Ca’ Foscari University of Venice

Department of Economics

Master’s Degree in Data Analytics for Business and Society

Final Thesis

Modeling Default Probability and Energy Efficiency in Italian Residential Buildings

Graduand
Pierfrancesco Montello
887309

Supervisors
Prof. Michele Costola
Prof. Roberto Casarin

Academic Year 2021-2022

SUMMARY

INTRODUCTION

The concern of energy efficiency has always existed in a latent form (sometimes disguised even as resource saving), but awareness of this issue has occurred only in the 19th century, the 20th century helped to insert into the collective mentality through media, standardization and regulation and 21st century is found in the conjuncture of widespread concerns in many areas.

Starting from about 5500 B.C. in Dacia (Romania), where people used to have “bordei” or “coliba” houses, partially or totally built into the ground to keep a constant indoor temperature during the year, crossing through about 500 B.C. in Greece, where houses were oriented South (Socratic Houses) and through the 1500s in Italy, as Leonardo Da Vinci was building the first mechanical indoor air cooler, arriving to 2004 in Germany, where is built the solar city of Solarsiedlung am Schlierberg, which is a self-sustaining city projected by the architect Rolph Disch (Ionescu *et al.*, 2015).

Nowadays, buildings’ energy efficiency (EE) is also one of the main directions to which the EU pushes to reduce CO_2 emissions and fossil fuel consumption to mitigate their impact on climate change in the next years. EU has set the target to reach -60% emissions in 2030 compared to 2015 and to achieve climate neutrality by 2050[1].

According to the European Commission[2] in fact, buildings account for 36% of the total carbon dioxide emissions and are the single largest energy consumer in Europe (40% of the total), of which 80% comes from heating, cooling and domestic hot water. Currently, 3 out of 4 buildings in Europe are not efficient and about 35% are over 50 years old, while only somewhere between 0.4% and 1.2% are renovated each year (renovation can reduce the emissions by 5%). At this rate, it is estimated that 75-90% of the old building stock will be still standing by 2050, failing the target established for that year (Economidou *et al.*, 2019).

Moreover, energy efficient buildings are involved in some of the largest economic sectors: the construction of buildings (that accounts for the 9% of European GDP), the investments (since making a building energy efficient increases its value) and the

[1] For further details, see: https://ec.europa.eu/clima/news-your-voice/news/delivering-european-green-deal-2021-07-14_en.

[2] For further details, check: https://energy.ec.europa.eu/topics/energy-efficiency/energy-efficient-buildings/energy-performance-buildings-directive_en.

mortgage market. Specifically, in the recent years, the scientific literature studied the main drivers to mortgages default, assuming that buildings' energy efficiency may be one of them (in particular, Kaza *et al.* (2014) and An and Pivo (2020) in the US market, Billio *et al.* (2021) in the Dutch market, Billio *et al.* (2022) in Italian market, while Zancanella *et al.* (2018) adopted a more general perspective).

Those analysis identified three main channels through which energy efficiency may impact the mortgages' default risk:

i) personal characteristics of the borrowers captured by the choice of an EE building (e.g., environmental consciousness);

ii) improvements in building performance that help free up a borrower's disposable income through lower utility bills;

iii) the positive effect on the dwelling value and thus, on the loan-to-value ratio (LTV).

In addition to that, a work by Burt *et al*. (2010) argues that house EPC ratings (the European main EE classification) can accurately predict annual energy costs, which should translate into lower default risk. This work has the objective of expanding the work of Billio *et al.* (2022) to corroborate and/or undermine their findings, by operating on the same dataset and applying different or more sophisticated statistical tools to model the default risk with special consideration for the relationship between buildings' EE and default.

The first chapter will perform an exploratory data analysis that aims to investigate the structure of the variables and their interactions among them and with the target (*default*).

The second chapter will expand the type of models and approaches used in previous works and has to find, as main objective, if there's a better performing model than the logistic one, in that case what it is, and studying which variables are more relevant to the default risk prediction.

Finally, the third chapter wants to dig deeper into the relationship between the energy efficient component and the probability of default, and try to define it clearly proposing an additional approach compared to the one used in Billio *et al*. (2022).

Chapter I – Exploratory data analysis

In this chapter we proceed with a brief explanation of the data cleaning process, then we move on the analysis of the plots associated with the different variables, we will check the graphs that show the relationship of variables pairs and, finally, we will analyze the dependence or independence structure that we encounter using the Pearson's chi-squared test on couples of categorical variables.

The dataset is the same used in Billio *et al.* (2022). It contains data from mortgages starting from January 2010 and the last "observation" for the ones still "alive" was recorded the 31st December 2019. The dataset is obtained from financial institutions and composed by three main types of data:

i) characteristics of the mortgage and/or of the building itself (e.g., property value, loan amount, the energy class (in form of 7 EPC classes, from A to G), construction year, property status, etc.);

ii) information about the person who requested the loan, the borrower (e.g., age when requesting the loan, credit score value, residence region, etc.);

iii) some key macroeconomics variables (e.g., HPI, GDP, unemployment rate, inflation rate, etc.).

The initial dataset includes 104 472 observations and 107 variables.

First, some data cleaning (van der Loo and de Jonge, 2018) had to be made by eliminating duplicated, redundant, or useless variables, then merging some others or obtaining some useful information by combining them. In the process, some observations had to be discarded as they lacked some essential data (e.g., the end date of the mortgage, the loan-to-value proportion or the age of the borrower); however they are roughly the 3% of the total number of observations. The *score value* and *HPI change rate* variables were a different story since they were lacking data for, respectively, 30% and 8% of all the observations. The imputation method used to deal with that problem was to substitute the missing values with the mean of the variable. After the process, we ended up with a dataset composed of 101 152 observations and (only) 26 variables that are:

- ***default****:* it is a binary variable that identifies if the loan default or not and the target variable of the analysis. A loan is considered defaulted if there are arrears older than 90 days that haven't been paid yet;
- *ID*: it is a variable that works as a unique key to identify differently each and every different mortgage;

- *property_value*: it is a numeric variable that represents the value that the financial institution estimated for the associated dwelling;
- *date_contract_begin*: it is a date variable that records the date from which the loan starts;
- *date_contract_end*: it is a date variable that record the date which the loan is supposed to end in;
- *loan_length*: it is a numeric variable that represents the supposed duration of the loan in years;
- *no_of_instalments*: it is a numeric variable that represents the number of instalments in which the borrower has to repay the loan plus the interest;
- *periodicity*: it is a categorical variable that records the frequency of the instalments. It has only three values: M that stands for monthly instalment, T that stands for quarterly instalment and S that stands for half-yearly instalment;
- *loan_amount*: it is a numeric variable that represents the amount of money that the financial institution lent to the borrower;
- *ltv*: (stands for "loan-to-value" ratio) it is a numerical variable that comes out from the fraction: *loan _amount* divided by *property_value*. It represents the fraction of money that the financial institution lent to the borrower with respect to the estimated value of the property;
- *epc*: (stands for "energy performance certificate") it is a categorical variable that represents the energy efficiency class to which the building belongs to. The rating of the EPC is based on strict rules and depends on: the amount of energy consumed per m^2 and the level of carbon dioxide emissions, in tonnes per year. It is codified, in alphabetical order, with the letters from A to G, with the letter A identifying the "greener" buildings while the letter G identifies the ones which pollute the most;
- *property_region*: it is a categorical variable that identifies the Italian Region which the building belongs to;
- *residence_region*: it is a categorical variable that identifies the Italian Region where the borrower lives in;
- *age_borrower_orig*: it is a numerical variable that represents the age of the borrower when she got the loan;

- *construction_year*: it is a numerical variable that represents the year in which the building was built (if it took more than one year, the last year is considered);
- *property_status*: it is a categorical variable that represents the status in which the building is, i. e. if it is new (NUOV), it is renovated (RISTR), has to be renovated (DARIS), it is almost new (SNUOV) or used (USAT);
- *score_value*: it is a numerical variable that represents the credit score of the borrower. This variable presented about 30% of missing values, mostly regarding old loans, that have imputed with the mean;
- *cadastral_category*: it is a categorical variable that identifies more in-depth the type of dwelling. It has five categories: "appartamento" identifies an apartment, "attico/mansarda" identifies an attic, "loft" identifies a loft, "villa/villino" identifies either a manor or a cottage and "villetta a schiera" identifies a townhouse;
- *perf_default_date*: it is a date variable that records either the date of default of the loan or the day of the last observation (31st December 2019) for loans still alive;
- *region_macroarea*: initially labelled as "NUTS1_region" identifies the division of the EU adopted by the Eurostat. It is a 5-categories categorical variable that in this case applies only to Italy. Its categories are: "North-West", "North-East", "Centre", "South" and "Islands";
- *HPI*: (that stands for "House Price Index") is a broad measure of the movement of single-family property prices in the US. In our case, it is a numerical variable that indicates the HPI value.
- *HPI_chng*: is a numerical variable that indicates the difference between the current (referred to time in which the loan started) and the last measure of HPI;
- *GDP*: (stands for "Gross Domestic Product") is a broad measure of the wealth produced by a Country in a certain period of time. In our case, it is a numerical variable that records the GDP values;
- *inflation*: it is a numerical variable that, as the name suggests, keeps track of the inflation values;

- *mtgrt*: it is a numerical variable that keeps track of the interest rate applied to each mortgage;
- *unemployment*: it is a numerical variable that, as the name suggests, keeps track of the unemployment rate.

1.1 Exploration through graphs and plots

In this section we focus on the exploration of the variables' structures and distribution in the whole dataset. Once we have finished looking at each variable alone, we will pair some of them and checking if there are meaningful distribution differences when organized by other characteristics' classes.

Since many of the meaningful variables are categorical, and the target is binary, it was decided to start investigating their structure and the relationship between them and the *default* one through some graphical representations.

1.1.1 Variable distributions

The first variable to check is obviously the target: the binary variable named as *default* that keeps the record of whether the borrower repaid the loan or not.

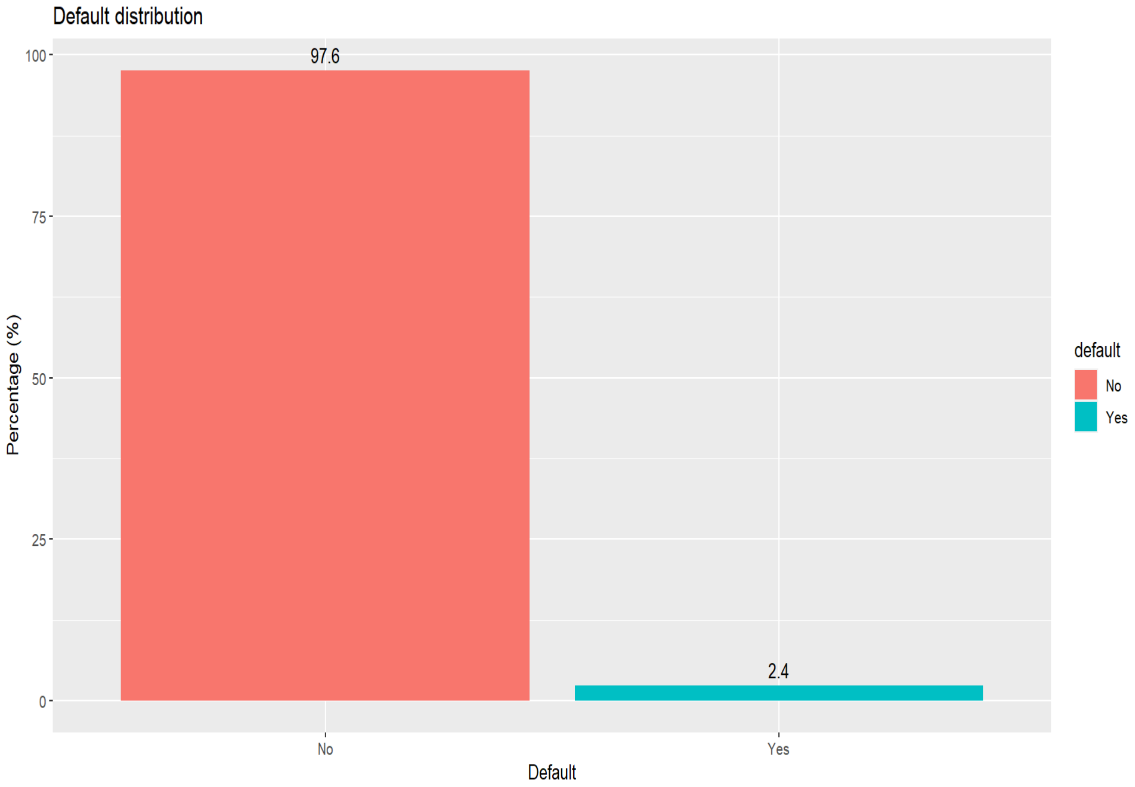

Figure 1 – This figure presents the proportion of defaulted and non-defaulted loans in the entire dataset.

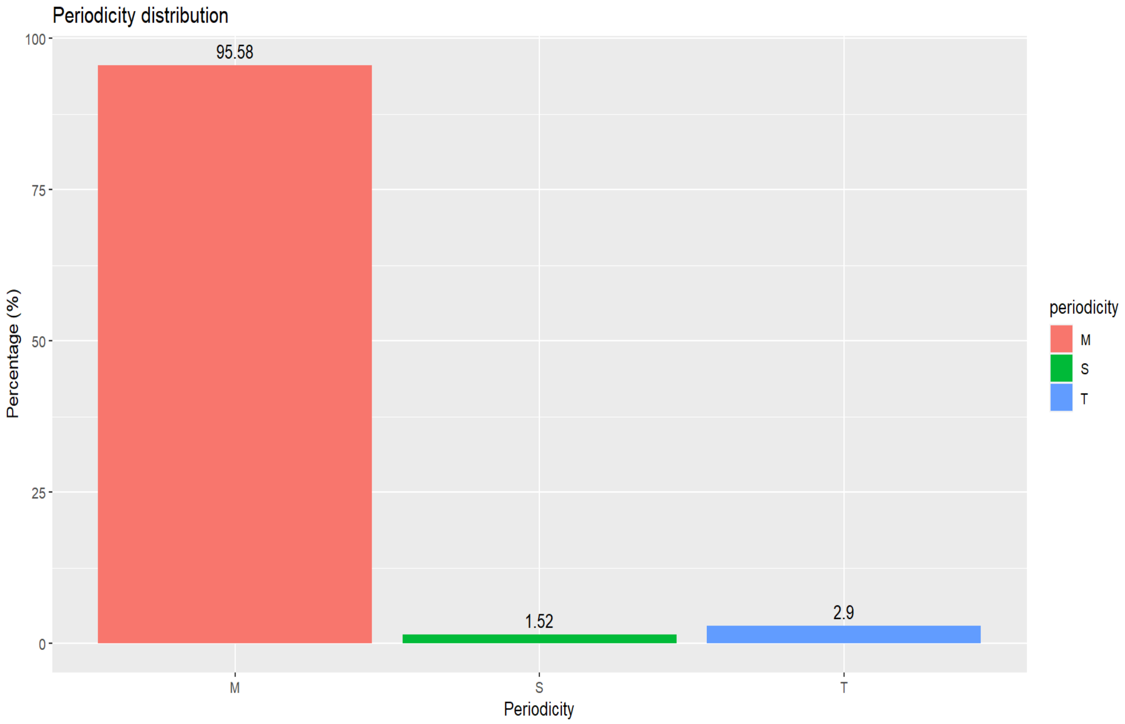

***Figure 2* –** This figure presents the proportion of different type of periodicity for the loans in the dataset. M stands for monthly, S for half-yearly and T for quarterly payment.

In Figure 1, we can clearly see the imbalance between the two classes of the *default* variable: the repaid loans account for the 97.6% of the total ones, while only 2.4% defaulted. Next, we wanted to inspect the different periodicity of the loans.

As we can spot in Figure 2, once more, there's a clear imbalance in the classes: almost 96% of the loans has to pay an instalment every month (M), almost 3% every quarter (T) and 1.52% only twice a year (S). Then, we move on to the, arguably, most important independent variable.

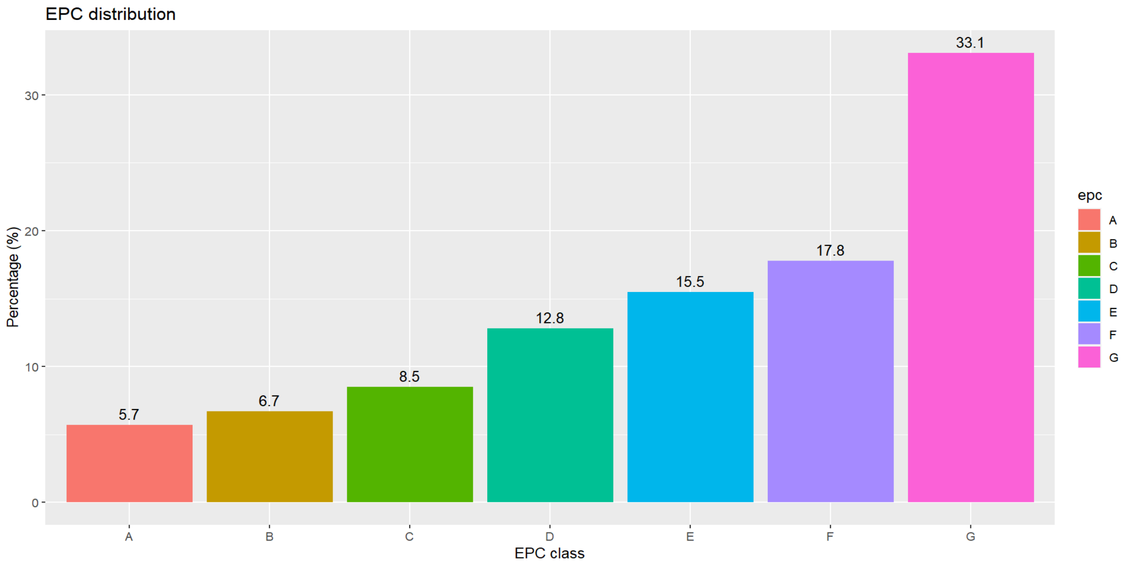

Figure 3 – This figure presents the percentage, in the entire dataset, of buildings belonging to each class of energy efficiency. The EPC rating define 7 classes: from A (best) to G (worst energy efficiency).

From Figure 3, we can tell that, the higher the energy efficiency class, the less it is present in the dataset, ranging from a little more than 1 out of 20 for A class buildings to almost 1 out of 3 for G class buildings.

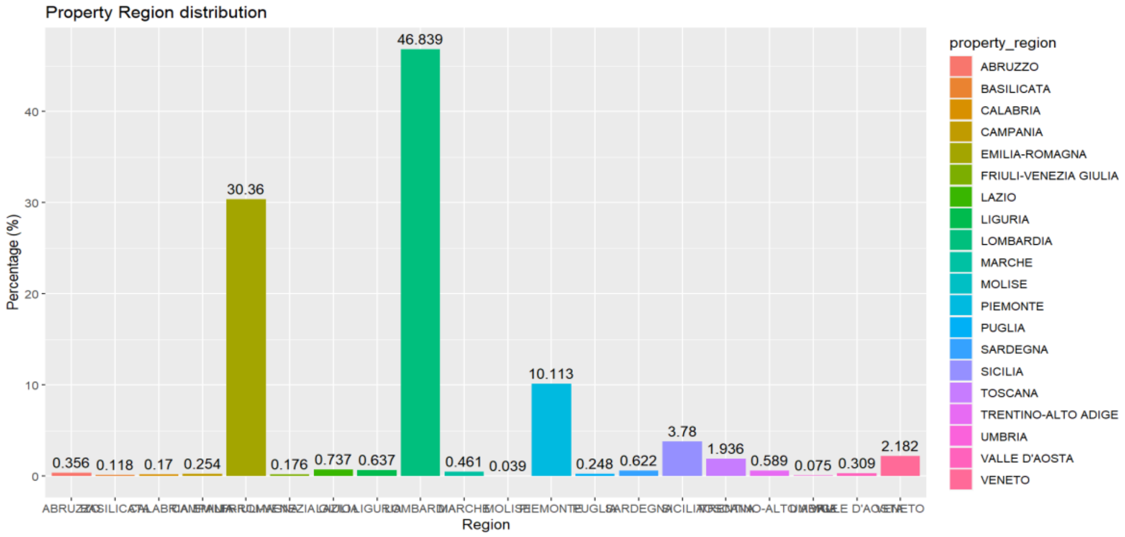

Figure 4 – The figure presents the percentage of buildings in the dataset belonging to each region.

We're also interested in getting a picture of the geographical distribution of the buildings.

In Figure 4, the geographical distribution is clearly uneven among the Italian regions: we've almost half of the buildings located in Lombardy, almost a third in Emilia Romagna and, in the third, spot Piedmont with about 1 building out of 10. After, the geographical distribution of the buildings we are interested in how are geographically distributed the owners of those buildings.

