develop an academic vocabulary. The 'long words' you learn in the course of your studies are long usually because they have more precise meanings than their less formal equivalents. They are therefore better when you want to be accurate. (Also they allow you to sound like someone who deserves a degree.)
- Use as few words as you can; but use enough words to express your meaning as fully as you can. Your judgment of what is appropriate here is part of what you should learn throughout your course.
- Avoid lazy words such as "nice". It is usually better to say "acquire" or "obtain" than "get;" and it may be better, if you mean "through the use of money," to say "purchase" or—better still—"buy."
- A short word like "buy" is better than a long one like "purchase"—unless the long one is more accurate. A "statutory instrument" is better than a "rule"—to a lawyer, at any rate.
- Proof-read with care. Ask someone else to help—you may be too close to your work to be able to see your mistakes.
- If in doubt, choose the more formal, or possibly just the more old-fashioned, of two words. For example, say quotation rather than quote whenever you mean the use of somebody else's words.
- You will often sound more academic if you include doubts in your work—and qualifications. Within the scope of this thesis, the current writer cannot hope to cover all the possible implications of the question.

- In this context, the use of litotes sounds very academic. This is the construction where a writer uses a negative with a negative adjective, e.g. it is not unlikely that ... This does not mean the same as it is probable that ... It has a shade of meaning and qualification that can be useful to academic writers.

Text text Haufler & Wooton (2006).

Text text text text text text text (see, *inter alia*, Haaparanta 1996, pg. 10).

Special Czech, Slovak, and German letters:

ů, á, š, ď, ť, ř, ô, ß, ö

5.3 Itemization and Environments

Many people use simple n-dash in many occasions – like this –, where however typographic convention—it looks a bit strange at first sight—requires m-dash. Text text text text text text Haufler & Wooton (2006).

Text text text text text text Wells *et al.* (2001). Let us describe the following animals:

Item 1 Text.

Item 2 Text.

See what Edmund Burke said about the duties of a Member of Parliament (Speech To The Electors Of Bristol At The Conclusion Of The Poll, November 3, 1774):

> It ought to be the happiness and glory of a representative to live in the strictest union, the closest correspondence, and the most unreserved communication with his constituents. Their wishes ought to have great weight with him; their opinion, high respect; their business, unremitted attention. It is his duty to sacrifice his repose, his pleasures, his satisfactions, to theirs; and above all, ever, and in all cases, to prefer their interest to his own. But his unbiased opinion, his mature judgment, his enlightened conscience, he ought not to sacrifice to you, to any man, or to any set of men living. These he does not derive from your pleasure; no, nor from the law and the constitution. They are a trust from Providence, for the abuse of which he is deeply answerable. Your representative owes you, not his industry only, but his judgment; and he betrays, instead of serving you, if he sacrifices it to your opinion.

Text text text text.

(i) The first item, the first item, the first item, the first item, the first item, the first item, the first item,

(ii) and the second item.

(a) The first item, the first item, the first item, the first item, the first item, the first item, the first item,

(b) and the second item.

TText text text text text text Blomstrom & Kokko (2003).

5.4 Acronyms

Politicians usualy like inward **FDI!** (**FDI!**) and an **MNC!** appreciates **FDI!** subsidies. Are **MNC!**s greedy?

