Credit Risk Prediction - HELOC Case

Bonacci Andrea, Petrillo Pamela, Reitano Alessandro

University of Pisa, Pisa

1 DATA UNDERSTANDING

1.1 Dataset Description

The dataset in question contains information regarding people requesting a Home Equity Banking Credit service, which is a line of credit offered by a bank based on Home Equity (the difference between the market value of a house and the purchase price). The purpose of the work is to find in advance customer who will be considerably late for payments. The subjects considered here are 10459 and for each there are 24 variables, 3 of which are categorical variables while the others are quantitative. When a dataset variable refers to a "trade", it means a credit agreement between the consumer and the lending institution that causes the opening of a credit line.
In detail the columns are:

- **RiskPerformance**: Binary variable that can take the values "Good" or "Bad". The "Bad" value indicates that the customer has been late for payments at least once by at least 90 days in a period of time starting 12 months ago and ending 36 months ago;
- **ExternalRiskEstimate**: It is the target variable of our dataset. It is a quantitative variable that represents in percentage a consolidated version of the risk markers of each customer;
- **MsinceOldestTradeOpen**: Quantitative variable indicating, for each customer, the months passed from the oldest trade;
- **MsinceMostRecentTradeOpen**: Quantitative variable that indicates, for each customer, the months elapsed since the last trade. This variable will be useful to analyze the subjects tendency to request capital on credit;
- **AverageMInFile**: Quantitative variable indicating the average period for which a customer has been observed;
- **NumSatisfactoryTrades**: Quantitative variable that represents the number of trades paid without any delay from the various subjects. Very interesting as it highlights the positive payment history of the subject;
- **NumTrades60Ever2DerogPubRec**: Quantitative variable indicating the number of trades whose payment was made at least 60 days late;
- **NumTrades90Ever2DerogPubRec**: Quantitative variable indicating the number of trades whose payment was made at least 90 days late;
- **PercentTradesNeverDelq**: Quantitative variable indicating the percentage of trades whose payment took place without any delay;
- **MsinceMostRecentDelq**: Quantitative variable that indicates, for each customer, the months elapsed since the last trade whose payment was made late;
- **MaxDelq2PublicRecLast12M**: Categorical variable that indicates through values between 0 and 9, the longest delay in payments occurred in the last 12 months. In particular, values between 0 and 4 show a bad customer behavior, with a delay of more than 30 days; 5 and 6 show an unknown degree of delinquency; 7 indicates subjects who have never been in a state of delinquency; 8 and 9 have an ambiguous meaning but, since there are only 2 values "9" and no value "8" in the dataset, they can be considered as not statistically significant;
- **MaxDelqEver**: Categorical variable that represents the same information as the previous variable but considering the entire period of activity of the subject. In this case the values range from 2 to 9. Values between 2 and 6 show a bad customer behavior; 7 shows an unknown degree of delinquency; 8 indicates subjects who have never been in a state of delinquency; 9 still has an ambiguous meaning but there are no subjects with this value in the dataset;
- **NumTotalTrades**: Quantitative variable indicating the number of total trades of a person;
- **NumTradesOpeninLast12M**: Quantitative variable indicating the number of trades made in the last 12 months;
- **PercentInstallTrades**: Quantitative variable indicating the percentage of trades paid in instalments;

- **MSinceMostRecentInqexcl7days**: Quantitative variable indicating the months passed since the last inquiry on the subject excluding the last 7 days. That exclusion is made in order to remove the inquiries which could be due to a phenomena of price comparison shopping;
- **NumInqLast6M**: Quantitative variable indicating the number of inquiries carried out on the subject in the last 6 months;
- **NumInqLast6Mexcl7days**: Quantitative variable equal to the previous one, but in this case the last 7 days are excluded from the analysis for the same reason as MSinceMostRecentInqexcl7days;
- **NetFractionRevolvingBurden**: Quantitative variable that represents the revolving balance, which is the portion of credit card spending that goes unpaid at the end of a billing cycle, divided by the credit limit;
- **NetFractionInstallBurden**: Quantitative variable representing the installment balance divided by original loan amount;
- **NumRevolvingTradesWBalance**, **NumInstallTradesWBalance** and **PercentTradesWBalance**: These are quantitative variables that compare the elements mentioned in the name of the variable itself to the balance. Since the balance sheet of the subjects is absent in the dataset, these variables cannot be considered as particularly useful;
- **NumBank2NatlTradesWHighUtilization**: Quantitative variable that counts the number of credit cards on a consumer credit bureau report carrying a balance that is at 75% of its limit or greater.