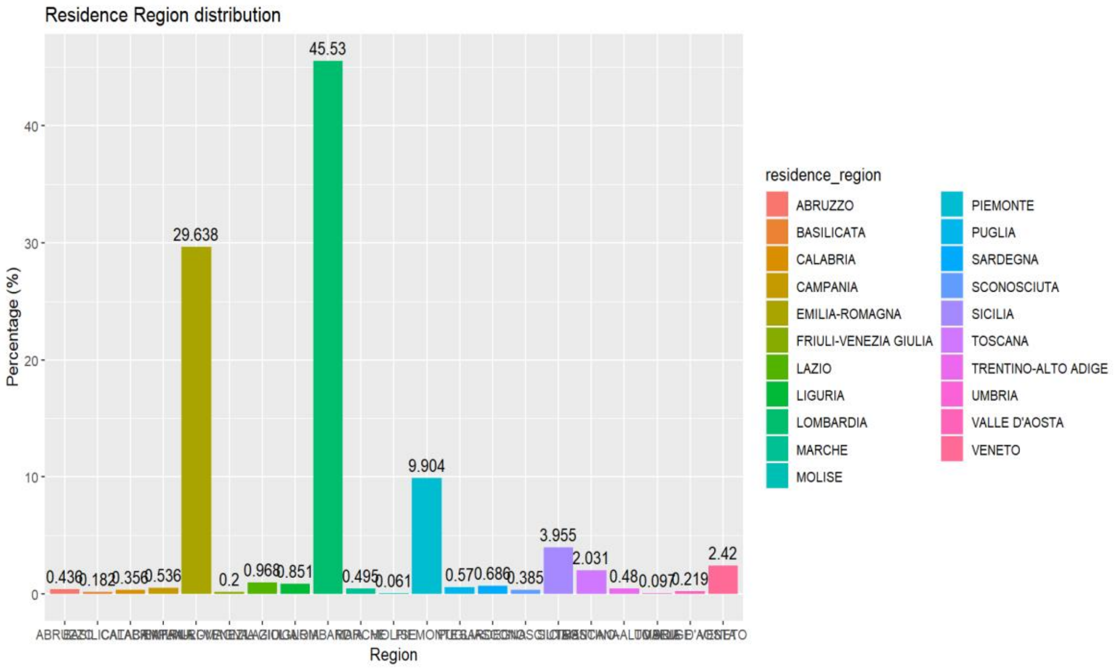

Figure 5 – The figure presents the geographical distribution (the residence) in percentage of the owners of the buildings in the dataset.

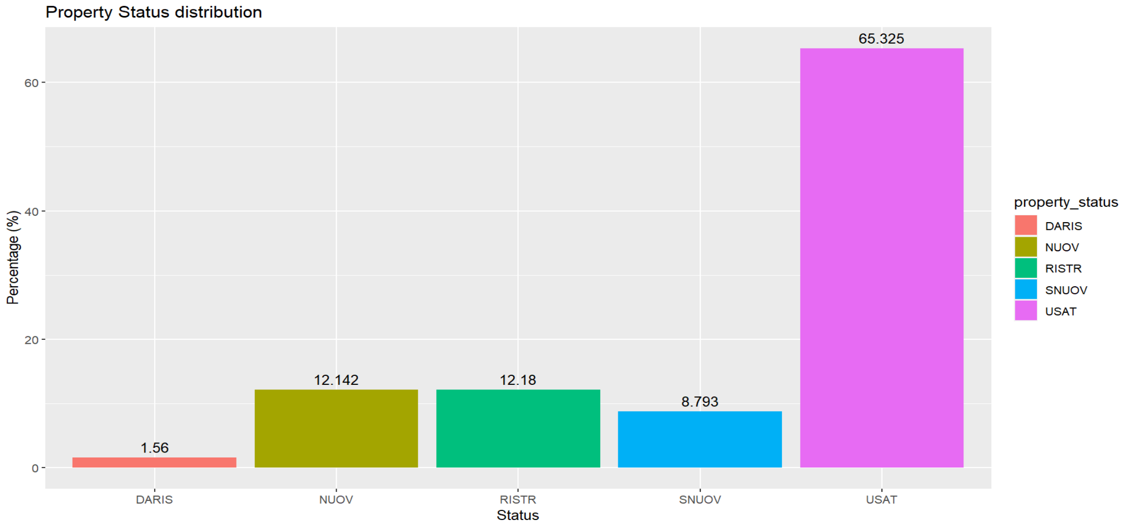

Figure 6 – The figure presents the percentage distribution of the property status of the buildings in the dataset.

The plot, in Figure 5, reflects mostly the previous one telling us that the distribution of the owners mirrors, more or less, the distribution of the buildings, in the Country. Together with the geographical distribution of buildings and owners, we're also interested in understanding the status of the properties.

We see that more than 65% of the buildings are used/in-use, about 24% are new or renovated and the remainder are almost new or to be renovated (Fig. 6). Another one of the variables to explore is the cadastral category of the buildings.

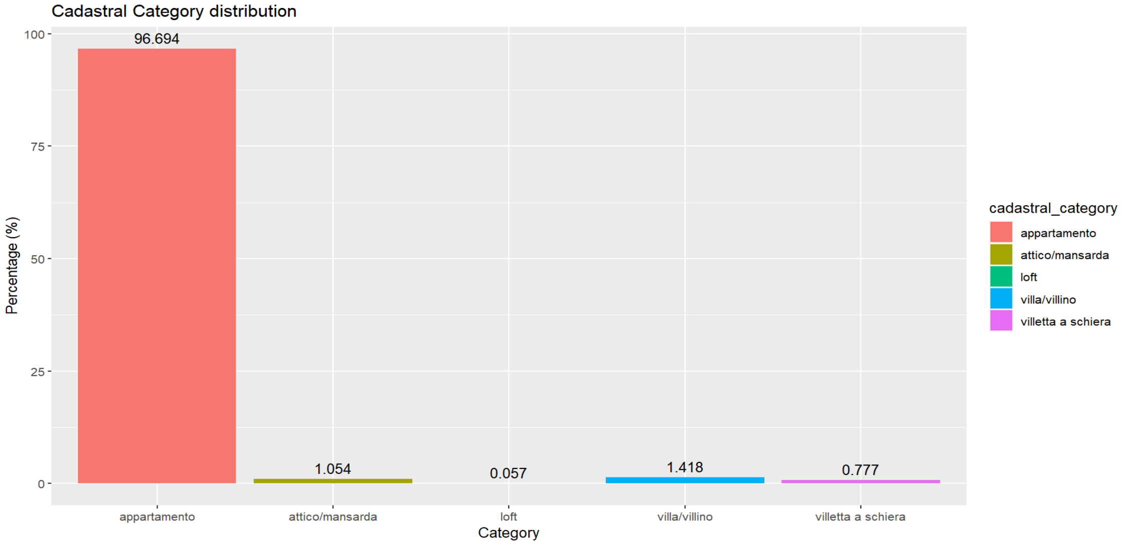

Figure 7 – This figure presents the percentage distribution of the buildings in the dataset by cadastral category.

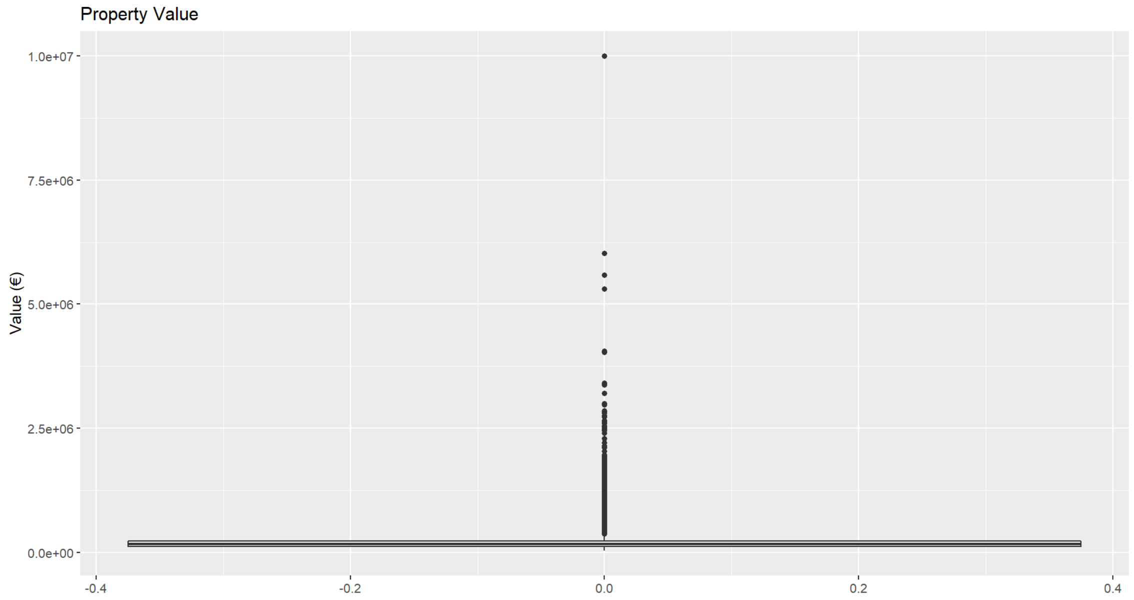

Figure 8 – This figure presents the box plot that represents the distribution of the values for the "Property Value" variable.

From the plot in Figure 7, it's crystal clear that almost every loan was requested to buy an apartment (just 3.3% were needed to buy a different type of housing). Despite not being a categorical variable, we could also plot the distribution of values for the *property value* variable.

Unluckily the 5807 outliers (all the property evaluated above € 374 000), flatten the important part of the plot, in Figure 8. However, we can say that 25% of the properties are worth less than € 124 000, 50% are between € 124 000 and € 224 000 and the remainder are worth more. The mean value is € 193 098, while the minimum is € 32 000, and the maximum is € 10 000 000. The amount of money lent is highly important, too.

As in the previous plot, in Figure 9, we have some outliers (4019) that flatten the most important part of the plot. We can safely say that 25% of the amounts lent is below € 73 130, 50% are between that value and € 140 000 and the 25% that remains is higher. The mean value is € 116 796, while the minimum is € 30 009, and the maximum is € 7 000 000. Combining the previous two variables we can obtain one of the main predictors of default probability: the loan-to-value ratio.

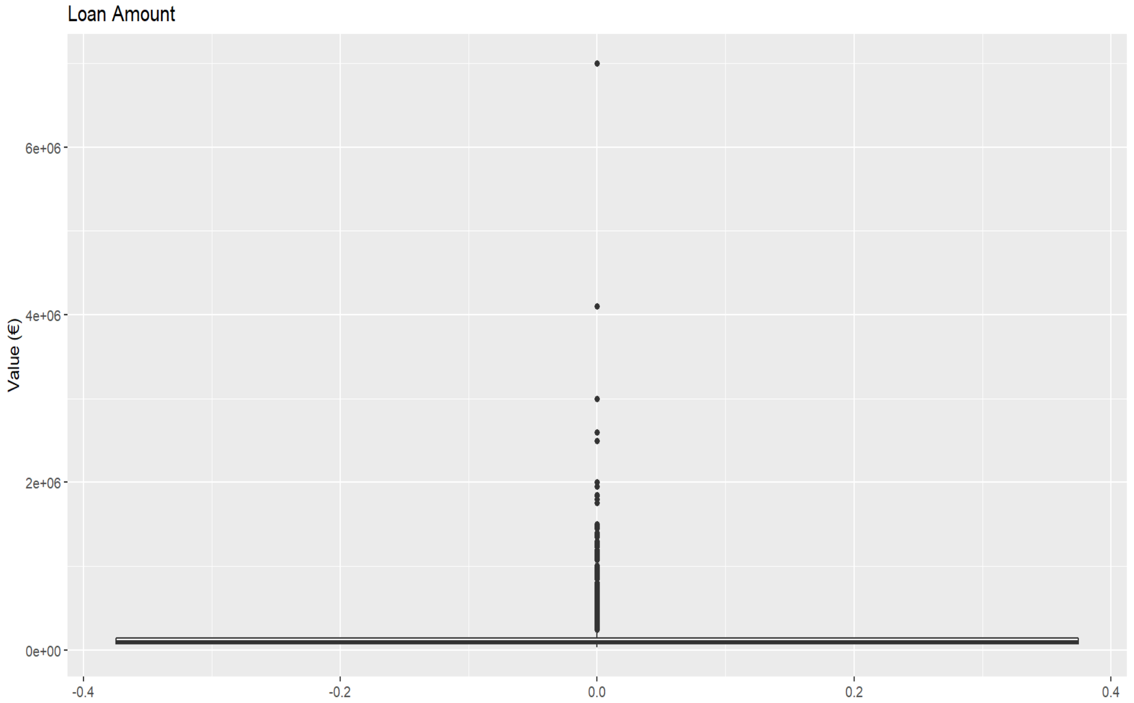

Figure 9 – This figure presents the box plot that represents the distribution of the values of the "loan amount" variable.

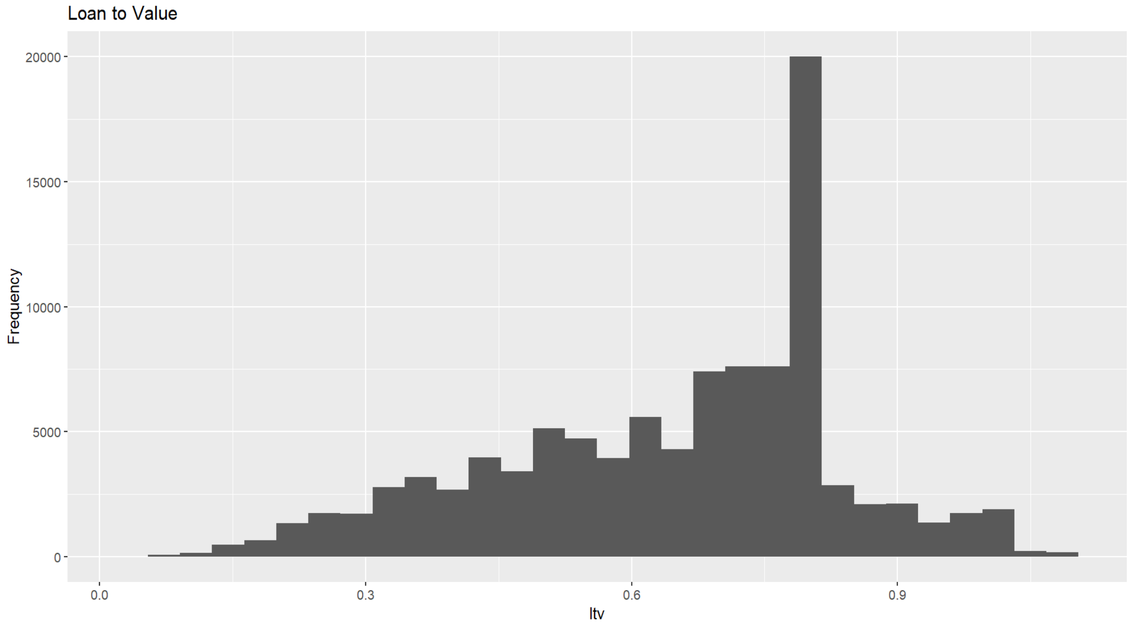

Figure 10 – This figure presents the histogram of distribution of the values of "loan-to-value" variable. The height of the bars represents the absolute frequency of the associated values.

Even this time (Fig. 10), there are some outliers (670 loan-to-value ratios higher than 1), but they don't have much impact on the plot. We can say that 25% of the ratios are lower than 0.51, 50% of the total are between 0.51 and 0.79, and the remainder 25% is made of higher values. The mean is about 0.65, the minimum 0.04 and the maximum 1.09 (even though some loans exceed the value of the building, we can keep them). We want to investigate one more variable: the age of the borrower when she requested the loan.

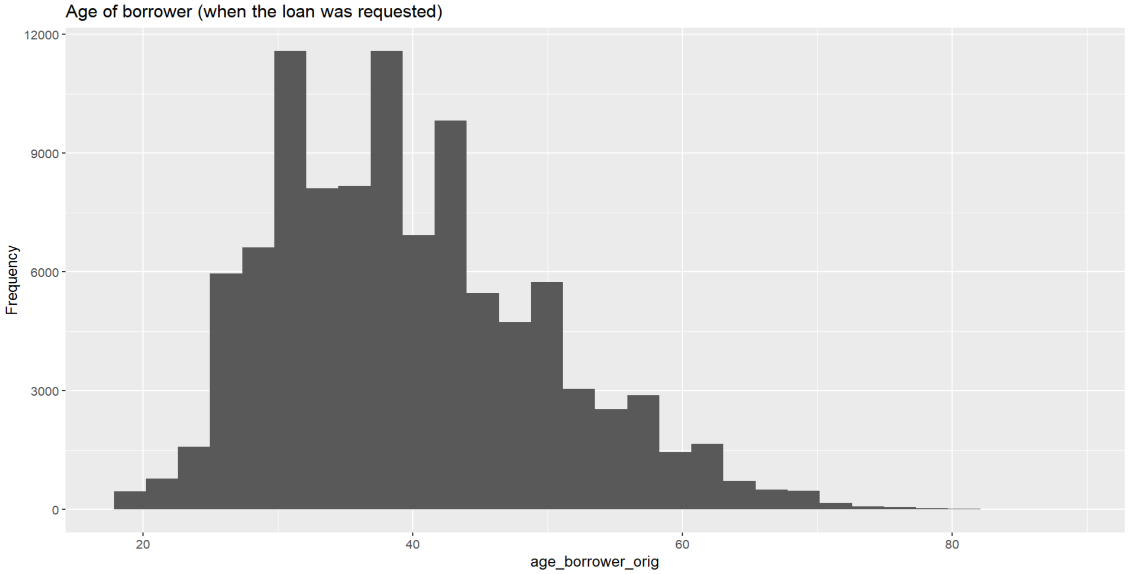

Figure 11 – The figure presents the histogram of the distribution of the values of *Age of borrower* variable. The age of the borrower taken in consideration is the one she had when the loan was requested. The height of the bars represents the absolute frequency of the associated values.

In Figure 11, half of the borrowers had between 32 and 46 years when they asked for the loan. The total mean is about 40 years, but we have some extreme values like 18 or 87 years. The construction year of the building is worth investigating as well.

As long as Figure 12 is concerned, despite having some buildings dating back to the first year of the 1900s, 50% of the housing was built between 1964 and 2005. The average year of construction is 1980, but there are loans for buildings built in 2019, too.

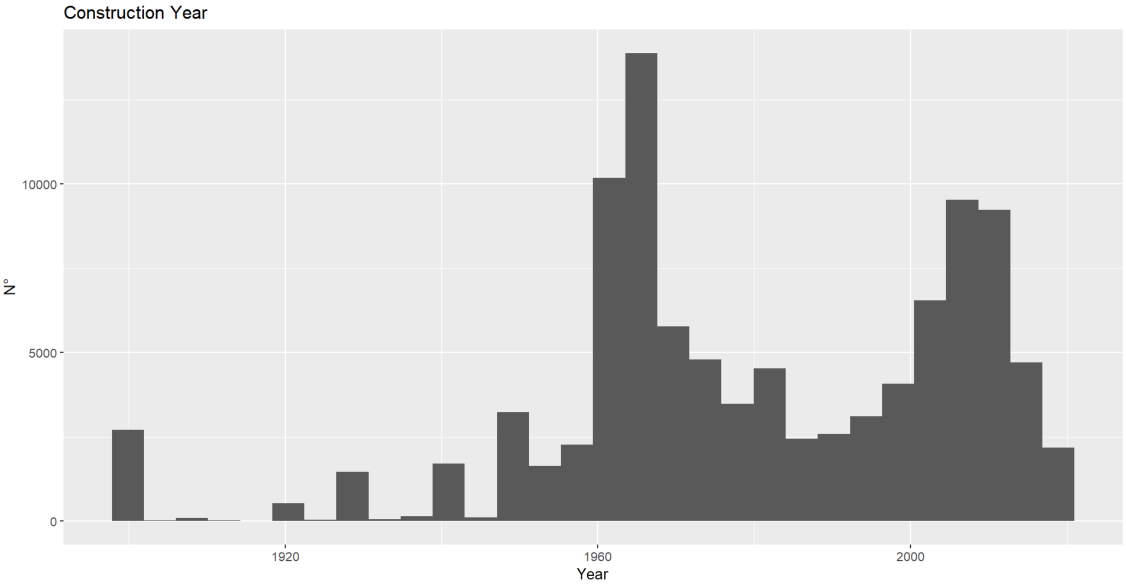

Figure 12 – The figure presents the histogram of frequency of the variable *construction_year*. The height of the bars represents the absolute frequency of the associated values.

In addition, from Figure 13, we have the variable *score_value* distribution, even though we had to perform some imputation (in 30% of the values). The imputation may have had a great impact on the distribution of this variable since the interquartile range[3] is only 35 (from 503 to 538), the median and the mean overlap at 516.1, even though the range of the variables goes from 167 to 598. The loan length is also very important.