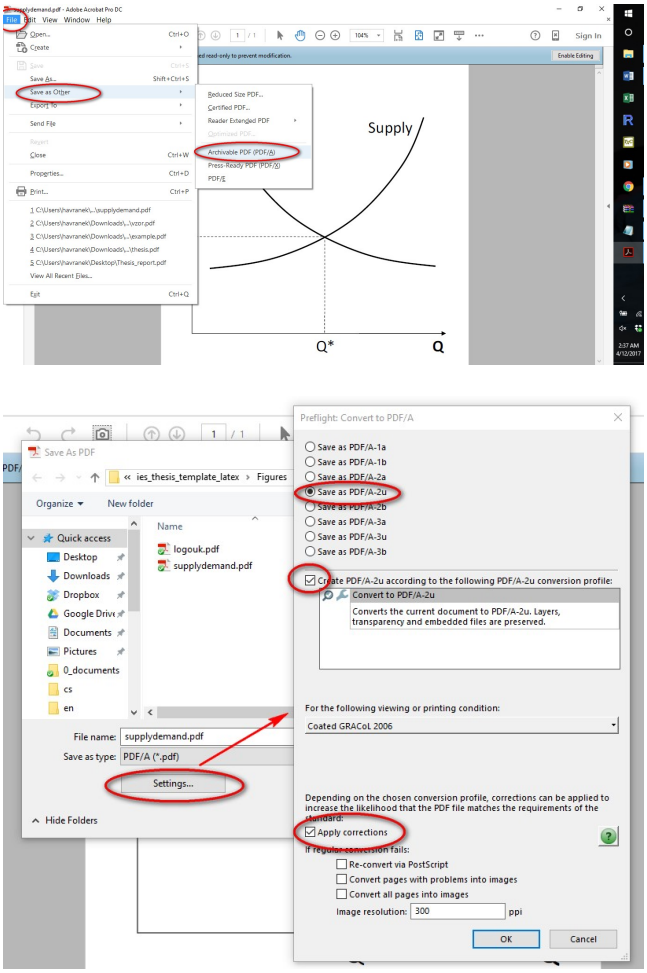
5.5 Figures

To achieve compatibility with PDF/A 2u, your file must not include links to external fonts, audio, video, or scripts. On the other hand, your file must declare each color environment you use, it must include all the pictures/figures either in jpeg or PDF/A 2u format, used fonts compliant under Unicode (your file cannot use any external fonts), and it must include meta-data in XMP format.

Most troubleshooting comes from the conversion of figures to compliant formats. You can convert from simple PDF using Adobe Acrobat:

- Select File » Save as Other » Archivable PDF (PDF/A)
- Save as PDF/A-2u:

But most of the vector graphics gets distorted to lower quality in Adobe (like pictures in pdfs generated from Stata, unless jpeg is sufficient for you). You



can also use GhostScript, the conversion tool is provided by courtesy of the Faculty of Mathematics and Physics at

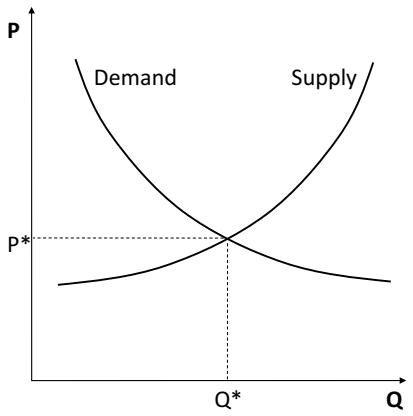
https://kam.mff.cuni.cz/pdfix/

Text text text text text text.[1] Font of Latin phrases should be consistent: Furthermore, there is no *ex post* price effect, all things being equal (*ceteris paribus*). This is *per se* truth.

Look at the Figure 5.1. Text text text text text text text text text text.

[1]Text text text text text text text text text text text text text text text text. Text text text text text text text text text text.Text text text text text text.

Figure 5.1: Market equilibrium



Source: Haufler & Wooton (2006).

5.6 Tables

If you use Stata, you might want to check the `sutex`, `outtable`, `outtex`, and `estout` tools, which help you with exporting Stata tables to LaTeX.

Table 5.1: Model's predictions

Case	Y_1	Y_2	τ_1	τ_2	a	n
CR—Slovakia	10.9	10	0.24	0.19	1,000	2.16
CR—Poland	13.3	12	0.24	0.19	1,000	0.38
CR—Hungary	10.4	8	0.24	0.16	1,000	1.10

Source: If the source is author himself (like a calculation output), this line is redundant.

Text text text text text text text text text text text text text text text text.

5.7 Boxes

Text text text text text text text text text text text text text text text text. Let us make a box:

Text text text text text text text text text text text text text text text text.

5.8 Theorems, Definitions, . . .

Definition 5.1 (My original definition). This is a definition.

Figure 5.2: Boxy's example

- Welcome to Boxy paragraph. We sincerely hope you will all enjoy the show.
- Welcome to Boxy paragraph. We sincerely hope you will all enjoy the show.
- Welcome to Boxy paragraph. We sincerely hope you will all enjoy the show.

Source: Haaparanta (1996)

Assumption 5.1 (My realistic assumption). This is an assumption.

Proposition 5.1 (My clever proposition). *This is a proposition.*

Lemma 5.1 (My useful lemma). *This is a lemma.*

Example 5.1. This is an example.