1.2 Data cleaning and data reshaping

With regards to the cleaning of the missing data, we identified 588 people with "-9" value in all the columns. This value is a special character to indicate subjects completely unknown to the bank. Because of this they were excluded from the analysis. The special character "-8" has been identified scattered around the dataset. It indicates trade or inquiries that cannot be used or are invalid, so it can be treated as a missing value. These values were treated differently depending on the number contained by each column. Furthermore the value "-7" indicates the lack of information of a given type depending on the column. In particular, 4664 customers with the "-7" value are found in "MsinceMostRecentDelq" and 1855 in "MSinceMostRecentInqexcl7days". Given the context specified by the 2 columns, such values will not be interpreted as missing values but as a positive feature, as if the customer has never committed any payment delay or has never been subjected to inquiries.
We are now going to identify the interesting labels and any transformation and cleaning works on them. Given the columns described above, we have chosen to delete the following once from the dataset:

- **AverageMInFile**: Deleted because it has a cryptic meaning and has very often inconsistent values with the other variables related to the duration of the relationship;
- **NumTrades90Ever2DerogPubRec**: Very similar to the variable referring to 60 days and also incorporated by it. As a matter of fact, the 2 columns have a high correlation value (about 89%) which means that the analysis of both would be redundant and would lead to a useless increase in the dimensionality of the dataset. We also believe that a delay of 60 days is sufficient to identify a bad financial situation;
- **MaxDelq2PublicRecLast12M**: We believe that the variable that covers the entire period (MaxDelqEver) is more interesting. The 2 variables also have a correlation value of 61%, considered, in this context, sufficient to suggest elimination;
- **NumTradesOpeninLast12M**: For reasons of consistency with the previous case we will consider the information regarding the entire period, expressed by the "NumTotalTrades" column;
- **PercentInstallTrades**: It is considered less significant because, although in general all those in a precarious economic situation require payment in installments, it is not certain that those requesting a payment in installments will have economic problems;
- **NumInqLast6M**: We preferred the variable that considers the same information, but cleaned up of inquiries that probably occurred because of phenomena of Price Comparison Shopping ("NumInqLast6Mexcl7days"). The elimination is also justified by the high degree of correlation between the two variables (that is about 99%);

- **NetFractionInstallBurden**: Eliminated because it has 3418 unusable values ("-8") which, on a data set of 10,000 values, make the substitution of these values unacceptable;
- **NumRevolvingTradesWBalance, NumInstallTradesBalance and PercentTradesBalance**: the columns have been eliminated because they refer to the balance of the subjects, which is not available in the dataset.
- **NumBank2NatlTradesWHighUtilization**: Eliminated as it is of cryptic interpretation, in a context in which data on the balance is not available, and because it has a correlation of 45% with the column "NetFractionRevolvingBurden".

In the remaining interesting columns, we proceeded with the cleaning of the missing values. In particular, the missing values in the "NetFractionRevolvingBurden" (179) and "MsinceOldestTradeOpen"(239) columns have been replaced by the average value in these columns, while those in the "MSinceMostRecentInqexcl7days" (476) and "MsinceMostRecentDelq" (176) columns have been replaced by the Mode since the presence of special characters would have distorted the average. The same treatment was reserved for the 132 values "7" in the "MaxDelqEver" column.
Finally, for a better observation in the analysis phase, on values in the "PercentTradesNeverDelq" column, which contains percentage values, we performed a binning operation in classes containing ranges of 10%.
For similar reasons we performed a semantic binning of the continous variable "NetFractionRevolvingBurden" in the following classes: Low, the subject never went "red" in his bank account or it happened for a very low amount of debt (range 0-40); Medium, range 41-80; High, range 81-300. There are very few people with an amount higher than 100, so we grouped them all in the "high" class.
We have also grouped the variable "MSinceMostRecentDelq" into six-month classes. The value "7" in this grouping is a special value indicating that 37 months or more have passed since the delinquency.