As we can see from the plot (Fig. 14), the most common loan lengths are multiple of five and lay between 10 and 30 years, in fact 50% of loan have a duration between 15 and 25 years. The mean is almost 19 years, the maximum 50 years and the minimum one day (may be just an error in the data considering that has also 121 instalments and € 31 000 to repay). Lastly, we check the default time.

[3] The distance between the first and the third quartile of the distribution, it identifies the range in which fall 50% of the values.

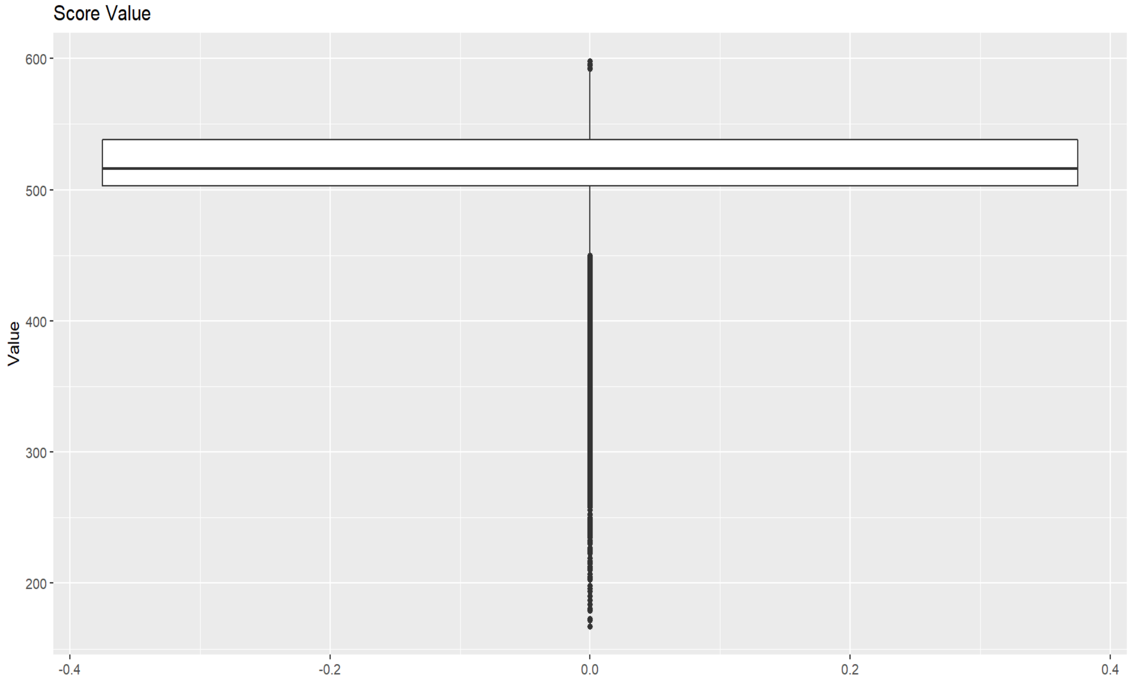

Figure 13 – The figure presents the box plot of the distribution of the *score_value* variable.

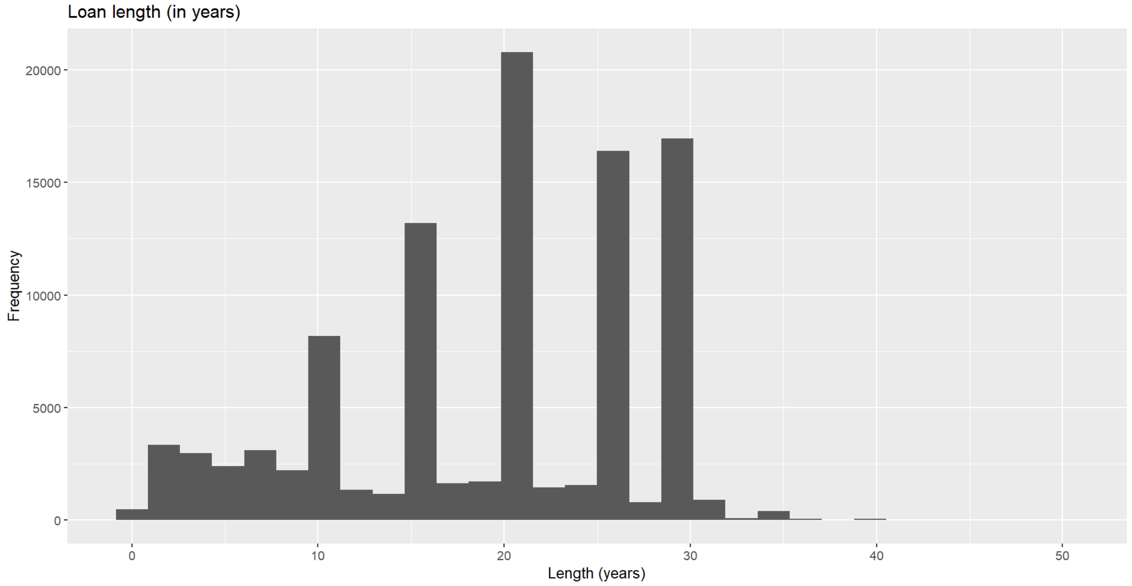

Figure 14 – The figure presents the histogram of the distribution of the values of the variable *Loan length*. That variable indicates the loan duration expressed in years. The height of the bars represents the absolute frequency of the associated values.

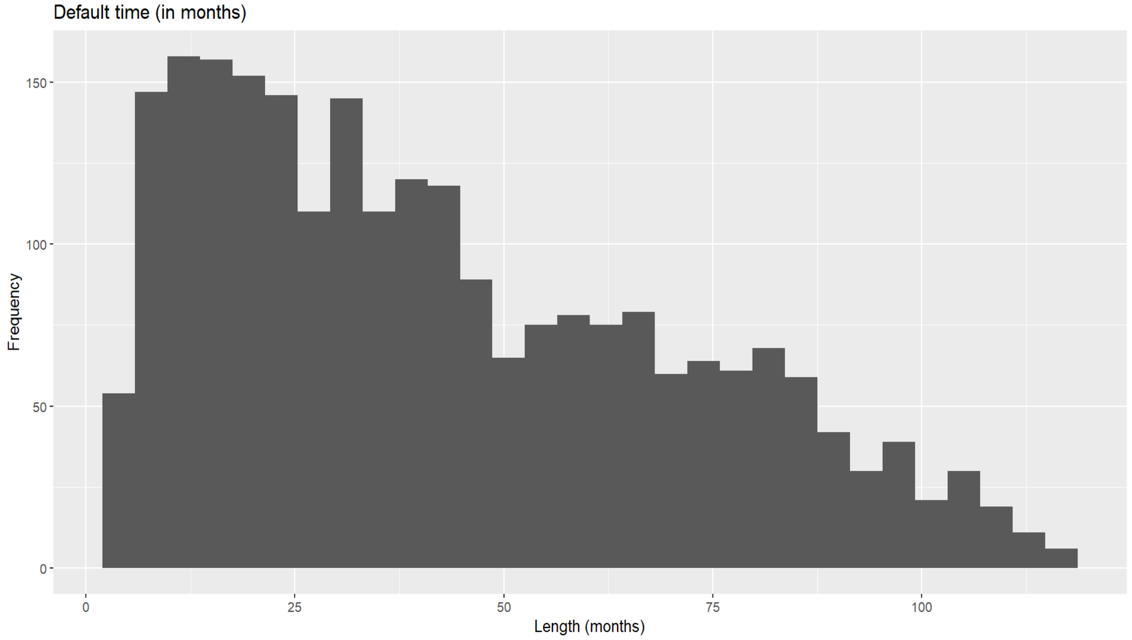

Figure 15 – The figure presents the histogram of distribution of the variable *Default time*. That variable indicates the time elapsed between the start of the loan and its default, expressed in months. The height of the bars represents the absolute frequency of the associated values.

In Figure 15, we see that 50% of times the default occurred between 20 and 64 months; the average default time is 43 months, but the range is from 5 to 118 months (almost 10 years).

1.1.2 Variable combination and comparisons

Now we move on by giving a look to the plots of paired variables. To address one of our main points we compare the default rates of each energy efficiency class.

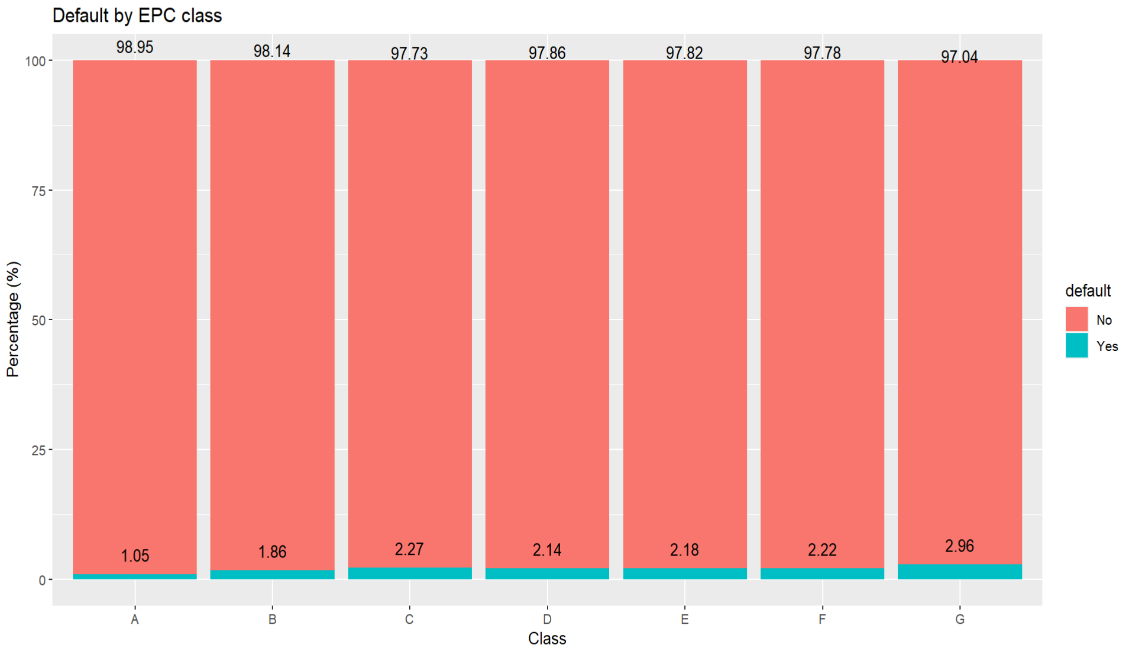

Figure 16 – This figure presents default/non-default percentage of the loans in the dataset organized by energy efficiency classes (A is the best, G the worst).

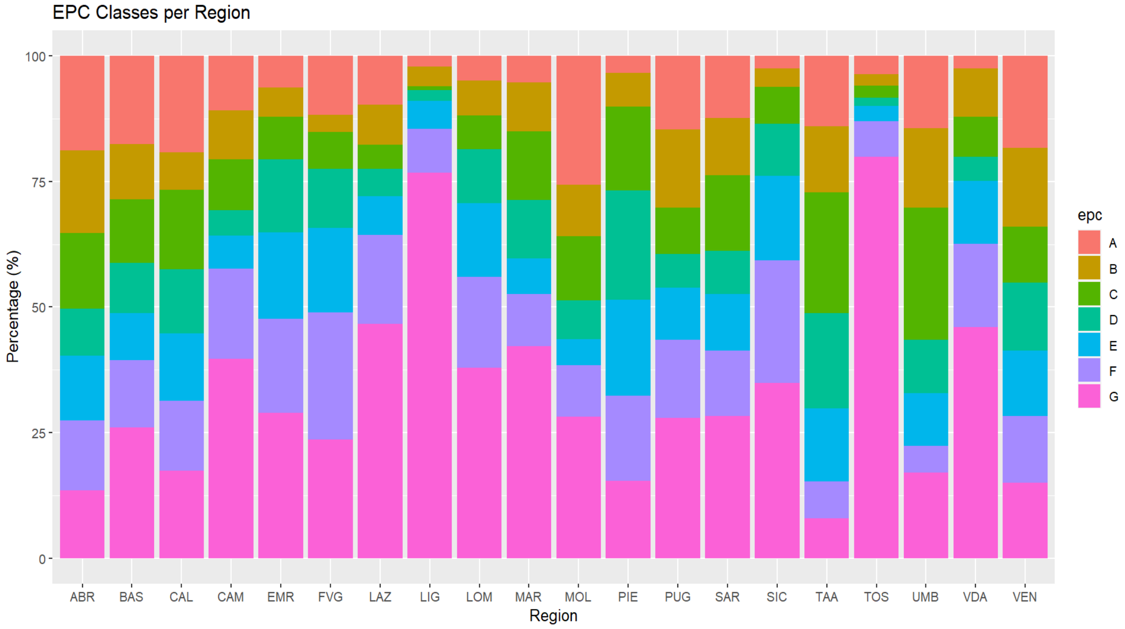

Figure 17 – This figure presents the percentage of buildings belonging to each EPC class organized by region, in alphabetical order. Each region is identified by a three-character acronym.

As we expected, as we spot from Figure 16, the A class is the one with smallest rate of default and the G class the one with the highest. If we exclude the C class, we can assume that the lower the energy efficiency class the higher the default probability (which is still really small, never reaching even 3%). Next, we compared the shares of buildings, organized by EPC classes, in each region.

In Figure 17, the region with the highest proportion of A class buildings are Molise, Veneto and Calabria, while the ones with the highest proportion of G class buildings are Tuscany, Liguria and Lazio. Afterwards, we inspected the default rate per region.

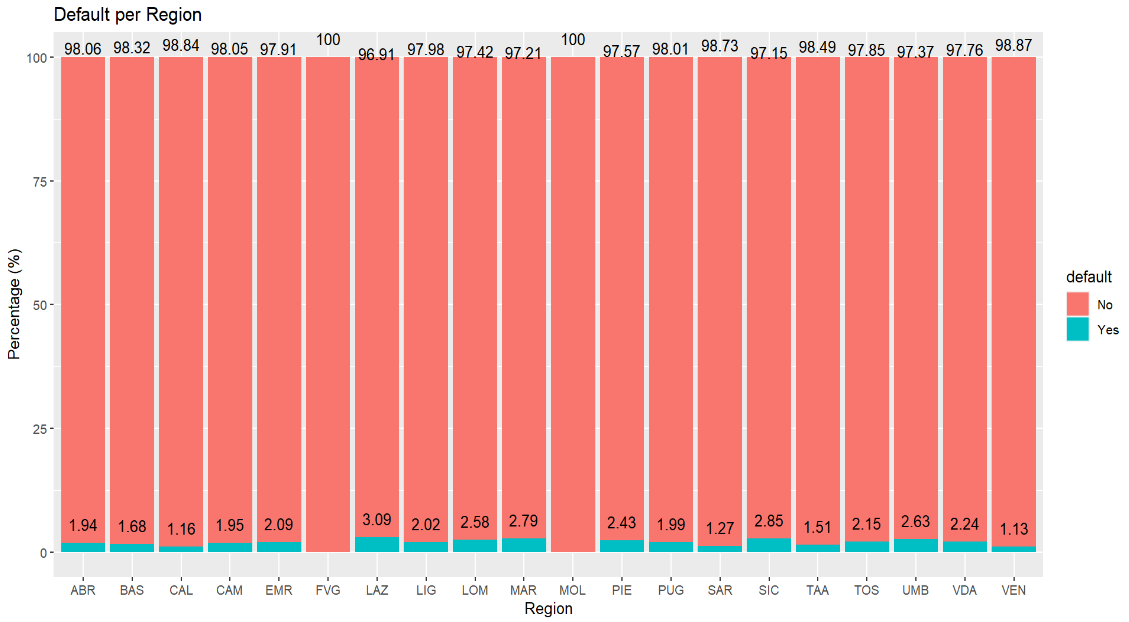

Figure 18 – This figure presents the percentage of defaulted/non-defaulted loans organized by region, in alphabetical order. Each region is identified by a three-character acronym.

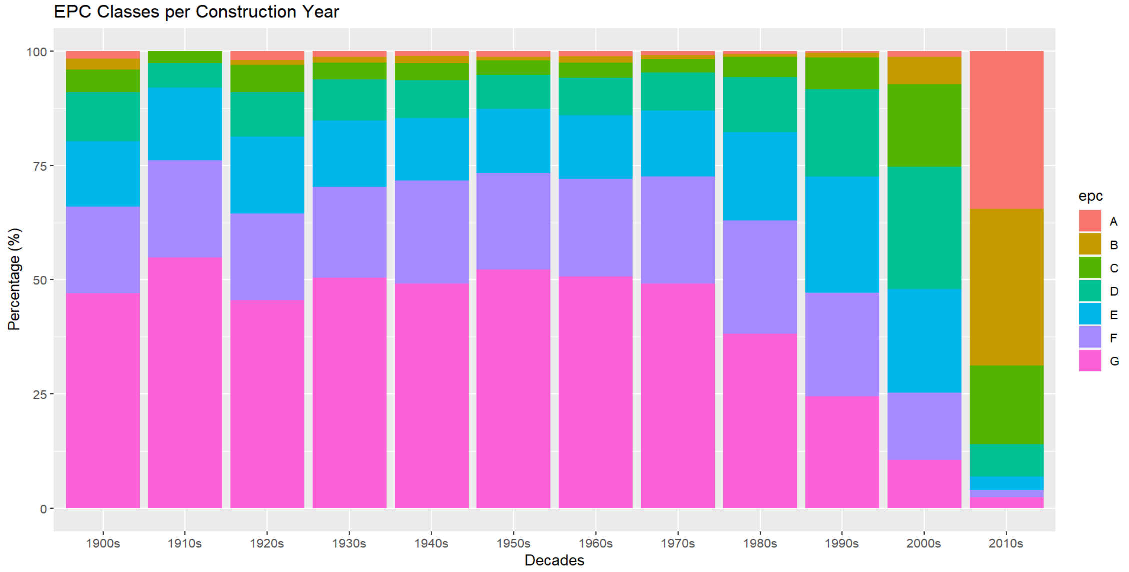

Figure 19 – This figure presents the percentage of buildings belonging to each EPC class organized by the decade of construction, in chronological order starting from the 1900s to the 2010s.

There are two regions with 0% default: Molise and Friuli Venezia Giulia. The third best is Veneto. The highest default rates are found in Lazio, Sicily and Marche (Fig.18). Then, we analyze the buildings, organized by EPC class, by construction decade.

As expected, in Figure 19, there is a significant increase in A class building construction only in the last 10 years. As long as G class buildings construction are concerned, the rate is decreasing starting from the 1950s. We need to investigate the default by decade of construction of the building, too.

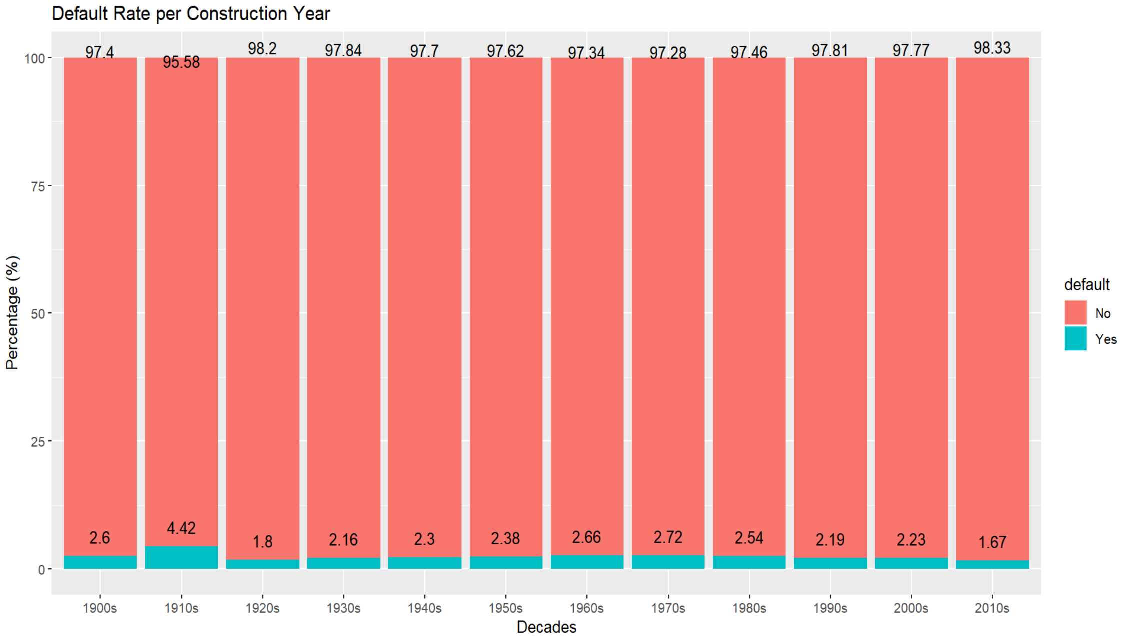

Figure 20 – This figure presents the defaulted/non-defaulted percentage of the loans associated to buildings organized by construction decade, in chronological order starting from the 1900s to 2010s.