Proof. This is a proof. □

5.9 Nonumbered Equations

$$U = \underbrace{\int_0^\infty \frac{1}{1-\sigma}\left(C^{1-\sigma}-1\right)e^{-\rho t}\,\mathrm{d}t}_{\text{meaning of life}}$$

5.10 Numbered Equations

$$U = \int_0^\infty \overbrace{\frac{1}{1-\sigma}\left(C^{1-\sigma}-1\right)}^{\text{instantaneous utility}} e^{-\rho t}\,\mathrm{d}t \tag{5.1}$$

5.11 Matrix Equations

$$\boldsymbol{A} = \boldsymbol{B} + \boldsymbol{C} \tag{5.2}$$

5.12 Cross-references

- to literature (Bjorvatn & Eckel 2006, pg. 10) or Haufler & Wooton (2006, pg. 10),
- to Figure 5.1,
- see Table 5.1,
- to **??**,
- to Definition 5.1, to Proposition 5.1, Example 5.1,
- to equations like this: see (5.1).

5.13 Source codes

You can input a source code like this:

```
omega = 1;
syms zeta;
jmn = [1 2*zeta*omega omega^2];
figure(1);
    for zeta = 1E-5 : 0.2 : 1+1E-12
        G = tf(omega^2,subs([1 2*zeta*omega omega^2]));
        bode(G); hold on;
    end
legend('\zeta = 0','\zeta = 0,2','\zeta = 0,4','\zeta = 0,6',');
```

Should you prefer a different font size, redefine file `Styles/Mystyle.sty`.

5.14 Paragraphs

Usually you should not use the first person singular (I) in your text, write we instead. As a general recommendation, use the first person sparsely, sometimes it can be replaced by a phrase like "This work presents ...."

Text text text text text text (Haufler & Wooton 2006). Let us make two paragraphs:

Proin Text text text text text text text text text text text text text text text. Text text text text text text. And a subparagraph:

Velit Text text text text text text text text text text text text text text text text.

Chapter 6

Conclusion

The conclusion should briefly summarize the problem statement and the general content of the work and the emphasize on the main contribution of the work.

When writing the conclusion keep in mind that some readers may not have gone through the whole thesis, but have jumped directly to the conclusion after having read the abstract in order the decide on the personal relevance of the thesis. Therefore, the conclusion should be self contained, which means that a reader should be able to understand the essence of the conclusion without having to read the whole thesis.

The conclusion typically ends with an outlook that describes possible extensions of the presented approaches and of planned future work.

Bibliography

Bjorvatn, K. & C. Eckel (2006): "Policy Competition for Foreign Direct Investment Between Asymmetric Countries." *European Economic Review* **50(7)**: pp. 1891–1907.

Blomstrom, M. & A. Kokko (2003): "The Economics of Foreign Direct Investment Incentives." *NBER Working Papers 9489*, National Bureau of Economic Research, Inc.

Haaparanta, P. (1996): "Competition for Foreign Direct Investment." *Journal of Public Economics* **63(1)**: pp. 141–53.

Haufler, A. & I. Wooton (2006): "The Effects of Regional Tax and Subsidy Coordination on Foreign Direct Investment." *European Economic Review* **50(2)**: pp. 285–305.

He, H., Y. Bai, E. A. Garcia, & S. Li (2008): "Adasyn: Adaptive synthetic sampling approach for imbalanced learning." In "2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)," pp. 1322–1328.

Igareta, A. (2021): "Stratified sampling: You may have been splitting your dataset all wrong."

Kornbrot, D. (2014): "Point biserial correlation." *Wiley StatsRef: Statistics Reference Online* .

Scikit-learn (n.d.): "Sequentialfeatureselector." `https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SequentialFeatureSelector.html`.

Wells, L. T., N. Allen, J. Morisset, & N. Pirnia (2001): *Using Tax Incentives to Compete for Foreign Investment: Are They Worth the Cost?* Washington, DC: FIAS.

Wendler, T. & S. Gröttrup (2021): *Data Mining with SPSS Modeler: Theory, Exercises and Solutions*. Cham: Springer.

Appendix A

Title of Appendix A

Text text text text text text text text text text text text text text text. Text text text text text text text text text text. Text text text text text text text text text text text text text text text text text text text text.

Appendix B

Project's website

You can create a special website for your project which contains empirical data and MatLab/R/Stata source codes, see meta-analysis.cz/sigma, for example. Stating in your thesis that the data and source codes are available upon request is enough but please, have them prepared for such requests. The faculty does not allow enclosed DVD.

- File 1: Master's thesis
- File 2: Empirical data
- File 3: Source codes