1.3 Data Analysis

Since the target variable is continuous, it's necessary for a meaningfull analysis to discretize it. In order to obtain a good distribution when we compared it to the other variables, we divided ExternalRiskEstimate in 4 different classes: Very Low Risk(87-100%), Low Risk(74-86%), Medium Risk(62-73%), High Risk(0-61%).
Such partition leads to the following distribution:

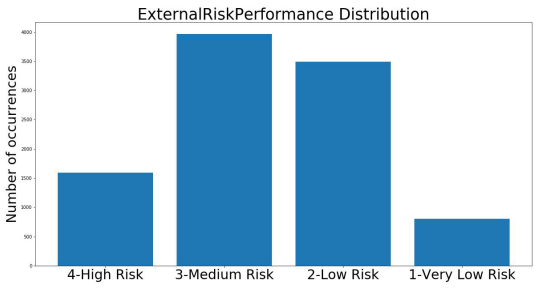


Fig. 1. Risk Estimate Distribution

We immediately identify 2 low frequency classes which, as we will see in the following analysis, contains the "extreme cases", like customers who have never committed a delinquency. Despite the particular origins of our dataset, the distribution is inline with the expectation in a banking context. After a research on the sources of the dataset, we found out that the it has been constructed in order to be almost perfectly balanced on the "RiskPerformance" variable by using a random under-sampling selection. The original dataset is composed of over 242,000 accounts who continued to pay as negotiated in the period observed by the RiskPerformance variable. Only 5,459 accounts made a seriously late payment in the time period and these were all included in the dataset as "bad" accounts, while only a random sample of 5,000 subjects was selected to represent all the "good" accounts.

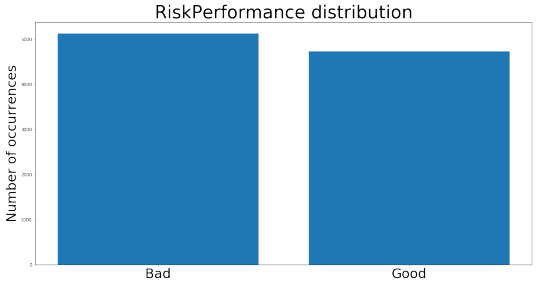


Fig. 2. Risk Performance Distribution of the dataset

Now we are going to analyze some of the most important variables, by the observation of their distribution in relation to the target variable:

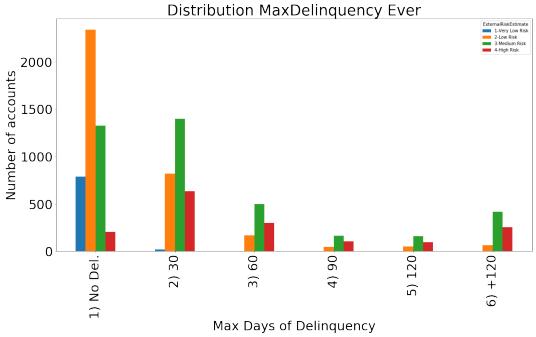


Fig. 3. Distribution of Max Delinquency

From the plot we can see that about half of the accounts considered never performed delinquencies and a quarter of the total had only delays of maximum 30 days. Among these we found all the VeryLowRisk subjects and most of the LowRisks, which present a decreasing trend as the days of delay increase.
Also the majority of MediumRisk is in the first 2 classes but there are about a third of them distributed among the 30+ classes. The same trend is observed by the High Risk but they are distribuited in an almost equivalent way between the classes "30 days or less" and "30+" with maximum frequency in the class "from 0 to 30 days late".
It's interesting to note that there is some HighRisk in the "NoDelinquency" class (about 200 subjects). These customers were probably in an economic situation such as to initially provide few guarantees on the future payment of their debts, but in the end they managed to pay as established. We can say that this distribution is in line with the meaning of these values in a banking context.