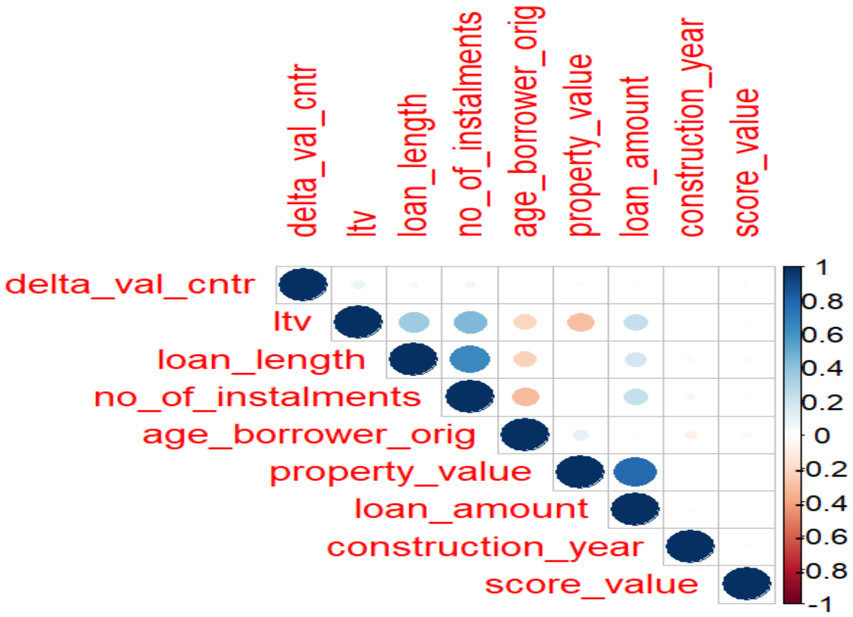

Figure 21 – This figure presents the correlogram built on the numerical variables that were not dropped in the dataset. The bigger the circle and closer to the blue, the higher and positive is the correlation, while the bigger the circle and closer to the red, the higher and negative is the correlation.

From Figure 20, we see that the buildings whose loan are more likely to not be repaid are the ones built in the 1910s, in the 1970s and in the 1960s. On the other hand, the ones of housing built in the 2010s, in the 1920s and in the 1930s have the highest probability of being repaid. Finally, we built a correlogram for the numeric variables.

In Figure 21, there's a quite high positive correlation between *loan_length* and *no_of_instalments*, between *property_value* and *loan_amount* and *ltv*, between *ltv* and *no_of_instalments*. There's a quite low negative correlation between *ltv* and *property_value*.

1.2 Pearson's chi-squared tests

A different approach we still wanted to use, specifically, to look at relationships between categorical variables and the default, is the Pearson's chi-squared test for statistical independence (Pearson, 1900). For that test, each observation is allocated to a cell of a contingency table according to the values of two outcomes (the two categorical variables whose relationship we want to study). The null hypothesis is that the occurrence of the outcomes is statistically independent while the alternative hypothesis states the opposite. Through the formula:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{\left(O_{i,j} - E_{i,j}\right)^2}{E_{i,j}},$$

we get our test statistic where $O_{i,j}$ are the observed occurrences at row i and column j and $E_{i,j}$ are the expected occurrences at row i and column j (the expected occurrences are computed by multiplying the sum of the occurrences in row i times the sum of the occurrences in column j divided by the total number of occurrences in the table).

The test statistic is distributed as a χ^2 with degrees of freedom equal to the product of the number of rows, minus 1, times the number of columns, minus 1, and the null hypothesis is rejected for big values.

In the case of the relationship between *default* and *energy efficiency* class, we computed the following contingency table:

EPC/Default	NO	YES	TOTAL
A	5723	61	5784
B	6641	126	6767
C	8371	194	8565
D	12665	277	12942
E	15294	341	15635
F	17607	400	18007
G	32463	989	33452
TOTAL	*98764*	*2388*	*101152*

Table 1 - The table presents the distribution of the defaulted(YES) and non-defaulted(NO) loans organized by EPC class.

With the data in Table 1, the value of the test statistic is 106.39, hence the p-value is <$2.2*10^{-16}$, so we reject the null hypothesis (at 5% confidence level) and we can assume that there's some kind of relationship between the energy efficiency class and default probability.

For *Property Status* and *default*, we get the following:

Property Status/Default	NO	YES	TOTAL
DARIS	1535	43	1578
NUOV	12030	252	12282
RISTR	12047	273	12320
SNUOV	8676	218	8894
USAT	64476	1602	66078
TOTAL	*98764*	*2388*	*101152*

Table 2 - This table presents the distribution of defaulted (YES)/non-defaulted(NO) loans organized by the property status of the building.

For the data in Table 2, the value of the test statistic is 8.5923, hence the p-value is 0.07214, so we cannot reject the null hypothesis (at 5% confidence level) and we cannot assume that there's some kind of relationship between the property status and the default probability.

While for *Cadastral Category* and *default*, we have the following:

Cadastral Category/Default	**NO**	**YES**	**TOTAL**
APPARTAMENTO	95453	2355	97808
ATTICO/MANSARDA	1035	31	1066
LOFT	58	0	58
VILLA/VILLINO	1433	1	1434
VILLETTA A SCHIERA	785	1	786
TOTAL	*98764*	*2388*	*101152*

Table 3 - The table presents the distribution of defaulted(YES) and non-defaulted(NO) loans organized by the cadastral category of the buildings.

Unluckily, from the data in Table 3, even though a very low p-value ($7.063*10^{-11}$), the result of this type of test is not reliable because, as a constraint of the methodology, we must have at least 5 occurrences in each cell of the table.

For *Region Macroarea* and *default*, we come up with the following:

Region Macroarea/Default	**NO**	**YES**	**TOTAL**
CENTRE	3165	80	3245
ISLANDS	4336	117	4453
NORTH-EAST	33014	677	33691
NORTH-WEST	57072	1493	58565
SOUTH	1177	21	1198
TOTAL	*98764*	*2388*	*101152*

Table 4 - This table presents the distribution of defaulted(YES) and non-defaulted(NO) loans organized by region macroarea of the Country.

As long as data in Table 4 are concerned, the value of the test statistic is 30.52, hence the p-value is $3.835*10^{-6}$, so we can reject the null hypothesis (at 5% confidence level) and assume that there's some kind of relationship between the region (dividing Italy in macroareas) of the building and the default probability. With further investigations on a higher granularity (splitting into the twenty regions), the methodology is not applicable due to not having at least 5 occurrences in each cell of the contingency table. After all the EDA, we decided to drop one more variable (*periodicity*) and keep a dataset made of 101 152 observations and 24 variables plus the target one (*default).*

CHAPTER II – Models development

Now, it's time to create and test models. After the training of a model, we test its performance on new data, so it can be compared with other models and we can choose the one that fits best our necessity. For this purpose, we decided to split our dataset in two parts: the first, called Training Set, contains 80% of the data of the original dataset (80 921 observations) and is used to train the models; the second, called Test Set, contains the remaining 20% of the data (20 231 observations) and is used to measure the performance of each model.

2.1 Overview of the methodologies

To address the model choice issue we will make use of 5 different tools that are useful to solve a problem of classification (since we want to be able to predict whether a mortgage will be repaid or not, we will use models of binary classification):

i) the Logistic Model;

ii) the Logistic Model with Ridge penalization;

iii) the Logistic Model with LASSO penalization;

iv) the Linear Discriminant Analysis;

v) the Random Forest.

2.1.1 The Logistic Model

The logistic model (Cramer, 2004) is a generalized linear model that exploit the maximum likelihood estimator. Indicating with Y_i the values of the target variable for each observation *i* in the training set, we can assume they are random samples from a Bernoulli distribution:

$$Y_i \sim Ber(\pi_i),\ \text{i = 1, ..., n},$$

where π_i is the value of probability of "success". By this model, denoting with $\boldsymbol{x_i}$ the vector of variables of the *i*-th statistical unit and with $\boldsymbol{\beta}$ the vector of estimated parameters, we can assume that

$$g(\mu_i) = \mathbf{x}_i^T \boldsymbol{\beta} = \beta_1 x_{i1} + \cdots + \beta_p x_{ip} = \sum_{r=1}^{p} \beta_r x_{ir},$$

where $g(\pi_i)$ is called the link function[4]. The link function we are going to use is the logit:

$$g(\mu_i) = \log\left(\frac{\pi_i}{1 - \pi_i}\right) = \boldsymbol{x}_i^T \boldsymbol{\beta}$$

2.1.2 The Logistic Model with Ridge penalization

Basically, the model is the same as the logistic model presented before. The main difference is the addition of a penalization parameter aimed at reducing the total mean error of the model and able to deal better with multicollinearity. To obtain the estimates of the parameters for the model with Ridge, we must minimize the following formula:

$$-\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\left\{y_i(\beta_0 + \boldsymbol{\beta}^T\boldsymbol{x}_i) - \log\left(1 + e^{\beta_0 + \boldsymbol{\beta}^T\boldsymbol{x}_i}\right)\right\} + \lambda||\boldsymbol{\beta}||_2^2,$$

where λ is a penalization coefficient (James *et al.*, 2014; Hoerl and Kennard, 1970) and *n* the sample size.

2.1.3 The Logistic Model with LASSO penalization

Similarly to the case of Ridge penalization, this model is really similar to the logistic one. We still intend to reduce the total mean error of the model with the addition of a penalization parameter but, in this case, we also operate a variable selection. To obtain the estimates of the parameters for the model with LASSO, we must minimize the following formula:

$$-\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\left\{y_i(\beta_0 + \boldsymbol{\beta}^T\boldsymbol{x}_i) - \log\left(1 + e^{\beta_0 + \boldsymbol{\beta}^T\boldsymbol{x}_i}\right)\right\} + \lambda||\boldsymbol{\beta}||_1,$$

where λ is a penalization coefficient (James *et al.*, 2014; Tibshirani, 1996) and *n* the sample size. It also does exist a model which combines linearly the Ridge and LASSO penalization, called Elastic Net (James *et al.*, 2014; Zou and Hastie, 2005), but we decided to not follow that path in this analysis.

[4] The link function is a function that directly links the usual formulation of linear model with the "new" type of model. Different link functions are utilized in different contexts (Agresti, 2015).

2.1.4 The Linear Discriminant Analysis

In the LDA (Fisher, 1936), we assume to have a p-dimensional variable ***X*** ed a categorical random variable *Y,* that represents the class to which an observations belong. The whole population of interest is divided into K classes having the probability distribution function (p.d.f.), for the distribution of ***X***, respectively $p_1(x)$, ..., $p_K(x)$, with weight π_1, ..., π_K. We hypothesize, also, that each density $p_k(x)$ is a random variable with mean μ_k and variance (matrix) Σ, hence

$$p_k(x) = \frac{1}{(2\pi)^{p/2}\det(\Sigma)^{1/2}} \, e^{\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k\right\}},$$

for k = 1, ..., K (in this case K = 2).

The other component of the density for the whole population is π_k that, unless differently specified, is estimated as follows:

$$\hat{\pi}_k = \frac{n_k}{n},$$

where n_k is the number of subjects belonging to the *k*-th class and *n* the size of the sample.

The prior probability of a non-classified subject to belong to the *k*-th class is π_k, while the posterior probability is computed by Bayes' theorem. We compute a discriminant analysis for each k:

$$d_k(x) = log\pi_k + logp_k(x_0),$$

the value of k which gives the highest value of the function, identifies the group to which we assign the subject.

2.1.5 The Random Forest

The random forest (Ho, 1995) is an ensemble learning method that operates by constructing a multitude of decision trees (Breiman *et al.*, 1984) at training time. The decision tree learning is a supervised learning approach used to build a predictive model to draw conclusions about a set of observations.

A tree is built by splitting the source dataset (the root node) into subsets (the successor children). The splitting is based on a set of rules applied to the classification rules (Shalev-Schwartz and Ben-David, 2014). The process is repeated in a recursive manner and it’s completed when the subset at a node has all the same values of the target variable, or when splitting no longer adds value to the predictions. For classification tasks, the output value of the random forest is the class selected by most trees.

2.2 Application

In this section, we are going to fit different models on our Training Set and check their prediction performances on the Test Set, so we can choose which one is the best depending on what metric or measure we will use to decide. In our case, since the data come from financial institutions it is reasonable to go for the model that has the highest sensitivity.

The first model applied to the Training Set is the Logistic Model. At first, we used as predictors, the following variables: the loan duration, the number of instalments, the amount of money loaned, the loan-to-value ratio, the energy efficiency class, the age of the borrower when the loan was requested, the construction year of the building, the score value of the borrower, the cadastral category of the building and the region macroarea of the building. After the estimation of the parameters, the construction year of the building resulted non statistically significant, so the variable was dropped. The parameters associated with some classes of the categorical predictors were not statistically significant, however the variable (in the complex) was significant, so each dummy created must be kept in the model. Only the dummies, associated with the categories “attico/mansarda” and “loft” of the variable *cadastral category* were eliminated, and those two categories were merged with the baseline one (“apartment”).

To check if the model had any predictive ability we evaluated its performance on the Training Set. The confusion matrix came out as follows:

Predicted/Actual	**NO**	**YES**	**TOTAL**
NO	44411	580	44991
YES	34598	1332	35930
TOTAL	*79009*	*1912*	*80921*

Table 5 - The table presents the confusion matrix of the logistic model fitted on the training set.

The resulting ROC Curve is the following:

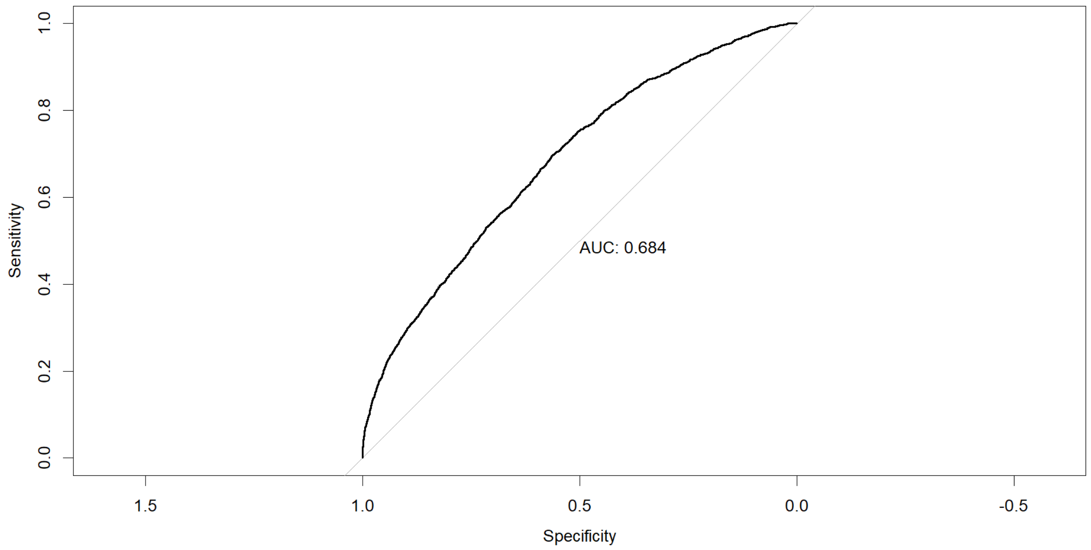

Figure 22 – The figure presents the ROC curve of the logistic model fitted on the training set.

And the main metrics resulted as follows:

ACCURACY	SENSITIVITY	SPECIFICITY
0.5652797	0.6966527	0.5621005

Table 6 - The table presents the value of the main metrics associated to the logistic model fitted on the training set.

Indeed, not a great performance. Then we evaluated the performance on the Test Set. The confusion matrix is the following:

Predicted/Actual	NO	YES	TOTAL
NO	10276	116	10392
YES	9479	360	9849
TOTAL	*19755*	*476*	*20231*

Table 7 - The table presents the confusion matrix of the logistic model fitted on the test set.

The ROC Curve as follows:

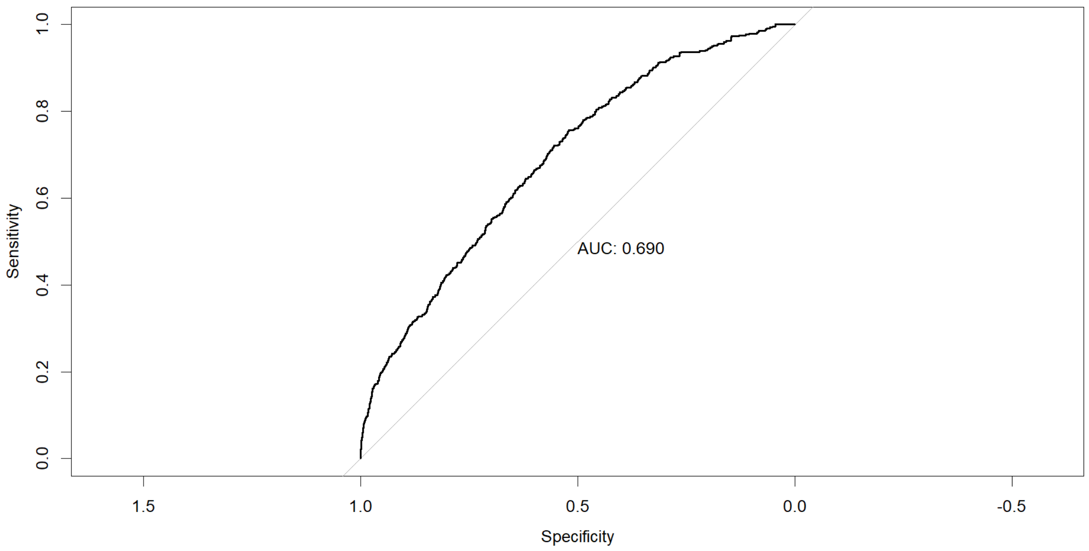

Figure 23 – The figure presents the ROC curve of the logistic model fitted on the test set.

The main metrics became as follows:

ACCURACY	SENSITIVITY	SPECIFICITY
0.5257278	0.7563025	0.5201721

Table 8 - This table presents the values of the main metrics associated to the logistic model fitted on the test set.

Even though the ROC curve had a higher AUC, as expected the 2 out of 3 of the main metrics were lower on the Test Set compared with the Training Set. Then, seeking out a further improvement we tried to add to the model some macroeconomics measures like: HPI change over year, HPI, inflation rate, mortgage rate, unemployment rate and GDP. The only variable that turned out to be not statistically significant, in the end, was the unemployment rate, which was eliminated in the subsequent step. With the introduction of those new variables we computed, once more, our measures on the Test Set. The Confusion Matrix is the following:

Predicted/Actual	NO	YES	TOTAL
NO	13596	113	13709
YES	6159	363	6522
TOTAL	*19755*	*476*	*20231*

Table 9 - The table presents the confusion matrix of the logistic model fitted with the macroeconomics variables on the test set.

The ROC Curve as follows:

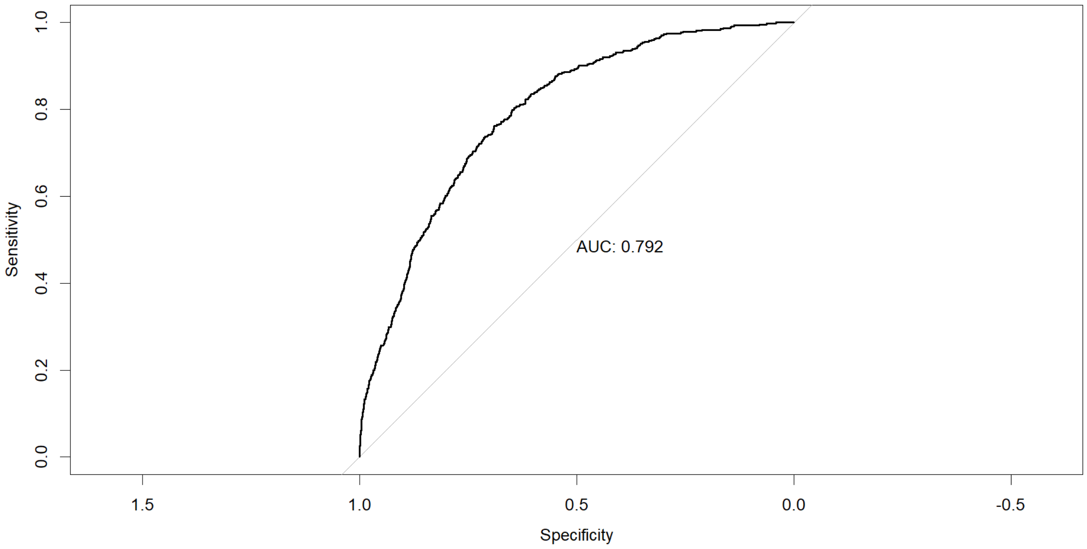

Figure 24 – The figure presents the ROC curve of the logistic model fitted with the macroeconomics variables on the test set.