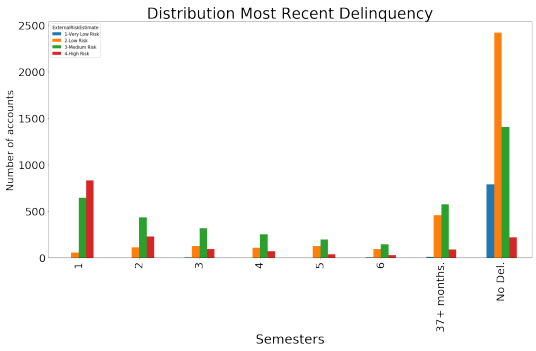


Fig. 4. Distribution of Most Recent Delinquency

Even this plot highlight an expected trend. As we have just seen in the previous image, the VeryLowRisk have never committed a delinquency and, like them, also almost all of the LowRisk. The remaining LowRisks are instead distributed among all the other classes but have the highest frequency in the "37 months and over" class, which confirms their good credit status.
The distribution of HighRisks is also quite predictable, the majority of them is located in the class "less than 1 semester from the last delinquency" but, as shown in the previous plot, 200 of them are among the "NoDelinquency".
MediumRisk subjects present instead a slightly surprising distribution: we note that most of them are in the "NoDelinquency" label, but we can also find many in the "37 month or more" and "1 semester or less" classes. The remaining MediumRisks are distributed in a decreasing trend.

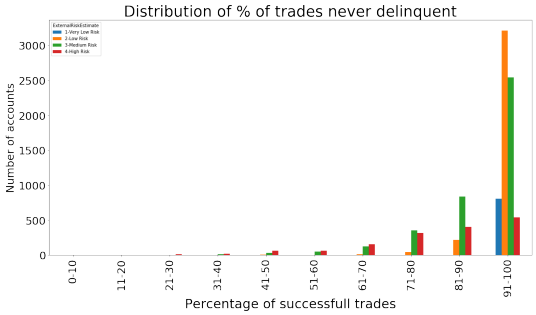


Fig. 5. Distribution of Percentage of Trades Without Delays

The good credit status of the "VeryLowRisk" class is once again confirmed, as every single one of them is in the "91-100%" range.
The trend is completely in line with a healthy bank as we find a constant upward trend for all 4 risk classes, with the peak of the positive classes in the 91-100% range. As expected we also note the preponderance of "HighRisk" on all other classes in the labels between 0 and 70%

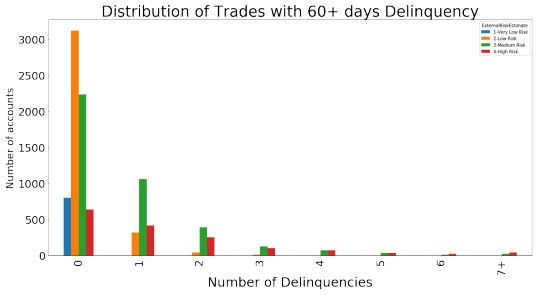


Fig. 6. Trades with 60+ Days of Delinquency

The distribution is in line with the previous observations. We have in fact that most individuals have never committed a delinquency of more than 60 days and among these we find all the "VeryLowRisks" and almost all the "LowRisks" (less than 500 "LowRisk" subjects are distributed in the other classes, in particular in class "1"). This makes sense if one considers that delays of this magnitude are extreme cases and therefore even a good number of subjects considered "High Risk" will not commit such a delay.

We can then observe that all the risk classes present a decreasing trend as the number of Delinquency 60+ increases, with a growing relative superiority of the HighRisk class.

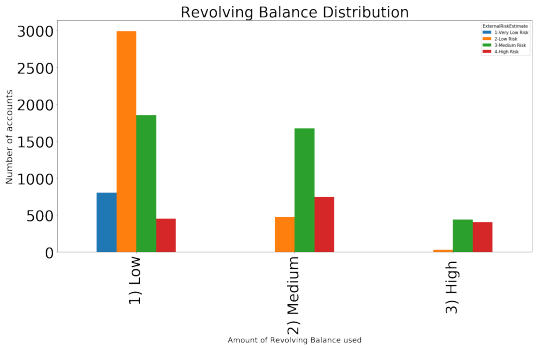


Fig. 7. Distribution of Customer's Revolving Balance

This plot also confirms the expected forecasts. In fact, we have that all the "VeryLowRisks" and most of the "LowRisks" have a "Low" revolving Balance, while the "MediumRisks" have a decreasing trend with a similar frequency in the "Low" and "Medium" classes. The "HighRisks" have a similar distribution in all 3 classes, with a slightly higher frequency among the "MediumRevolvingBalance".
The almost relative superiority of the "HighRisks" in the "High" class, almost equal to the "MediumRisks", is also significant. It can be deduced from all this that a subject with "HighRevolvingBurden" will be extremely problematic.