And the main metrics became the following:

ACCURACY	SENSITIVITY	SPECIFICITY
0.6899807	0.762605	0.6882308

Table 10 - The table presents the values of the main metrics associated to the logistic model fitted with the macroeconomics variables on the test set.

The addition of the macroeconomics measures as variables led to a big improvement in the performance of the model. However we still thought there was room for improvement, so we went on fitting the logistic model with the Ridge penalization. The penalization coefficient λ was estimated through a 5-fold Cross-Validation on the Training Set (Allen, 1974; Stone, 1977). The results were the following.

The Confusion Matrix turned out to be as follows:

Predicted/Actual	NO	YES	TOTAL
NO	12575	89	12664
YES	7180	387	7567
TOTAL	*19755*	*476*	*20231*

Table 11 - The table presents the confusion matrix of the logistic model with ridge penalization fitted on the test set.

The ROC Curve, the following:

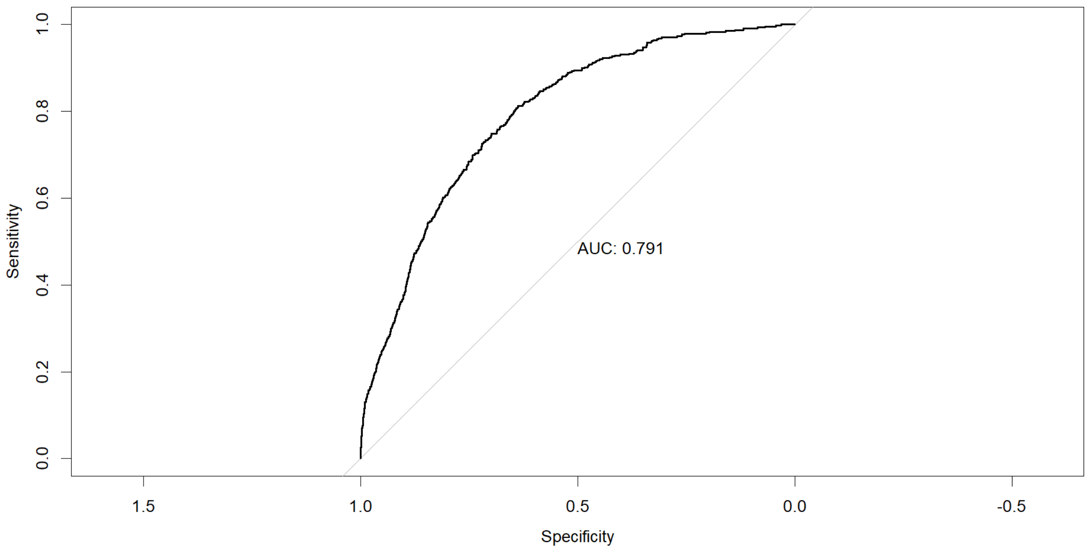

Figure 25 – The figure presents the ROC curve of the logistic model with ridge penalization fitted on the test set.

And the main metrics became the following:

ACCURACY	SENSITIVITY	SPECIFICITY
0.6406999	0.8130252	0.6365477

Table 12 - The table presents the values of the main metrics associated to the logistic model with ridge penalization fitted on the test set.

The value of the AUC for the ROC curve decreased slightly and the accuracy and the specificity did it by about 0.05 compared to the previous model. On the other hand, the sensitivity gained about 0.05. We went on fitting the logistic model with the LASSO penalization, too. The penalization coefficient λ, was still estimated as in the previous case. The results came as follows.

The Confusion Matrix, the following:

Predicted/Actual	NO	YES	TOTAL
NO	13522	112	13634
YES	6233	364	6597
TOTAL	*19755*	*476*	*20231*

Table 13 - The table presents the confusion matrix of the logistic model with LASSO penalization fitted on the test set.

The ROC Curve as follows:

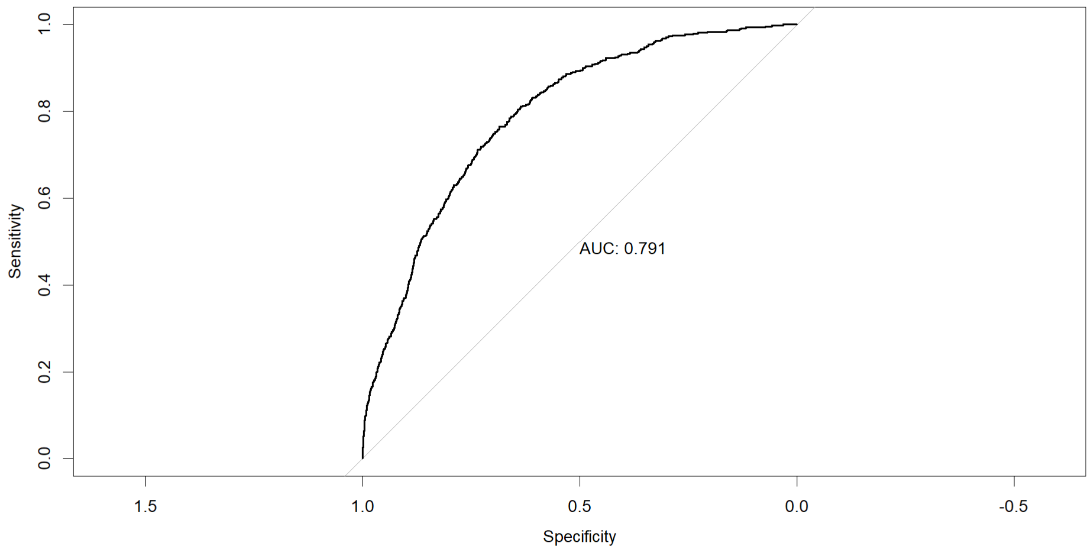

Figure 26 – The figure presents the ROC curve of the logistic model with LASSO penalization fitted on the test set.

And the main metrics resulted as follows:

ACCURACY	SENSITIVITY	SPECIFICITY
0.6863724	0.7647059	0.6844849

Table 14 - The table presents the values of the main metrics associated to the logistic model with LASSO penalization fitted on the test set.

This version of the model is quite similar to the logistic one without penalization primarily in terms of main metrics. However, as in the previous case, the AUC of the ROC curve is slightly smaller than the basic logistic one. The next step was to tackle the problem by a different "angle" and with a different tool: the LDA. The result we got are the following.

The Confusion Matrix is shown below:

Predicted/Actual	NO	YES	TOTAL
NO	13736	116	13852
YES	6019	360	6379
TOTAL	*19755*	*476*	*20231*

Table 15 - The table presents the confusion matrix of the LDA fitted on the test set.

The ROC Curve is plotted below:

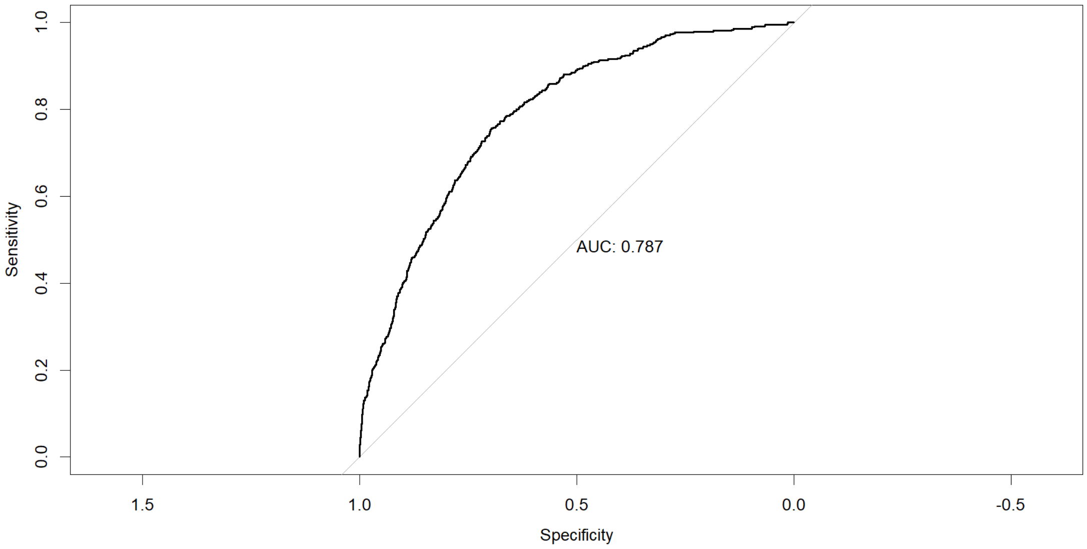

Figure 27 – The figure presents the ROC curve of the LDA fitted on the test set.

And the main metrics are the following:

ACCURACY	SENSITIVITY	SPECIFICITY
0.6967525	0.7563025	0.6953176

Table 16 - The table presents the values of the main metrics associated to the LDA fitted on the test set.

Even though the AUC of the ROC curve keeps getting smaller, compared with the one of the logistic model without penalization, this model has better accuracy and specificity compared to any other of the previous ones and has the lowest sensitivity among them. As last methodology applied to address the issue of modeling, we resort to the Random Forest, too. Our results, for the version composed by 50 classification trees[5], were the following.

The Confusion Matrix turned out to be, the following:

Predicted/Actual	NO	YES	TOTAL
NO	15297	198	13852
YES	4458	278	6379
TOTAL	*19755*	*476*	*20231*

Table 17 - The table presents the confusion matrix of Random Forest fitted on the test set.

[5] The number of 50 trees was defined by computational power and time constraints.

The ROC Curve as follows:

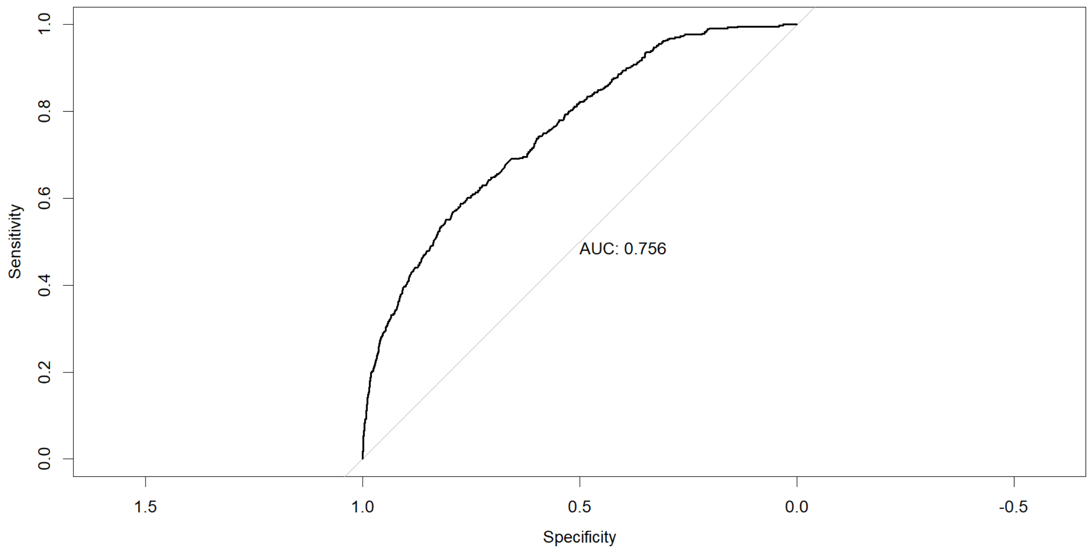

Figure 28 – The figure presents the ROC curve of the Random Forest fitted on the test set.

And the main metrics are the following:

ACCURACY	SENSITIVITY	SPECIFICITY
0.7689684	0.5882353	0.7733232

Table 18 - The table presents the values of the main metrics associated to the Random Forest fitted on the test set.

In Figure 29, we can also show the error decrease as the number of trees increased in the forest:

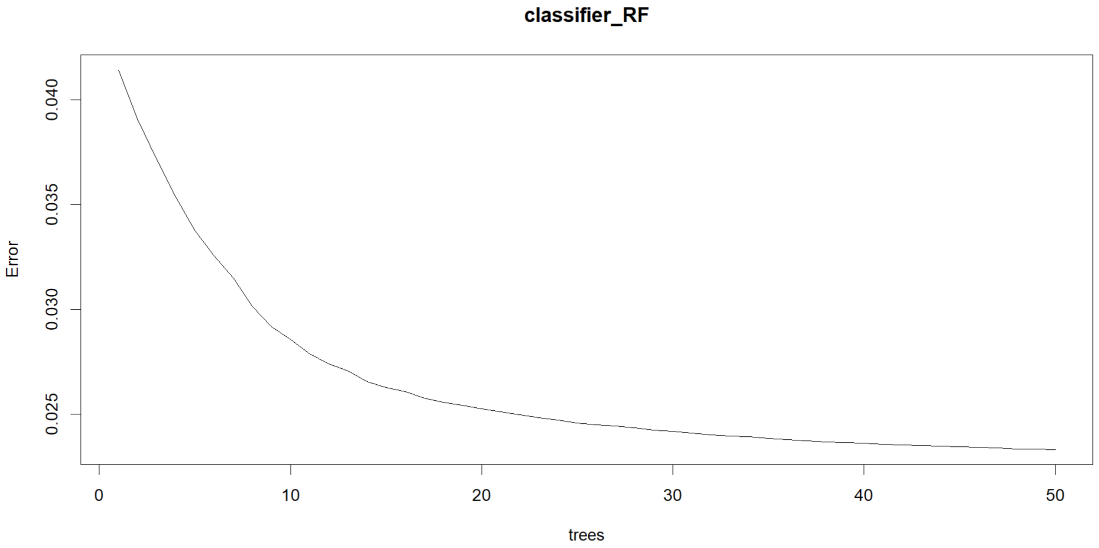

Figure 29 – The figure presents the fall of the error associated with the Random Forest as the number of trees in it increases.

And the features importance (in terms of Node Purity increase) is plotted in Figure 30:

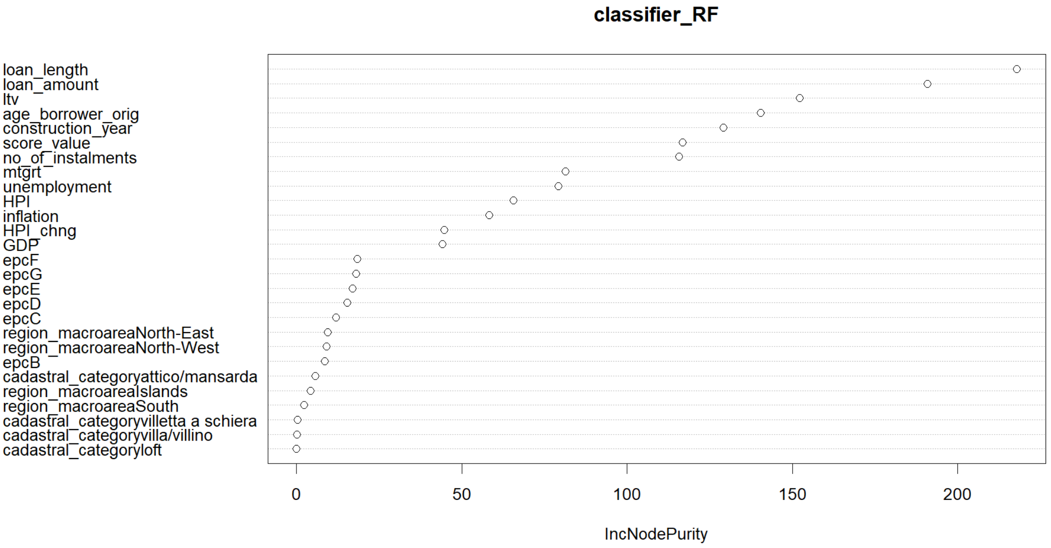

Figure 30 – The figure presents the variables importance in terms of increase in node purity.

Even though the AUC of the ROC curve is the lowest recorded among the models, with the Random Forest we get the highest accuracy and the highest specificity. Unluckily, we also record the lowest sensitivity.

2.3 Model selection and comments

Summarizing the performances of all the models, we have the following table:

MODEL	ACCURACY	SENSITIVITY	SPECIFICITY	AUC ROC
LOGISTIC	0.6899807	0.762605	0.6882308	0.792
RIDGE	0.6406999	0.8130252	0.6365477	0.791
LASSO	0.6863724	0.7647059	0.6844849	0.791
LDA	0.6967525	0.7563025	0.6953176	0.787
RANDOM FOREST	0.7689684	0.5882353	0.7733232	0.756

Table 19 - The table summarizes the values of the main metrics and the AUC of the different models fitted.

The main metrics we keep in consideration are accuracy, sensitivity, specificity and the AUC (Flach *et al.*, 2011) of the ROC curve (Fawcett, 2006).

The accuracy is computed as the sum between the True Positives (TP) and the True Negatives (TN), divided by the total number of observations (N).

The sensitivity, or True Positive Rate (TPR), is computed as the TP, divided by the sum of TP and False Negatives (FN).

The specificity, on the other hand, which can be obtained also as 1 – False Positive Rate (FPR), is more commonly computed as the TN, divided by the sum of TN and False Positives (FP).

The AUC (Area Under the Curve) measures the area that is under the ROC curve starting from the bottom-right corner of the plot to the curve itself; a value of 0.5 of the AUC indicates that the binary classification method associated has the same performance as a coin toss and is kept as lowest acceptable value. The ROC (Receiver Operating Characteristic) curve is a graphical representation of the diagnostic ability of a binary classifier as its discrimination threshold varies. It is created by plotting the TPR against the FPR at various threshold settings and is useful because it displays the trade-off between the two. In fact, the metrics in the previous table are computed with the threshold that maximizes the Youden's J statistic (Youden, 1950) in each model, which is, basically, the sum of sensitivity and specificity.

In general, to choose the more appropriate model, just one (or at most two) of the aforementioned metrics are used. If our goal is to correctly classify the highest number of observations possible, we should choose the model with the highest accuracy, in our case, the Random Forest. If we want to minimize the incorrect classification of the loans as non-defaulted when they are in fact, we should opt to the highest sensitivity, in our case, the Logistic Model with Ridge. If our goal is to have the least possible amount of incorrectly classified loans, as defaulted when they are not, we should go for the highest specificity, the Random Forest, again. Finally, if we want a model that can perform well at different threshold values, it would be a forced choice to go with the highest AUC of the ROC curve, the simple Logistic Model, in this case.

Since the data we got come from banks, is safe to assume that making a type I error (predicting that a loan will be repaid while it won't) has a higher weight compared to making a type II error (predicting that a loan will not be repaid while it will), because that kind of companies prefer a missed chance of profit than a certain financial loss. Keeping that in mind, the model to choose is the one with the highest sensitivity: the *Logistic Model with Ridge* penalization.

Furthermore, the development of the aforementioned models made us understand what variables, apart from the energy efficient class, are relevant in predicting the probability of default of a loan and what kind of impact they have. As long as mortgage and/or building data are concerned, we have the loan-to-value ratio and the number of instalments as important predictors (both positively correlated with the default probability). The loan length has a remarkable effect, too (event though the correlation with the default is negative). The variables about the borrower that are impactful are the age when loan was requested, with positive correlation with the default and the credit score value. The credit score value has a difficult interpretation because, because out of three models, in two of them, the higher the value the higher the probability of default whereas, in the third one, the effect is quite the opposite, indicating that high credit score values are associated with low default probabilities. The last interpretation is more in line with what should be expected but, since the variable got 30% of the data imputed automatically during the data cleaning, the likely introduction of some kind of biases in the analysis is something we have to take into account. Lastly, the variables concerning macroeconomics indices that greatly affect the models are the Home Performance Index (HPI), the inflation rate, both with positive impact on the default, and the Gross Domestic Product (GDP), with opposite leverage. As a matter of fact, the EPC variable shows an increasing likelihood of default as the energy class gets worse in the models, but it’s true nature will be investigated in the next chapter.

CHAPTER III – Evaluation of the energy efficiency component

After having explored the different models we investigate further the relationship between the energy efficiency of the buildings and the probability of default of the associated loan.

3.1 Overview of the methodologies

The tools we mainly intend to use to evaluate the relationship mentioned above are two: the Cox Proportional Hazards Model and an array of different specification of the same model with different coding/splitting of the energy efficient variable.

3.1.1 The Cox Proportional Hazards Model

The Cox Proportional Hazards Model (Cox, 1972) is one of the most common models utilized with survival data. It's defined

$$h(t, \boldsymbol{x}) = h_0(t) exp\left(\sum_{l=1}^{p} \beta_l x_l\right),$$

where $h_0(t)$ is a baseline hazard function (that depends only on time *t*), each x_l a covariate and each β_l the associated parameter. The useful property is that, while the baseline hazard function depends only on the time t, the exponential function depends only on the covariates. The hazard ratio is defined as the ratio of hazards for two subjects in the study. Assuming that we have subject *i* and subject *j*, we can express the hazards ratio as

$$\widehat{HR} = \exp(\sum_{l=1}^{p} \widehat{\beta}_l (x_l^i - x_l^j)) = \theta,$$

so the relation between two subjects can be expressed as $\hat{h}(t_k^i, \boldsymbol{X}^i) = \theta \hat{h}(t_k^j, \boldsymbol{X}^j)$.