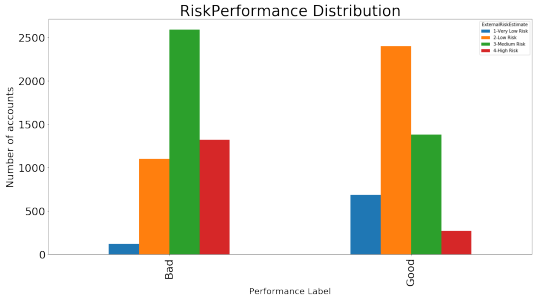


Fig. 8. RiskPerformance Distribution

In this plot we find a trend in line with expectations. Most "MediumRisk" and "HighRisk" are found in the "Bad" label while most "LowRisk" and "VeryLowRisk" are found in the "Good" label.
The presence of low-risk subjects in the "Bad" label is surprising at first sight but can be explained if the significance of the variable is considered, since a single payment with at least 90 days of delay in the recent period is enough to obtain the "Bad" label. Therefore it is not possible to exclude that subjects considered "safe" do not commit heavy delinquencies in extraordinary situations. Specular and reverse situation for the "HighRisks" in the "Good" label.
The relatively similar proportions of "LowRisk" and "Medium Risk" in the 2 labels are most likely due to the particular construction of our dataset, which derives from a totally random undersampling operation.

1.4 Problem formalization

Given the previous observations we can conclude that the problem can be formalized in the construction of a classification model. In the following sections we will create a model that is as understandable and easily explainable as possible. In the construction phase we will focus on obtaining a classifier as accurate as possible in the prediction of the "VeryLowRisk" and "HighRisk" labels. To be more specific, we want to build a classifier that maximizes the recall of the "HighRisk" label and the precision of "VeryLowRisk" while maintaining good values in the other measures. We will therefore test various classifiers to check which ones are the best for our objective (in

particular we will test Decision Tree, Random Forest, K-Neighbors and Naive-Bayes) and we will validate them in various ways, like with a comparison against a proper Dummy Classifier.

2 Classification and Validation

2.1 LinearRegression

Since the original RiskEstimate was a continuous variable, for a first classification operation we proceed to use a Linear Regression algorithm on the original, non discretized, target variable. First of all, we normalized the predictor variables using a Standard Scaler and then we built the train & test with a 70:30 split.
In order to evaluate the quality of the model we calculated the values of R2-Score and Mean Squared Error (respectively 0.7118097 and 27.48807) on our results, and we compared them with the results of a Dummy Regressor:

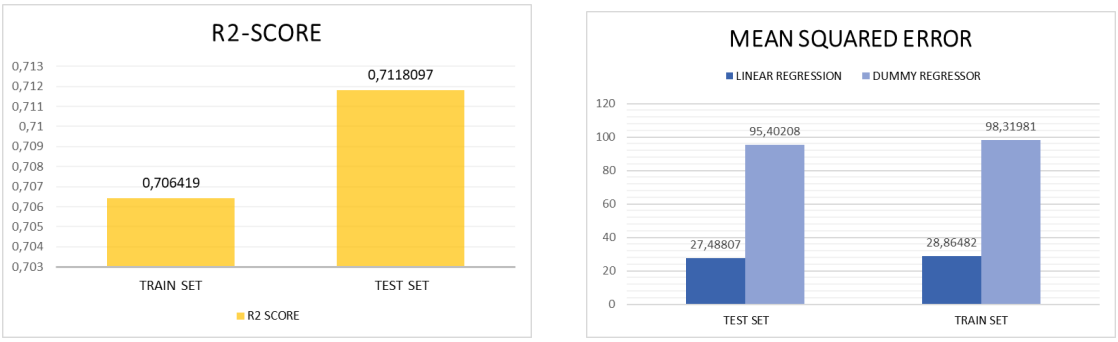


Fig. 9. Train-Test Results Observation and Comparison between Linear Regression and Dummy Regressor