The empirical survival function, represented by the Kaplan-Meier curve (Kaplan and Meier, 1958), can show if the proportional hazards assumption holds on. The definition of the function is

$$\hat{S}_{t_m} = \prod_{i=1}^{m} \Pr\ (T > t_i \,|\, T \geq t_i),$$

where t_m is the event time (ordered) and the probabilities are approximated by the frequency. In the context of mortgage analysis, we must take into consideration the left-truncated mortgages (that originated before the first observation date) and right-censored mortgages (which still ongoing by the end of the study). It is common practice to use a dummy variable to identify censored observations, so we followed that route.

3.1.2 The Array of Different Models

To investigate further the effect of the energy efficiency on the default probability we will also use a different approach. We will compare the predictive abilities of the Logistic Model with Ridge using different sets of variables. The first set does not contain any information on the energy efficiency of the building ("NO" model), the second has only a dummy that identifies if the building is an A class building ("EE_A" model), the third will have a dummy that identifies if the building falls into class A, B o C ("EE_ABC" model), and the last one will have a dummy for each class (minus 1, for basic multicollinearity reasons) as it was in the models fitted in the previous chapter ("EPC" model).

3.2 Application

To assess the impact of energy efficiency on default risk, we created a dummy (called "EE_A") that takes value 1 if the building belongs to class A and 0 otherwise, and then applied the Cox Proportional Hazards Model. The coefficient associated with the dummy indicates that, by the model, if the building of the class belongs to class A the "survival probability" of the loan is lower (conversely to the results obtained in Billio *et al.*, 2022) and the Log-Rank test (that evaluates if the variable has a significant impact on the survival function) has a p-value lower than $2.2*10^{-16}$, so we have the reject the null hypothesis that states the absence of impact by the energy efficiency on the default. The same result is found even when adding the credit score, the loan-to-value, the loan term, the building age, the borrower age when requesting the loan, the inflation, the unemployment and the HPI change as control variables, even though the magnitude of the EE_A variable decreases slightly.

The aforementioned results are also supported by the Kaplan-Meier curve, in Figure 31.

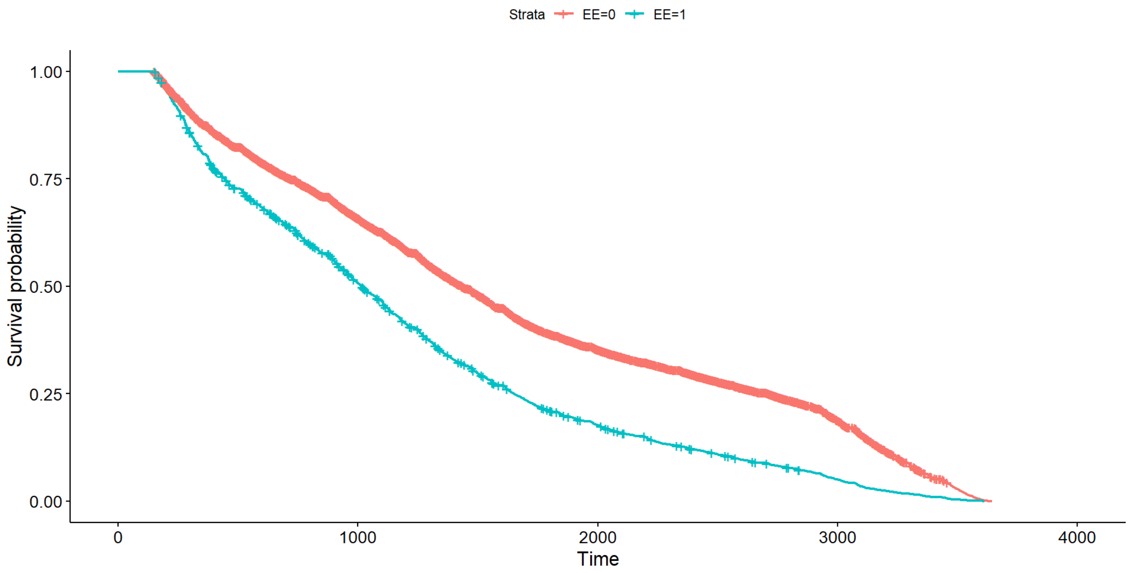

Figure 31 - The figure presents the Kaplan-Meier curve for the survival probability of A class buildings (in blue) and non-A class buildings (in red).

For the comparison of the models with the four different sets of variable, we also built another dummy variable (called "EE_ABC") which takes value 1 if the building belongs to class A, B or C and 0 otherwise. In each one of them the variable associated with the energy efficiency, if present, is statistically significant. The results are summarized in the following table:

MODEL	ACCURACY	SENSITIVITY	SPECIFICITY	AUC ROC
NO	0.7206268	0.7289916	0.7204252	0.788
EE_A	0.7221079	0.7268908	0.7219944	0.789
EE_ABC	0.64788177	0.8046218	0.6440395	0.789
EPC	0.6406999	0.8130252	0.6365477	0.791

Table 20 - The table summarizes the value of the main metrics and the AUC of the models fitted. The *NO* model is the model without any variable concerning the energy efficiency; the *EE_A* model is the model that splits the energy efficiency buildings between A-class dwellings and the other ones; the *EE_ABC* model is the model that divides the energy efficient buildings between A-, B-, or C-class dwellings and the other ones; the *EPC* model is the model that has a dummy for each class different from A-class.

As we can see in Table 20, no matter what our main metric is, there will always be a model including some type of energy efficiency variable with better performance than the model without. However, we can go even a step further. We noted that, if our main goal is to have the better predictive capability possible, there's no need to split the energy efficient variable into 7 categories (1 for each class), it would be better to just divide the A class buildings from the other ones. By doing that, we don't just improve the model performance but also reduce the model complexity and computational cost.

Whereas, if our most important metric is the sensitivity, as we claimed in the previous chapter in the model selection section, the splitting of the variables is necessary so that we can minimize the chance to identify a loan a non-defaulting, when in fact it is going to default. Conversely, if we want to rely mostly on specificity, so minimizing the missed chance of loan (minimizing the potential loans that are predicted as defaulting while they're going to be repaid), we can go back to the A-class splitting: we should have a dummy that is equal to 1 if the buildings belong to class A and 0, otherwise, exactly as we would do, if we wanted to give the most importance to the accuracy. In case we wanted more flexibility, so be generically better at predicting, independently by the threshold value, the 7-class splitting, as in the case of maximum sensitivity, is mandatory. Lastly, we can point out that the splitting between A, B and C classes buildings doesn't provide enough benefits in any case considered, since we found that the models with a different set of dummy variables perform better.

However, those last considerations hold as long as the logistic model with Ridge penalization is considered. The comparison between models was performed with that as a model structure because it was the best model chosen, for our purposes, in the selection section. If the same analysis was performed with a different model structure, for example a random forest, the results may have led to different conclusions.

CONCLUSION

The goals of the analysis are to find a better performing model than the logistic model (used as a benchmark) and to investigate the relationship between the energy efficiency of a dwelling and the default risk of the mortgage associated. The first part was carried out by fitting different types of models (Logistic, Logistic with both Ridge and LASSO penalizations, Linear Discriminant Analysis and Random Forest) and comparing their performances, while the second objective was achieved by two different approaches: we fitted a Cox Proportional Hazards Model to check if the A-class buildings had a significantly different probability of defaulting compared to all the other buildings and we compared the predictive performances of the best model found in the first part of the analysis (in our case, the Logistic Model with Ridge Penalization) with different sets of variables, by the point of view of the energy efficiency (we had a model without any information about energy efficiency, a model with a splitting between A-class buildings and the others, a model with a splitting between the A-, B- and C-class buildings and the others, and a model that had a dummy associated to each EE class).

In chapter II, we have been able to build different types of models to predict the default probability of a loan and, starting from the classical Logistic Regression as a benchmark, we obtained greater performance by adding more sophisticated elements (the Ridge or Lasso penalizations) or by taking different approaches (the Linear Discriminant Analysis and the Random Forest). As already mentioned, the “best” model, by our point of view, is the *Logistic Model with Ridge* penalization but, based on different premises and considerations, the model selection may lead to a different decision. However, primarily, if the choice has to fall on another and more complex/different model as the best one (e.g. LDA or Random Forest), we may end up facing some interpretability issues.

For this reason, it is common practice, primarily in business environments, to go back to the easier (and simpler) classical Logistic Model that, despite being less accurate, is much more easily understandable and explainable than the other ones consider may be. Anyway, counterintuitively, the models identify as default risk boosters the impacts of the loan-to-value, number of instalments, age of the borrower when requesting the loan, HPI and inflation variables. On the other side, we have the GDP and the loan term variables and the EPC dummies as long as the class is higher than the G one. The impact of the credit score value variable is mutable, in the classical logistic and in the Ridge models it has a positive impact on the default probability, while in the LASSO model, we have the opposite. This “double-agent” behavior of the variable may be due the imputation procedure that it had to go through and that causes some issues also, in the following part of the analysis. A further feature of the analysis, worth discussing, is that probably the most “correct” model, at least in terms of impact interpretation of the different components of introduced in modeling the default risk, is the LASSO one, since it's the only model in which the credit score variable behaves as it supposed to be, despite the missing values imputation and even though the prediction accuracy metrics are not as good as in other cases.

In chapter III, on the other hand, we investigated specifically the relationship between energy efficiency and the default risk, with two different approaches. In that case, the results of the Cox models (both with and without the control variables) were unexpectedly in contrast with the findings in previous works, in particular the one by Billio *et al.* (2022), since it shared the same initial data. The difference between the two analysis, and the reason why ours may be considered less influential, is primarily due to the dataset handling: in the aforementioned work, the Cox Proportional Hazard Model was fitted on a dataset of about 70 000 observations while, in our case, we kept 30 000 more observations in the analysis and those underwent an imputation procedure for the credit score value variable (almost every previous work, and even our findings, on the topic demonstrated the importance of the credit score in predicting the default risk in a mortgage). Considering that the score value is one of the most decisive factors also in our prediction of a loan default, and that the imputation method provided just the mean, having 30% of the values of a suboptimal quality, may have generated important biases in the model that led to the unexpected or misleading results. Moreover, the 30 000 observations which underwent the imputation procedure, had a share of defaulted loans twice as large as the average one in the dataset, so that may have played a big role in adding bias to the analysis, too. Supporting our hypothesis that the data imputation may have biased the results of the model, the results for the credit score value and the inflation control variables is the opposite from the one in Billio *et al.* (2020).

As long as the other approach adopted (the comparison between models with different sets of variables) is concerned, the results are in line with the findings of previous papers and strongly support the assumption according to which the energy efficiency component of a building is a significant predictor of the default probability of the loan associated to it.

According to what came out in this analysis, the usual procedures and/or models used to compute the credit score of a borrower, and therefore, the policies regarding loans should be updated. Generally, the credit score value takes into account just behavioral, financial and demographical information about the borrower, and those information, combined with loan-specific characteristics such as the LTV ratio are used to predict the probability of default of a mortgage. However, also considering the energy efficiency of the building involved in the loan, would bring benefits to both the lender and the borrower, since the default risk is used to determine the volume of credit granted and the interest rate charged.

APPENDIX

The analysis was carried out in RStudio environment mounting the software R version 4.1.1 (2021-08-10, “Kick Things”). The whole code running time lasts between one hour and a half and two hours, depending on the computational power of the machine that runs that (the longer lasting part is the fitting of the random forest with 50 trees).

The code below is the one used to perform the analysis, from uploading data to model development and final comparison.

```r
#Uploading useful libraries

library(tidyverse)
library(lubridate)
library(ggplot2)
library(Hmisc)
library(corrplot)
library(pROC)
library(glmnet)
library(MASS)
library(randomForest)
library(survival)
library(survminer)

#Uploading and giving a first look to data
data<-read.csv("EeDaPP_Portfolio_CRIF.csv")
data_cox_chapter<-data
str(data)
summary(data)

##################### DATA CLEANING ###########################

#Dropping useless columns in bunches
data<-subset(data, select=-X) #drop X column
data<-subset(data, select=-age_borrower_today) #drop redundant column
data<-subset(data, select=-c(perf_30_06_2010:perf_31_12_2019)) #drop useless columns
data<-subset(data, select=-c(perf_6m:perf_48m)) #drop useless columns
data<-subset(data, select=-c(anno_nascita)) #drop redundant column
data<-subset(data, select=-c(residence_province)) #drop column to reduce granularity
data<-subset(data, select=-cadastre_category) #drop useless column
data<-subset(data, select=-birthday) #drop useless column
data<-subset(data, select=-property_type_enc) #drop duplicated column
data<-subset(data, select=-score_class_enc) #drop duplicated column
data<-subset(data, select=-score_class) #drop redundant column
data<-subset(data, select=-id) #drop useless column
data<-subset(data, select=-NUTS1_region_enc) #drop duplicated column
data<-subset(data, select=-itter107) #drop useless column
data<-subset(data, select=-property_region_enc) #drop duplicated column
data<-subset(data, select=-score_class_4_groups) #drop useless column
data<-subset(data, select=-c(p30,p70)) #drop useless columns
data<-subset(data, select=-c(yq, ym, year)) #drop useless columns
data<-subset(data, select=-c(default_12m, default_24m)) #drop useless columns
data<-subset(data, select=-c(epc_date)) #drop useless column
data<-subset(data, select=-c(loan_term, EE, EE_A, EE_ABC, default_since_origination,
                             building_age_orig:lloan_term, last_perf_date))
data<-subset(data, select=-lGDP)
data<-subset(data, select=-case) #drop useless column
data<-subset(data, select=-role)#drop useless column
data<-subset(data, select=-date)# drop useless column
data<-subset(data, select=-c(territory, property_status_group, months_since_originati
on,
             months_since_origination_cat, lmonths_since_origination,
             months_beg_last_perf, tipo_valutazione)) #drop useless columns


#combining information to have onloy default date column
colnames(data)[24]<-"perf_default_date"
for (i in 1:dim(data)[1]){
  if (data$perf_default_date[i]=="" & data$perf_date[i]!=""){
    data$perf_default_date[i]<-data$perf_date[i]
  }
}

#Dropping some more useless columns and incomplete rows plus changing data type
data$perf_default_date<-as.Date(data$perf_default_date)
data<-subset(data, select=-c(default_date_G, default_date_I, default_date_S,
                             default_date_X, default_date_Y,perf_date))
data$date_contract_begin<-as.Date(data$date_contract_begin)
data$date_contract_end<-as.Date(data$date_contract_end)
data<- data[!(is.na(data$date_contract_end)),]
data<- data[!(is.na(data$ltv)),] #drop rows with ltv equal to Na (since they all
                                 #have a loan amount greater than the property value)
data<- data[!(is.na(data$age_borrower_orig)),] #drop rows with age_borrower_orig equa
l
                                               #Na (just 0.5% of data)


#Data imputation
score_val_mean<-mean(data$score_value, na.rm=TRUE)
for (i in 1:dim(data)[1]){
  if (is.na(data$score_value[i])){
    data$score_value[i]<-score_val_mean #replace Na with mean
  }
}

HPI_chng_mean<-mean(data$HPI_chng, na.rm=TRUE)
for (i in 1:dim(data)[1]){
  if (is.na(data$HPI_chng[i])){
    data$HPI_chng[i]<-HPI_chng_mean #replace Na with mean
  }
}

# Make regions name equal between two columns
for (i in 1:dim(data)[1]){
  if(data$property_region[i]=="EMILIA ROMAGNA"){
    data$property_region[i]<-"EMILIA-ROMAGNA"
  }
}
for (i in 1:dim(data)[1]){
  if(data$property_region[i]=="TRENTINO ALTO ADIGE"){
    data$property_region[i]<-"TRENTINO-ALTO ADIGE"
  }
}
for (i in 1:dim(data)[1]){
  if(data$property_region[i]=="VALLE D AOSTA"){
    data$property_region[i]<-"VALLE D'AOSTA"
  }
}

for (i in 1:dim(data)[1]){
  if(data$property_region[i]=="FRIULI VENEZIA GIULIA"){
    data$property_region[i]<-"FRIULI-VENEZIA GIULIA"
  }
}

for (i in 1:dim(data)[1]){
  if(data$residence_region[i]=="ND" | data$residence_region[i]==""){
    data$residence_region[i]<-"SCONOSCIUTA"
  }
}

#changing column names
colnames(data)[21]<-"region_macroarea"
colnames(data)[25]<-"inflation"

#default encoding
data$default<-ifelse(data$default==0, "No", "Yes")

data<- data[!(data$periodicity=="N"),] #only 4 occurrences
data<- data[!(data$periodicity=="V"),] #only 4 occurrences

#dropping useless variablr, creating a new one and arranging data by id
data<-subset(data, select=-property_type)
loan_length<-time_length(difftime(data$date_contract_end,data$date_contract_begin),
                         "years")
data<-cbind(data[,1:3],loan_length,data[,4:26])
r<-data$row_id
data<-subset(data, select=-row_id)
data<-data.frame(cbind(r, data))
colnames(data)[1]<-"ID"

data<-subset(data, select=-delta_val_cntr)
data<-arrange(data, ID) #arrange data by row_id
summary(data)

################################# DESCRIPTIVE ANALYTICS ######################

# CREATING VARIABLE'S GRAPHS and SUMMARY STATISTICS #

data<-as_tibble(data)
default_prop <- data %>%
                group_by(default) %>%
                summarise(cnt = n()) %>%
                mutate(freq = round(cnt / sum(cnt)*100, 1))

default_prop #highly unbalanced classes
ggplot(default_prop, aes(x = default, y = freq, fill = default)) +
  geom_col() +
  geom_text(aes(label = freq), vjust = -0.5)+
  labs(title = "Default distribution",
       y = "Percentage (%)", x = "Default")


periodicity_prop <- data %>%
  group_by(periodicity) %>%
  summarise(cnt = n()) %>%
  mutate(freq = round(cnt / sum(cnt)*100, 2))

periodicity_prop
ggplot(periodicity_prop, aes(x = periodicity, y = freq, fill = periodicity)) +
  geom_col() +
  geom_text(aes(label = freq), vjust = -0.5)+
  labs(title = "Periodicity distribution",
       y = "Percentage (%)", x = "Periodicity")


epc_prop <- data %>%
  group_by(epc) %>%
  summarise(cnt = n()) %>%
  mutate(freq = round(cnt / sum(cnt)*100, 1))

epc_prop
```

```
ggplot(epc_prop, aes(x = epc, y = freq, fill = epc)) +
  geom_col() +
  geom_text(aes(label = freq), vjust = -0.5)+
  labs(title = "EPC distribution",
       y = "Percentage (%)", x = "EPC class")


PReg_prop <- data %>%
  group_by(property_region) %>%
  summarise(cnt = n()) %>%
  mutate(freq = round(cnt / sum(cnt)*100, 3))

PReg_prop

ggplot(PReg_prop, aes(x = property_region, y = freq, fill = property_region)) +
  geom_col() +
  geom_text(aes(label = freq), vjust = -0.5)+
  labs(title = "Property Region distribution",
       y = "Percentage (%)", x = "Region")


RReg_prop <- data %>%
  group_by(residence_region) %>%
  summarise(cnt = n()) %>%
  mutate(freq = round(cnt / sum(cnt)*100, 3))

RReg_prop

ggplot(RReg_prop, aes(x = residence_region, y = freq, fill = residence_region)) +
  geom_col() +
  geom_text(aes(label = freq), vjust = -0.5)+
  labs(title = "Residence Region distribution",
       y = "Percentage (%)", x = "Region")

PStatus_prop <- data %>%
  group_by(property_status) %>%
  summarise(cnt = n()) %>%
  mutate(freq = round(cnt / sum(cnt)*100, 3))

PStatus_prop

ggplot(PStatus_prop, aes(x = property_status, y = freq, fill = property_status)) +
  geom_col() +
  geom_text(aes(label = freq), vjust = -0.5)+
  labs(title = "Property Status distribution",
       y = "Percentage (%)", x = "Status")

CCategory_prop <- data %>%
  group_by(cadastral_category) %>%
  summarise(cnt = n()) %>%
  mutate(freq = round(cnt / sum(cnt)*100, 3))

CCategory_prop

ggplot(CCategory_prop, aes(x = cadastral_category, y = freq, fill = cadastral_categor
y)) +
  geom_col() +
  geom_text(aes(label = freq), vjust = -0.5)+
  labs(title = "Cadastral Category distribution",
       y = "Percentage (%)", x = "Category")



ggplot(data) +
  geom_boxplot(aes(y = property_value))+
  labs(title = "Property Value",
```

```r
       y = "Value (???)")
IQR_prop_value<-IQR(data$property_value)
summary(data$property_value)


ggplot(data) +
  geom_boxplot(aes(y = loan_amount))+
  labs(title = "Loan Amount",
       y = "Value (???)")
IQR(data$loan_amount)
summary(data$loan_amount)

ggplot(data)+
  geom_histogram(aes(x=ltv))+
  labs(title = "Loan to Value",
       y = "Frequency")
IQR(data$ltv)
summary(data$ltv)
sum(data$ltv>1)

ggplot(data)+
  geom_histogram(aes(x=age_borrower_orig))+
  labs(title = "Age of borrower (when the loan was requested)",
       y = "Frequency")
IQR(data$age_borrower_orig)
summary(data$age_borrower_orig)
mean(data$age_borrower_orig)



ggplot(data)+
  geom_histogram(aes(x=construction_year))+
  labs(title = "Construction Year",
       y = "N°", x = "Year")

summary(data$construction_year)

ggplot(data) +
  geom_boxplot(aes(y = score_value))+
  labs(title = "Score Value",
       y = "Value")
summary(data$score_value)


ggplot(data)+
  geom_histogram(aes(x=loan_length))+
  labs(title = "Loan length (in years)",
       x = "Length (years)", y = "Frequency")
summary(data$loan_length)

def_data<- data %>%
  filter(default=="Yes") %>%
  mutate(def_time=time_length(difftime(perf_default_date,date_contract_begin),
                                  "months"))

def_data

ggplot(def_data)+
  geom_histogram(aes(x=def_time))+
  labs(title = "Default time (in months)",
       x = "Length (months)", y = "Frequency")

summary(def_data$def_time)

def_per_class<- data %>%
  group_by(epc, default) %>%
  summarise(cnt = n()) %>%
  mutate(freq = round(cnt / sum(cnt)*100, 2))

def_per_class

ggplot(def_per_class, aes(x = epc, y = freq, fill = default)) +
  geom_col() +
  geom_text(aes(label = freq), vjust = -1)+
  labs(title = "Default by EPC class",
       y = "Percentage (%)", x = "Class")


class_per_reg<- data %>%
  group_by(property_region, epc) %>%
  summarise(cnt = n()) %>%
  mutate(freq = round(cnt/sum(cnt)*100, 2))


class_per_reg$property_region<-ifelse(class_per_reg$property_region=="ABRUZZO",
                                      "ABR", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="BASILICATA",
                                      "BAS", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="CALABRIA",
                                      "CAL", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="CAMPANIA",
                                      "CAM", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="EMILIA-ROMAGNA
",
                                      "EMR", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="FRIULI-VENEZIA
GIULIA",
                                      "FVG", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="LAZIO",
                                      "LAZ", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="LIGURIA",
                                      "LIG", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="LOMBARDIA",
                                      "LOM", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="MARCHE",
                                      "MAR", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="MOLISE",
                                      "MOL", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="PIEMONTE",
                                      "PIE", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="PUGLIA",
                                      "PUG", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="SARDEGNA",
                                      "SAR", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="SICILIA",
                                      "SIC", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="TOSCANA",
                                      "TOS", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="TRENTINO-ALTO A
DIGE",
                                      "TAA", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="UMBRIA",
                                      "UMB", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="VALLE D'AOSTA",
                                      "VDA", class_per_reg$property_region)
class_per_reg$property_region<-ifelse(class_per_reg$property_region=="VENETO",
                                      "VEN", class_per_reg$property_region)



ggplot(class_per_reg, aes(x = property_region, y = freq, fill = epc)) +
  geom_col() +
  labs(title = "EPC Classes per Region",
       y = "Percentage (%)", x = "Region")

def_per_reg<- data %>%
  group_by(property_region, default) %>%
  summarise(cnt = n()) %>%
  mutate(freq = round(cnt/sum(cnt)*100, 2))


def_per_reg$property_region<-ifelse(def_per_reg$property_region=="ABRUZZO",
                                      "ABR", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="BASILICATA",
                                      "BAS", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="CALABRIA",
                                      "CAL", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="CAMPANIA",
                                      "CAM", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="EMILIA-ROMAGNA",
                                      "EMR", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="FRIULI-VENEZIA GIUL
IA",
                                      "FVG", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="LAZIO",
                                      "LAZ", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="LIGURIA",
                                      "LIG", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="LOMBARDIA",
                                      "LOM", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="MARCHE",
                                      "MAR", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="MOLISE",
                                      "MOL", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="PIEMONTE",
                                      "PIE", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="PUGLIA",
                                      "PUG", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="SARDEGNA",
                                      "SAR", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="SICILIA",
                                      "SIC", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="TOSCANA",
                                      "TOS", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="TRENTINO-ALTO ADIGE
",
                                      "TAA", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="UMBRIA",
                                      "UMB", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="VALLE D'AOSTA",
                                      "VDA", def_per_reg$property_region)
def_per_reg$property_region<-ifelse(def_per_reg$property_region=="VENETO",
                                      "VEN", def_per_reg$property_region)




ggplot(def_per_reg, aes(x = property_region, y = freq, fill = default)) +
  geom_col() +
  geom_text(aes(label = freq), vjust = -1)+
  labs(title = "Default per Region",
       y = "Percentage (%)", x = "Region")


decade<-rep("",dim(data)[1])

for (i in 1:dim(class_per_conyear)[1]){
  if (class_per_conyear$construction_year[i]<=1909){
    decade[i]<-"1900s"
  }
  if (class_per_conyear$construction_year[i]>1909 &
      class_per_conyear$construction_year[i]<=1919){
    decade[i]<-"1910s"
  }
  if (class_per_conyear$construction_year[i]>1919 &
      class_per_conyear$construction_year[i]<=1929){
    decade[i]<-"1920s"
  }
  if (class_per_conyear$construction_year[i]>1929 &
      class_per_conyear$construction_year[i]<=1939){
    decade[i]<-"1930s"
  }
  if (class_per_conyear$construction_year[i]>1939 &
      class_per_conyear$construction_year[i]<=1949){
    decade[i]<-"1940s"
  }
  if (class_per_conyear$construction_year[i]>1949 &
      class_per_conyear$construction_year[i]<=1959){
    decade[i]<-"1950s"
  }
  if (class_per_conyear$construction_year[i]>1959 &
      class_per_conyear$construction_year[i]<=1969){
    decade[i]<-"1960s"
  }
  if (class_per_conyear$construction_year[i]>1969 &
      class_per_conyear$construction_year[i]<=1979){
    decade[i]<-"1970s"
  }
  if (class_per_conyear$construction_year[i]>1979 &
      class_per_conyear$construction_year[i]<=1989){
    decade[i]<-"1980s"
  }
  if (class_per_conyear$construction_year[i]>1989 &
      class_per_conyear$construction_year[i]<=1999){
    decade[i]<-"1990s"
  }
  if (class_per_conyear$construction_year[i]>1999 &
      class_per_conyear$construction_year[i]<=2009){
    decade[i]<-"2000s"
  }
  if (class_per_conyear$construction_year[i]>2009 &
      class_per_conyear$construction_year[i]<=2019){
    decade[i]<-"2010s"
  }
}
head(decade)
class_per_conyear<-as_tibble(data.frame(cbind(class_per_conyear,decade)))

class_per_conyear<- class_per_conyear %>%
  group_by(decade, epc) %>%
  summarise(cnt = n()) %>%
  mutate(freq = round(cnt/sum(cnt)*100, 2))
head(class_per_conyear)

ggplot(class_per_conyear, aes(x = decade, y = freq, fill = epc)) +
  geom_col() +
  labs(title = "EPC Classes per Construction Year",
       y = "Percentage (%)", x = "Decades")


def_per_conyear<-data

for (i in 1:dim(def_per_conyear)[1]){
  if (def_per_conyear$construction_year[i]<=1909){
    decade[i]<-"1900s"
  }
  if (def_per_conyear$construction_year[i]>1909 &
      def_per_conyear$construction_year[i]<=1919){
    decade[i]<-"1910s"
  }
  if (def_per_conyear$construction_year[i]>1919 &
      def_per_conyear$construction_year[i]<=1929){
    decade[i]<-"1920s"
  }
  if (def_per_conyear$construction_year[i]>1929 &
      def_per_conyear$construction_year[i]<=1939){
    decade[i]<-"1930s"
  }
  if (def_per_conyear$construction_year[i]>1939 &
      def_per_conyear$construction_year[i]<=1949){
    decade[i]<-"1940s"
  }
  if (def_per_conyear$construction_year[i]>1949 &
      def_per_conyear$construction_year[i]<=1959){
    decade[i]<-"1950s"
  }
  if (def_per_conyear$construction_year[i]>1959 &
      def_per_conyear$construction_year[i]<=1969){
    decade[i]<-"1960s"
  }
  if (def_per_conyear$construction_year[i]>1969 &
      def_per_conyear$construction_year[i]<=1979){
    decade[i]<-"1970s"
  }
  if (def_per_conyear$construction_year[i]>1979 &
      def_per_conyear$construction_year[i]<=1989){
    decade[i]<-"1980s"
  }
  if (def_per_conyear$construction_year[i]>1989 &
      def_per_conyear$construction_year[i]<=1999){
    decade[i]<-"1990s"
  }
  if (def_per_conyear$construction_year[i]>1999 &
      def_per_conyear$construction_year[i]<=2009){
    decade[i]<-"2000s"
  }
  if (def_per_conyear$construction_year[i]>2009 &
      def_per_conyear$construction_year[i]<=2019){
    decade[i]<-"2010s"
  }
}
head(decade)
def_per_conyear<-as_tibble(data.frame(cbind(def_per_conyear,decade)))
def_per_conyear<- def_per_conyear %>%
  group_by(decade, default) %>%
  summarise(cnt = n()) %>%
  mutate(freq = round(cnt/sum(cnt)*100, 2))
head(def_per_conyear)

ggplot(def_per_conyear, aes(x = decade, y = freq, fill = default)) +
  geom_col() +
  geom_text(aes(label = freq), vjust = -1) +
  labs(title = "Default Rate per Construction Year",
       y = "Percentage (%)", x = "Decades")


#create new dataframe with only numerical variables
corr_data<-subset(data, select=-c(ID, date_contract_begin, date_contract_end, periodi
city,
                                  epc, property_region, residence_region, property_st
atus,
                                  cadastral_category, perf_default_date, default,
                                  region_macroarea, HPI, HPI_chng, GDP, inflation,
                                  mtgrt, unemployment))


res<-rcorr(as.matrix(corr_data))
res
corrplot(res$r, type="upper", order="hclust",
         p.mat = res$P, sig.level = 0.01, insig = "blank")

#high correlation between loan_length, no_of_instalments (and periodicity)
data<-subset(data, select=-periodicity)
#high correlation between property_value, loan_amount and ltv (use only loan_amount
#and ltv in models)

# PERFORMING INDEPENDENCE TESTS #

epc_def<-table(data$epc,data$default)
epc_def_chitest<-chisq.test(epc_def)
epc_def_chitest #p-value < 2.2e-16, there's dependence
sum(data$default=="No")

PStatus_def<-table(data$property_status,data$default)
PStatus_def_chitest<-chisq.test(PStatus_def)
PStatus_def_chitest #p-value = 0.07214, no dependence at 5%

CadCat_def<-table(data$cadastral_category, data$default)
CadCat_def_test<-chisq.test(CadCat_def)
CadCat_def_test #Even though p-value = 7.063e-11, the chi-squared is not reliable
                #(loft, villa/villino and villetta a schiera default equal to 0,1 and
1)

RegMA_def<-table(data$region_macroarea,data$default)
RegMA_def_test<-chisq.test(RegMA_def)
RegMA_def_test #p-value = 3.835e-06, there's dependence (dig deeper with region)


Reg_def<-table(data$property_region, data$default)
Reg_def_test<-chisq.test(Reg_def)
Reg_def_test #Even though p-value = 3.986e-05, the chi-squared is not reliable
             #(many regions have less than 5 default occurencies)

ResReg_def<-table(data$residence_region, data$default)
ResReg_def_test<-chisq.test(ResReg_def)
ResReg_def_test #Even though p-value = 7.167e-05, the chi-squared is not reliable
                #(many regions have less than 5 default occurencies)

############################ Models ############################################

# TRANSFORMING DATA TYPE #

data$epc<-as.factor(data$epc)
data$property_region<-as.factor(data$property_region)
data$residence_region<-as.factor(data$residence_region)
data$property_status<-as.factor(data$property_status)
data$cadastral_category<-as.factor(data$cadastral_category)
data$default<-as.factor(data$default)
data$region_macroarea<-as.factor(data$region_macroarea)
relevel(data$default, ref="Yes")

# Train/Test Splitting
set.seed(42)
sample <- sample.int(n = nrow(data), size = floor(.8*nrow(data)), replace = F)
data_train <- data[sample, ]
data_test  <- data[-sample, ]
#keep unbalanced classes, roughy 2.36% defaults for data and each split
```

```
# Logistic Regression #
## MAIN METRIC: SENSITIVITY ##

log_model<-glm(default~loan_length+no_of_instalments+loan_amount+ltv+epc+
                age_borrower_orig+construction_year+score_value+
                 cadastral_category+region_macroarea, family=binomial, data=data_trai
n)
summary(log_model) #move on dropping non-significant variables

log_model2<-update(log_model, .~.-construction_year)
summary(log_model2)
anova(log_model2, test="Chisq")

#enhance model's fitting by merging some categories#
data_train_merge=data_train
data_train_merge$cadastral_category<-as.character(data_train_merge$cadastral_categor
y)
for(i in 1:dim(data_train_merge)[1]){
  if (data_train_merge$cadastral_category[i]=="appartamento" |
      data_train_merge$cadastral_category[i]=="attico/mansarda" |
      data_train_merge$cadastral_category[i]=="loft"){
    data_train_merge$cadastral_category[i]<-"appartamento/attico/loft"
  }
}
data_train_merge$cadastral_category<-as.factor(data_train_merge$cadastral_category)

data_test_merge=data_test
data_test_merge$cadastral_category<-as.character(data_test_merge$cadastral_category)
for(i in 1:dim(data_test_merge)[1]){
  if (data_test_merge$cadastral_category[i]=="appartamento" |
      data_test_merge$cadastral_category[i]=="attico/mansarda" |
      data_test_merge$cadastral_category[i]=="loft"){
    data_test_merge$cadastral_category[i]<-"appartamento/attico/loft"
  }
}
data_test_merge$cadastral_category<-as.factor(data_test_merge$cadastral_category)

log_model4<-glm(default~loan_length+no_of_instalments+loan_amount+ltv+epc+
                  age_borrower_orig+score_value+
                  cadastral_category+region_macroarea, family=binomial, data=data_tra
in_merge)
summary(log_model4)

## Evaluation on Train Set ##

pred_values_log_OnTrain<-predict(log_model4, data_train_merge, type="response")
my_roc_train <- roc(data_train_merge$default, pred_values_log_OnTrain)
metrics_log_train<-coords(my_roc_train, "best", ret ="all")

plot(my_roc_train, print.auc=TRUE) #AUC=0.684

pred_class_log_OnTrain<-ifelse(pred_values_log_OnTrain>as.numeric(metrics_log_train
[1]), "Yes", "No")
conf_matrix_log_Train<-table(pred_class_log_OnTrain, data_train_merge$default)

Accuracy_log_Train=as.numeric(metrics_log_train[4]) #0.5652797
Specificity_log_Train=as.numeric(metrics_log_train[2]) #0.5621005
Sensitivity_log_Train=as.numeric(metrics_log_train[3]) #0.6966527
Threshold_log_Train=as.numeric(metrics_log_train[1]) #0.02093805

## Evaluation on Test Set ##

pred_values_log_OnTest<-predict(log_model4, data_test_merge, type="response")
my_roc_test <- roc(data_test_merge$default, pred_values_log_OnTest)
metrics_log_test<-coords(my_roc_test, "best", ret ="all")
```

```r
plot(my_roc_test, print.auc=TRUE) #AUC=0.690

pred_class_log_OnTest<-ifelse(pred_values_log_OnTest>as.numeric(metrics_log_test[1]),
"Yes", "No")
conf_matrix_log_Test<-table(pred_class_log_OnTest, data_test_merge$default)

Accuracy_log_Test=as.numeric(metrics_log_test[4]) #0.5257278
Specificity_log_Test=as.numeric(metrics_log_test[2]) #0.5201721
Sensitivity_log_Test=as.numeric(metrics_log_test[3]) #0.7563025
Threshold_log_Test=as.numeric(metrics_log_test[1]) #0.01977467


# Attempt with Macroeconomics metrics #
log_model5<-update(log_model4, .~.+HPI_chng+HPI+inflation+mtgrt+unemployment+GDP)
summary(log_model5)


anova(log_model5, test="Chisq")
log_model6<-update(log_model5, .~.-unemployment)
summary(log_model6)
anova(log_model6, test="Chisq")

#Predictions w/ Macroeconomics metrics #
pred_values_log_OnTest_ME<-predict(log_model6, data_test_merge, type="response")
my_roc_test_ME <- roc(data_test_merge$default, pred_values_log_OnTest_ME)
metrics_log_test_ME<-coords(my_roc_test_ME, "best", ret ="all")

plot(my_roc_test_ME, print.auc=TRUE) #AUC=0.792

pred_class_log_OnTest_ME<-ifelse(pred_values_log_OnTest_ME>as.numeric(metrics_log_test_ME[1]), "Yes", "No")
conf_matrix_log_Test_ME<-table(pred_class_log_OnTest_ME, data_test_merge$default)

Accuracy_log_Test_ME=as.numeric(metrics_log_test_ME[4]) #0.6899807
Specificity_log_Test_ME=as.numeric(metrics_log_test_ME[2]) #0.6882308
Sensitivity_log_Test_ME=as.numeric(metrics_log_test_ME[3]) #0.762605
Threshold_log_Test_ME=as.numeric(metrics_log_test_ME[1]) #0.02639929

#Data handling to prepare for advanced models

def_train<-data_train$default
data_train<-subset(data_train, select=-default)
data_train<-data.frame(cbind(data_train,def_train))
colnames(data_train)[26]<-"default"


def_test<-data_test$default
data_test<-subset(data_test, select=-default)
data_test<-data.frame(cbind(data_test,def_test))
colnames(data_test)[26]<-"default"

data_test_ridge<-ifelse(data_test$default=="Yes",1,0)


X_train<-model.matrix(object= default~loan_length+no_of_instalments+loan_amount+ltv+
                        epc+age_borrower_orig+construction_year+score_value+
                        cadastral_category+region_macroarea+HPI_chng+HPI+inflation+
                        mtgrt+unemployment+GDP,data_train)[,-1]
X_test<-model.matrix(object= default~loan_length+no_of_instalments+loan_amount+ltv+
                       epc+age_borrower_orig+construction_year+score_value+
                       cadastral_category+region_macroarea+HPI_chng+HPI+inflation+
                       mtgrt+unemployment+GDP,data_test)[,-1]
y_train<-ifelse(data_train$default=="Yes",1,0)
```

```
# Logistic Regression with Ridge #

set.seed(42)
cv.ridge<-cv.glmnet(X_train, y_train, alpha=0, family="binomial")
ridge_model<-glmnet(X_train, y_train, alpha=0, family="binomial", lambda=cv.ridge$lam
bda.min)

prob_ridge<-ridge_model %>% predict(newx= X_test, type="response")
my_roc_test_ridge <- roc(data_test_ridge, prob_ridge)
metrics_log_test_ridge<-coords(my_roc_test_ridge, "best", ret ="all")

plot(my_roc_test_ridge, print.auc=TRUE) #AUC=0.791

pred_class_log_OnTest_ridge<-ifelse(prob_ridge>as.numeric(metrics_log_test_ridge[1]),
"Yes", "No")
conf_matrix_log_Test_ridge<-table(pred_class_log_OnTest_ridge, data_test_ridge)

Accuracy_log_Test_ridge=as.numeric(metrics_log_test_ridge[4]) #0.6406999
Specificity_log_Test_ridge=as.numeric(metrics_log_test_ridge[2]) #0.6365477
Sensitivity_log_Test_ridge=as.numeric(metrics_log_test_ridge[3]) #0.8130252
Threshold_log_Test_ridge=as.numeric(metrics_log_test_ridge[1]) #0.0223043

# Logistic Regression with LASSO #

set.seed(42)
cv.lasso<-cv.glmnet(X_train, y_train, alpha=1, family="binomial")
lasso_model<-glmnet(X_train, y_train, alpha=1, family="binomial", lambda=cv.lasso$lam
bda.min)

prob_lasso<-lasso_model %>% predict(newx= X_test, type="response")
my_roc_test_lasso <- roc(data_test_lasso, prob_lasso)
metrics_log_test_lasso<-coords(my_roc_test_lasso, "best", ret ="all")

plot(my_roc_test_lasso, print.auc=TRUE) #AUC=0.791

pred_class_log_OnTest_lasso<-ifelse(prob_lasso>as.numeric(metrics_log_test_lasso[1]),
"Yes", "No")
conf_matrix_log_Test_lasso<-table(pred_class_log_OnTest_lasso, data_test$default)

Accuracy_log_Test_lasso=as.numeric(metrics_log_test_lasso[4]) #0.6863724
Specificity_log_Test_lasso=as.numeric(metrics_log_test_lasso[2]) #0.6844849
Sensitivity_log_Test_lasso=as.numeric(metrics_log_test_lasso[3]) #0.7647059
Threshold_log_Test_lasso=as.numeric(metrics_log_test_lasso[1]) #0.02632447


# Linear Discriminant Analysis #

data_test_lda<-ifelse(data_test$default=="Yes",1,0)

lda_model<-lda(default~loan_length+no_of_instalments+loan_amount+ltv+
                 epc+age_borrower_orig+construction_year+score_value+
                 cadastral_category+region_macroarea+HPI_chng+HPI+inflation+
                 mtgrt+unemployment+GDP, data=data_train)
lda_model

prob_lda<-predict(lda_model, newdata= data_test)
prob_lda_class01<-ifelse(prob_lda$class=="Yes",1,0)
my_roc_test_lda <- roc(data_test_lda, prob_lda$posterior[,2])
metrics_log_test_lda<-coords(my_roc_test_lda, "best", ret ="all")

plot(my_roc_test_lda, print.auc=TRUE) #AUC=0.787

pred_class_log_OnTest_lda<-ifelse(prob_lda$posterior[,2]>as.numeric(metrics_log_test_
lda[1]), "Yes", "No")
conf_matrix_log_Test_lda<-table(pred_class_log_OnTest_lda, data_test$default)
```

```
Accuracy_log_Test_lda=as.numeric(metrics_log_test_lda[4]) #0.6967525
Specificity_log_Test_lda=as.numeric(metrics_log_test_lda[2]) #0.6953176
Sensitivity_log_Test_lda=as.numeric(metrics_log_test_lda[3]) #0.7563025
Threshold_log_Test_lda=as.numeric(metrics_log_test_lda[1]) #0.02339972

# Random Forest #

data_test_RF=data_test_lda

set.seed(42)  # Setting seed
classifier_RF = randomForest(x = X_train,
                             y = y_train,
                             ntree = 50)

classifier_RF

# Predicting the Test set results
y_pred = predict(classifier_RF, newdata = X_test)

my_roc_test_RF <- roc(data_test_RF, y_pred)
metrics_log_test_RF<-coords(my_roc_test_RF, "best", ret ="all")

plot(my_roc_test_RF, print.auc=TRUE) #AUC=0.787

pred_class_log_OnTest_RF<-ifelse(y_pred>as.numeric(metrics_log_test_RF[1]), "Yes", "N
o")
conf_matrix_log_Test_RF<-table(pred_class_log_OnTest_RF, data_test$default)

Accuracy_log_Test_RF=as.numeric(metrics_log_test_RF[4]) #0.7689684
Specificity_log_Test_RF=as.numeric(metrics_log_test_RF[2]) #0.7733232
Sensitivity_log_Test_RF=as.numeric(metrics_log_test_RF[3]) #0.5882353
Threshold_log_Test_RF=as.numeric(metrics_log_test_RF[1]) #0.03666667

# Plotting model
plot(classifier_RF)

# Importance plot
importance(classifier_RF)

# Variable importance plot
varImpPlot(classifier_RF)



########### HIGHEST ACCURACY: Random Forest ##################
########### HIGHEST SENSITIVITY: Logistic Regression with Ridge Penalization


################   Energy Efficiency Evaluation #################
#Creating new dataset with EE variable, dropping almost all other variables and
#adding censoring dummy
data_EEE<-data
EE<-ifelse(data_EEE$epc=="A",1,0)
data_EEE<-data.frame(cbind(data_EEE,EE))
data_EEE$EE<-as.factor(data_EEE$EE)
data_cox<-data_EEE
data_cox1<-subset(data_EEE, select=-c(property_value, date_contract_end, ltv,
                                     loan_length,no_of_instalments, loan_amount,
                                    property_region, residence_region,
                                     age_borrower_orig, construction_year,
                                     property_status, delta_val_cntr, score_value,
                                     cadastral_category, HPI_chng,
                                     HPI, GDP, region_macroarea, inflation, mtgrt, un
employment))
censored<-ifelse(data_cox1$perf_default_date=="2019-12-31",1,0)
head(censored)
```

```
data_cox1<-data.frame(cbind(data_cox1,censored))
time<-time_length(difftime(data_cox1$perf_default,data_cox1$date_contract_begin),
                  "days")
data_cox1<-data.frame(cbind(data_cox1,time))
data_cox<-data.frame(cbind(data_cox,censored,time))

############## COX Model with our data ###############

cox_model_1<-coxph(Surv(time, censored)~ EE, data=data_cox1)
summary(cox_model_1)
fit <- survfit(Surv(time, censored) ~ EE, data = data_cox1)
ggsurvplot(fit, data = data_cox1)


cox_model_2<-coxph(Surv(time, censored)~ EE+score_value+ltv+loan_length+construction_
year+
                     age_borrower_orig+inflation+unemployment+HPI_chng, data=data_co
x)
summary(cox_model_2)
fit <- survfit(Surv(time, censored) ~ EE, data = data_cox)
ggsurvplot(fit, data = data_cox)

############# COX Model in Billio et al (2020) data ######################

data_cox_chapter<- data_cox_chapter[!(is.na(data_cox_chapter$score_value)),]
data_cox_chapter$date_contract_begin<-as.Date(data_cox_chapter$date_contract_begin)
data_cox_chapter<- data_cox_chapter[!(data_cox_chapter$date_contract_begin<2012-01-0
1),]
data_cox_chapter$ltv<-round(data_cox_chapter$loan_amount/data_cox_chapter$property_va
lue, 2)
data_cox_chapter<- data_cox_chapter[!(is.na(data_cox_chapter$ltv)),]
data_cox_chapter<- data_cox_chapter[!(data_cox_chapter$ltv>1.1),]
sum(data_cox_chapter$property_status=="USAT") #48164
sum(data_cox_chapter$property_status=="SNUOV") #5894
sum(data_cox_chapter$property_status=="RISTR") #9285
sum(data_cox_chapter$property_status=="NUOV") #6762
sum(data_cox_chapter$property_status=="DARIS") #951
data_cox_chapter<- data_cox_chapter[!(is.na(data_cox_chapter$age_borrower_orig)),]
censored<-ifelse(data_cox_chapter$default_date=="",1,2)
data_cox_chapter<-data.frame(cbind(data_cox_chapter, censored))


cox_model_chap<-coxph(Surv(months_since_origination, censored)~ EE_A, data=data_cox_c
hapter)
summary(cox_model_chap)
fit_chap <- survfit(Surv(months_since_origination, censored) ~ EE_A, data = data_cox_
chapter)
ggsurvplot(fit_chap, data = data_cox_chapter)

## Comparing models' performances ##


data_for_perf=data
EE_A =ifelse(data_for_perf$epc=="A",1,0)
EE_ABC=ifelse(data_for_perf$epc=="A" | data_for_perf$epc=="B" | data_for_perf$epc=="C
",
              1,0)
data_for_perf<-data.frame(cbind(data_for_perf,EE_A,EE_ABC))
data_for_perf$EE_A<-as.factor(data_for_perf$EE_A)
data_for_perf$EE_ABC<-as.factor(data_for_perf$EE_ABC)

set.seed(42)
sample_perf <- sample.int(n = nrow(data_for_perf), size = floor(.8*nrow(data_for_per
f)), replace = F)
data_train_perf <- data_for_perf[sample_perf, ]
data_test_perf  <- data_for_perf[-sample_perf, ]
```

```
## Ridge EE_A ##

X_train_perf_A<-model.matrix(object= default~loan_length+no_of_instalments+loan_amoun
t+ltv+
                        EE_A+age_borrower_orig+construction_year+score_value+
                        cadastral_category+region_macroarea+HPI_chng+HPI+inflation+
                        mtgrt+unemployment+GDP,data_train_perf)[,-1]
X_test_perf_A<-model.matrix(object= default~loan_length+no_of_instalments+loan_amount
+ltv+
                       EE_A+age_borrower_orig+construction_year+score_value+
                       cadastral_category+region_macroarea+HPI_chng+HPI+inflation+
                       mtgrt+unemployment+GDP,data_test_perf)[,-1]
y_train_perf_A<-ifelse(data_train_perf$default=="Yes",1,0)


set.seed(42)
cv.ridge_A<-cv.glmnet(X_train_perf_A, y_train_perf_A, alpha=0, family="binomial")
ridge_model_A<-glmnet(X_train_perf_A, y_train_perf_A, alpha=0, family="binomial", lam
bda=cv.ridge_A$lambda.min)

prob_ridge_A<-ridge_model_A %>% predict(newx= X_test_perf_A, type="response")
my_roc_test_ridge_A <- roc(data_test_perf$default, prob_ridge_A)
metrics_log_test_ridge_A<-coords(my_roc_test_ridge_A, "best", ret ="all")

plot(my_roc_test_ridge_A, print.auc=TRUE) #AUC=0.789

pred_class_log_OnTest_ridge_A<-ifelse(prob_ridge_A>as.numeric(metrics_log_test_ridge_
A[1]), "Yes", "No")
conf_matrix_log_Test_ridge_A<-table(pred_class_log_OnTest_ridge_A, data_test_perf$def
ault)

Accuracy_log_Test_ridge_A=as.numeric(metrics_log_test_ridge_A[4]) #0.7221096
Specificity_log_Test_ridge_A=as.numeric(metrics_log_test_ridge_A[2]) #0.7219944
Sensitivity_log_Test_ridge_A=as.numeric(metrics_log_test_ridge_A[3]) #0.7268908
Threshold_log_Test_ridge_A=as.numeric(metrics_log_test_ridge_A[1]) #0.03145736


## Ridge w/o epc ##

X_train_perf_NO<-model.matrix(object= default~loan_length+no_of_instalments+loan_amou
nt+ltv+
                               age_borrower_orig+construction_year+score_value+
                               cadastral_category+region_macroarea+HPI_chng+HPI+infla
tion+
                               mtgrt+unemployment+GDP,data_train_perf)[,-1]
X_test_perf_NO<-model.matrix(object= default~loan_length+no_of_instalments+loan_amoun
t+ltv+
                              +age_borrower_orig+construction_year+score_value+
                              cadastral_category+region_macroarea+HPI_chng+HPI+inflat
ion+
                              mtgrt+unemployment+GDP,data_test_perf)[,-1]
y_train_perf_NO<-ifelse(data_train_perf$default=="Yes",1,0)


set.seed(42)
cv.ridge_NO<-cv.glmnet(X_train_perf_NO, y_train_perf_NO, alpha=0, family="binomial")
ridge_model_NO<-glmnet(X_train_perf_NO, y_train_perf_NO, alpha=0, family="binomial",
lambda=cv.ridge_NO$lambda.min)

prob_ridge_NO<-ridge_model_NO %>% predict(newx= X_test_perf_NO, type="response")
my_roc_test_ridge_NO <- roc(data_test_perf$default, prob_ridge_NO)
metrics_log_test_ridge_NO<-coords(my_roc_test_ridge_NO, "best", ret ="all")

plot(my_roc_test_ridge_NO, print.auc=TRUE) #AUC=0.788
```

```
pred_class_log_OnTest_ridge_NO<-ifelse(prob_ridge_NO>as.numeric(metrics_log_test_ridg
e_NO[1]), "Yes", "No")
conf_matrix_log_Test_ridge_NO<-table(pred_class_log_OnTest_ridge_NO, data_test_perf$d
efault)

Accuracy_log_Test_ridge_NO=as.numeric(metrics_log_test_ridge_NO[4]) #0.7206268
Specificity_log_Test_ridge_NO=as.numeric(metrics_log_test_ridge_NO[2]) #0.7204252
Sensitivity_log_Test_ridge_NO=as.numeric(metrics_log_test_ridge_NO[3]) #0.7289916
Threshold_log_Test_ridge_NO=as.numeric(metrics_log_test_ridge_NO[1]) #0.03127743

## Ridge EE_ABC ##

X_train_perf_ABC<-model.matrix(object= default~loan_length+no_of_instalments+loan_amo
unt+ltv+
                                EE_ABC+age_borrower_orig+construction_year+score_valu
e+
                                cadastral_category+region_macroarea+HPI_chng+HPI+infl
ation+
                                mtgrt+unemployment+GDP,data_train_perf)[,-1]
X_test_perf_ABC<-model.matrix(object= default~loan_length+no_of_instalments+loan_amou
nt+ltv+
                               EE_ABC+age_borrower_orig+construction_year+score_value
+
                               cadastral_category+region_macroarea+HPI_chng+HPI+infla
tion+
                               mtgrt+unemployment+GDP,data_test_perf)[,-1]
y_train_perf_ABC<-ifelse(data_train_perf$default=="Yes",1,0)


set.seed(42)
cv.ridge_ABC<-cv.glmnet(X_train_perf_ABC, y_train_perf_ABC, alpha=0, family="binomial
")
ridge_model_ABC<-glmnet(X_train_perf_ABC, y_train_perf_ABC, alpha=0, family="binomial
", lambda=cv.ridge_ABC$lambda.min)

prob_ridge_ABC<-ridge_model_ABC %>% predict(newx= X_test_perf_ABC, type="response")
my_roc_test_ridge_ABC <- roc(data_test_perf$default, prob_ridge_ABC)
metrics_log_test_ridge_ABC<-coords(my_roc_test_ridge_ABC, "best", ret ="all")

plot(my_roc_test_ridge_ABC, print.auc=TRUE) #AUC=0.789

pred_class_log_OnTest_ridge_ABC<-ifelse(prob_ridge_ABC>as.numeric(metrics_log_test_ri
dge_ABC[1]), "Yes", "No")
conf_matrix_log_Test_ridge_ABC<-table(pred_class_log_OnTest_ridge_ABC, data_test_perf
$default)

Accuracy_log_Test_ridge_ABC=as.numeric(metrics_log_test_ridge_ABC[4]) #0.6478177
Specificity_log_Test_ridge_ABC=as.numeric(metrics_log_test_ridge_ABC[2]) #0.6440395
Sensitivity_log_Test_ridge_ABC=as.numeric(metrics_log_test_ridge_ABC[3]) #0.8046218
Threshold_log_Test_ridge_ABC=as.numeric(metrics_log_test_ridge_ABC[1]) #0.02314406

## Ridge EPC ##

X_train_perf_EPC<-model.matrix(object= default~loan_length+no_of_instalments+loan_amo
unt+ltv+
                                 epc+age_borrower_orig+construction_year+score_value+
                                 cadastral_category+region_macroarea+HPI_chng+HPI+inf
lation+
                                 mtgrt+unemployment+GDP,data_train_perf)[,-1]
X_test_perf_EPC<-model.matrix(object= default~loan_length+no_of_instalments+loan_amou
nt+ltv+
                                epc+age_borrower_orig+construction_year+score_value+
                                cadastral_category+region_macroarea+HPI_chng+HPI+infl
ation+
                                mtgrt+unemployment+GDP,data_test_perf)[,-1]
```

```
y_train_perf_EPC<-ifelse(data_train_perf$default=="Yes",1,0)


set.seed(42)
cv.ridge_EPC<-cv.glmnet(X_train_perf_EPC, y_train_perf_EPC, alpha=0, family="binomial
")
ridge_model_EPC<-glmnet(X_train_perf_EPC, y_train_perf_EPC, alpha=0, family="binomial
", lambda=cv.ridge_EPC$lambda.min)

prob_ridge_EPC<-ridge_model_EPC %>% predict(newx= X_test_perf_EPC, type="response")
my_roc_test_ridge_EPC <- roc(data_test_perf$default, prob_ridge_EPC)
metrics_log_test_ridge_EPC<-coords(my_roc_test_ridge_EPC, "best", ret ="all")

plot(my_roc_test_ridge_EPC, print.auc=TRUE) #AUC=0.791

pred_class_log_OnTest_ridge_EPC<-ifelse(prob_ridge_EPC>as.numeric(metrics_log_test_ri
dge_EPC[1]), "Yes", "No")
conf_matrix_log_Test_ridge_EPC<-table(pred_class_log_OnTest_ridge_EPC, data_test_perf
$default)

Accuracy_log_Test_ridge_EPC=as.numeric(metrics_log_test_ridge_EPC[4]) #0.6406999
Specificity_log_Test_ridge_EPC=as.numeric(metrics_log_test_ridge_EPC[2]) #0.6365477
Sensitivity_log_Test_ridge_EPC=as.numeric(metrics_log_test_ridge_EPC[3]) #0.8130252
Threshold_log_Test_ridge_EPC=as.numeric(metrics_log_test_ridge_EPC[1]) #0.0223043

# Models order by Sensitivity: EE_A, NO, EE_ABC, EPC #
# Models order by Accuracy: EPC, EE_ABC, NO, EE_A #
# Models order by AUC: NO, EE_A - EE_ABC, EPC #
```

REFERENCES

Agresti, A. (2015), *Foundations of Linear and Generalized Linear Models*, Wiley, 165-168.

Allen, D. M. (1974), The Relationship between Variable Selection and Data Agumentation and a Method for Prediction, *Technometrics*, 16(1), 125-127.

Billio, M., Costola, M., Pelizzon, L., Portioli, F., Riedel, M., Vergari, D. (Forthcoming 2022), Creditworthiness and buildings' energy efficiency in the mortgage market, *Climate Investing*, ISTE/Wiley, Chapter 14.

An, X., Pivo, G. (2020), Green Buildings in Commercial Mortgage-Backed Securities: The Effects of Leed and Energy Star Certification on Default Risk and Loan Terms, *Real Estate Economics*, Vol. 48, Issue 1, 7-42.

Billio, M., Costola, M., Pelizzon, L., Riedel, M. (2021), Buildings' energy efficiency and the probability of mortgage default: The dutch case, *The Journal of Real Estate Finance and Economics* (May 2021).

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. (1984), *Classification and regression trees*, Wadsworth & Brooks/Cole Advanced Books & Software.

Burt, L., Goldstein, D.B., Leeds, S. (2010), A path towards incorporating energy and transportation costs into mortgage underwriting: Shifting to fact-based analysis, 2010 ACEEE *Summer Study on Energy Efficiency in Buildings*, Chapter 8, 53-64.

Cox, D. R. (1972), Regression models and life-tables, *Journal of Royal Statistical Society. Series B (Methodological)*, 34(2), 187-220.

Cramer, J. S. (2004), The early origins of the logit model, *Studied in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, 35(4), 613-626.

Economidou, M., Todeschi, V., Bertoldi, P. (2019), *Accelerating energy renovation investments in buildings*, Luxembourg: Publications Office of the European Union JRC117816.

Fawcett, T. (2006), An Introduction to ROC Analysis, *Pattern Recognition Letters*, 27(8), 861-874.

Fisher, R. A (1936), The Use of Multiple Measurements in Taxonomic Problems, *Annals of Eugenics*, 7(2), 179-188.

Flach, P. A., Hernandez-Orallo, J., Ferri, C. (2011), A coherent interpretation of AUC as a measure of aggregated classification performance, *Proceedings of the 28th International Conference on Machine Learning,* 657-664.

Ho, T. K. (1995), *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pp. 278-282.

Hoerl, A. E., Kennard, R. W. (1970), Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics*, 12(1), 55-67.

Ionescu, C., Baracu, T., Vlad, G., Necula, H., Badea, A. (2015), The historical evolution of the energy efficient buildings, *Renewable and Sustainable Energy Review*, 49, 243-253.

James, G., Witten, D., Hastie, T., Tibshirani, R. (2014), *An Introduction to Statistical Learning with Applications in R,* Springer, 237-250.

Kaplan, E. L., Meier, P. (1958), Nonparametric estimation for incomplete observations, *Journal of the American Statistical Association*, 53(282), 457-481.

Kaza, N., Quercia, R. G., Tian, C. Y. (2014), Home energy efficiency and mortgages risks, *Cityscape*, 16(1), 279-298.

Pearson, K. (1900), On the criterion that a given system of deviations from the probable in the case of a correlate system of variables is such that it can be reasonably supposed to have arisen from random sampling, *Philosophical Magazine. Series 5,* 50(302), 157-175.

Shalev-Schwartz, S., Ben-David, S. (2014), 18 Decision Trees, *Understanding Machine Learning*, Cambridge University Press.

Stone, M. (1977), An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike’s Criterion, *Journal of Royal Statistical Society. Series B (Methodological)*, 39(1), 44-47.

Tibshirani, R. (1996), Regression Shrinkage and Selection via the lasso, *Journal of Royal Statistical Society. Series B (Methodological)*, Wiley, 58(1), 267-288.

van der Loo, M., de Jorge, E. (2018), *Statistical Data Cleaning with Applications in R*, Hoboken: Wiley.

Youden, W. J. (1950), Index for rating diagnostic tests, *Cancer*, 3, 32-35.

Zancanella, P., Bertoldi, P., Boza-Kiss, B. (2018), *Energy efficiency, the value of buildings and the payment default risk*, Luxembourg: Publications Office of the European Union JRC113215, 15-27.

Zou, H., Hastie, T. (2005), Regularization and Variable Selection via the Elastic Net, *Journal of the Royal Statistical Society, Series B,* 67(2), 301-320.