We immediately observe that we obtained good results. In fact, we have the r2 score close to 1 for both the train and the test set. Similarly, the Mean Squared Error of our Linear Regressor is less than a third of what can be achieved with a Dummy Regressor. We can consider our model reliable.
However, we need to think from the perspective of a bank manager, who doesn't care to know the numerical risk percentage of a customer, but wants to know the risk class to which the subject belongs. To satisfy this request we proceeded to the classification on the target variable grouped as in the analysis phase (represented in fig.1).

2.2 Classification on categorical RiskEstimate

In order to evaluate which model provides the best predictions on the risk class while maintaining a good degree of explicability of the results, we proceeded with the creation of various classification models.
Regarding the composition or train and test set, we once again split the dataset in the following way: 70% train and 30% test set.

2.2.1 - Decision Tree

The first classifier we tested is the Decision Tree Classifier. After a Parameter Tuning phase we set our tree with the folowing parameters: MinSplit = 170; MinLeaf = 17; MaxDepth = 11; Criterion = Gini index

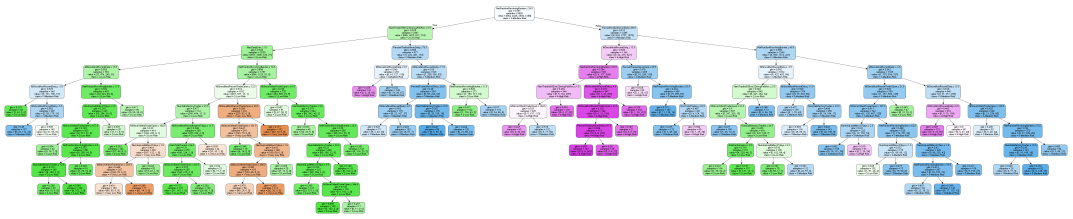


Fig. 10. Decision Tree

Although the structure of the tree may seem very imposing at first sight, it actually makes a clear and easily explainable separation by identifying 3 main branches per side. The first check is done on NetFractionRevolvingBurden; if this is less than or equal to 29.5 (so if the subject belongs to the Low class of our previous grouping) the verification of the subject will move to the left side of the tree where further checks will classify him as MediumRisk (first branch of the left side) or LowRisk / VeryLowRisk. If instead the Revolving balance is greater than 29.5, the verification moves to the right side of the tree where the subjects will be classified as MediumRisk, HighRisk and occasionally LowRisk if they can overcome many other controls.
In addition to RevolvingBurden there are 4 other main variables used for the first separation in the 6 different branches: MSinceMostRecentDelq, MaxDelinquencyEver, NumTrades60Ever2DerogPubRec and PercentTradesNeverDelq. Finally the other variables are used within the specific branches to assign subjects to the various class labels and increase the purity of the splits.
Regards the validation of the results we observe the precision, recall and accuracy values obtained by our model on the test-set and compare them to those of a Dummy Classifier:

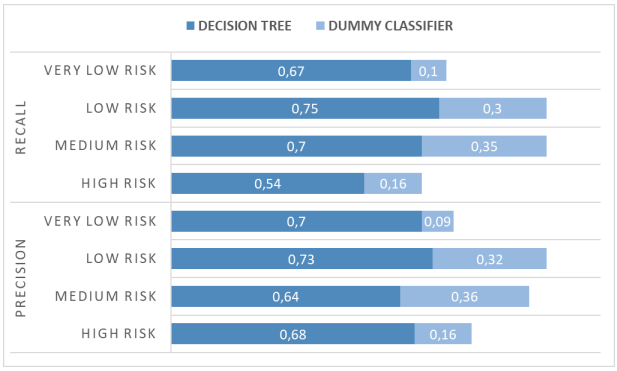


Fig. 11. Decision Tree Validation on Test-Set

From the previous image we can see that all risk classes are relatively balanced among themselves, with no anomalously low values. All the results of our decision tree are also much higher than those obtainable from a stratified Dummy Classifier. This is especially true for our objective measures; in fact we note that the recall of our HighRisk class is more than 4 times greater than that of the Dummy, and the precision of the VeryLowRisk is instead almost 7 times higher. Regarding our objective measures we can see that we obtained overall good values despite the fact that they are the minority classes and the unusual construction of our dataset. Finally, about the Accuracy we notice that the decision tree gives a much higher value than the Dummy (respectively 0.68333 is 0.37715)
For completeness of information, we also analyzed the Most Frequent Dummy Classifier, with which we can only verify the quality of our prediction on the majority class:

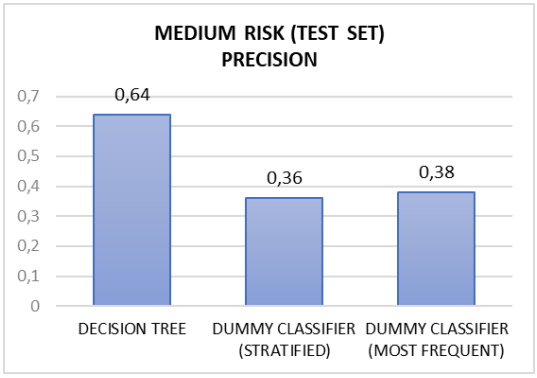


Fig. 12. MediumRisk Goodness

We immediately see that our model performs even better than the Most Frequent Dummy Classifier for the prediction of the majority class.

2.2.2 - Decision Tree Bagging

We continue the analysis by using some ensembling techniques in order to evaluate if they are able to significantly improve our results. We don't report the comparison with the values obtained with the Dummy Classifier because it would be a repetition of what we have already said in the Decision Tree section.
As a first technique we use a general bagging algorithm. Using 100 estimators the results obtained are the following:

	PRECISION	RECALL	F1-SCORE
VERY LOW RISK	0.71	0.63	0.67
LOW RISK	0.75	0.75	0.75
MEDIUM RISK	0.65	0.77	0.71
HIGH RISK	0.76	0.52	0.62
ACCURACY	0.702940		

Table 1.

We observe that we only achieved a slight improvement in general performance. Regarding our objective measures, we note that the precision of VeryLowRisk increases by 0.01 while the HighRisk recall decreases by 0.02. So, in our opinion, the slight improvement achieved at the general level does not justify the increase in complexity in the explanation of the model and the worsening of one of the key values.

2.2.3 - Random Forest

As a second ensemble technique we performed a Random Forest classification algorithm, obtaining the following results:

	PRECISION	RECALL	F1-SCORE
VERY LOW RISK	0.77	0.47	0.58
LOW RISK	0.74	0.78	0.76
MEDIUM RISK	0.65	0.79	0.71
HIGH RISK	0.79	0.49	0.61
ACCURACY	0.705306		

Table 2.

We notice again an increase in general performance as we had just seen for bagging and a significant increase in the precision of VeryLowRisk and HighRisk. This is however counterbalanced by a

significant decrease in the HighRisk recall. Since our work refers to a banking context, we think that the latter fact, combined with an increase in the complexity of explanation even higher than that of bagging, makes this model undesirable.

2.2.4 - Others classifiers: K-Neighbor and Bayes-Naive

Finally, we tested 2 other classifiers in order to observe if it's possible to get better results or if the results are more or less constant. In particular:

K-Neighbor: is the first other classifier that we tested. After a parameter tuning phase we set K = 15, with which we obtain the following results to compare with our Decision Tree:

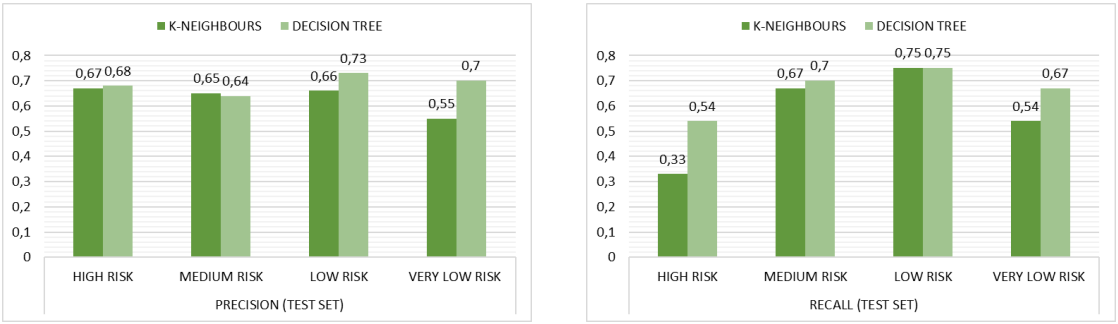


Fig. 13. K-Neighbours vs Decision Tree

As we can see, the only value that the K-Neighbor improves is the precision of the MediumRisk (which increases by 0.01) but there is a worse performance in all the other values both of precision and recall, including our target measures.
Also regarding accuracy, there is a better performance in the decision tree (Accuracy K-Neighbors = 0.64819).

Bayes-Naive: The probabilistic classifier turned up being the model with the worst performance:

	PRECISION	RECALL	F1-SCORE
VERY LOW RISK	0.35	0.96	0.51
LOW RISK	0.60	0.50	0.55
MEDIUM RISK	0.64	0.59	0.61
HIGH RISK	0.54	0.35	0.42
ACCURACY	0.55221		

Table 3.

We observe in fact that the this model has very low results across the board, and especially low ones in our key metrics.

2.3 Choise of the best model

To conclude the validation phase, we carried out a cross validation process on all the previously analyzed classification algorithms:

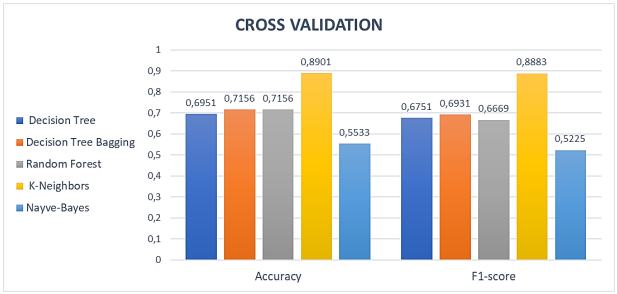


Fig. 14. Cross Validation

From the plot we note that the K-Neighbor shows a value higher than expected both in accuracy and F1-Score. This result could be explained by the interaction between a density-based algorithm, such as the K-Neighbors, and a cross-validation operation on a small dataset. We prove this hypothesis by executing that algorithm again with a K lower than the optimal one (K = 5) and, as reported in the following image, the values have been reduced to the expected levels.

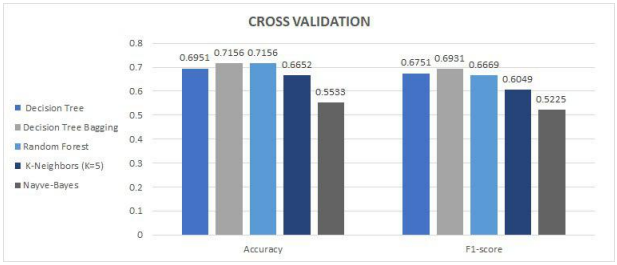


Fig. 15. Cross Validation with Modified K-Neighbours

So, considering all the analyzed elements we can say that the best general results are undoubtedly provided by the Random Forest algorithm. However, due to the considerable decrease in one of the objective metrics and the much higher explanation complexity of this model, we choose the standard Decision Tree as our optimal model.

2.4 Optimization of the best model

After having identified the optimal model in the decision tree, we decided to undertake an optimization process. To this end we have added a cost matrix in which we have given greater weight to the main labels, namely "Very low risk" and "High Risk", to ensure that the Decision Tree pays more attention to errors made on these labels, with the aim of maximizing the recall of "HR", without incurring intolerable losses of its precision and without affecting the precision of "VLR". We then proceeded to select the most suitable weights. First of all we have observed how the values of the key metrics vary, increasing separately the single weights of VLR and HR in a range [1.0, 3.0] and leaving constant those of LR and MR, obtaining the following results:

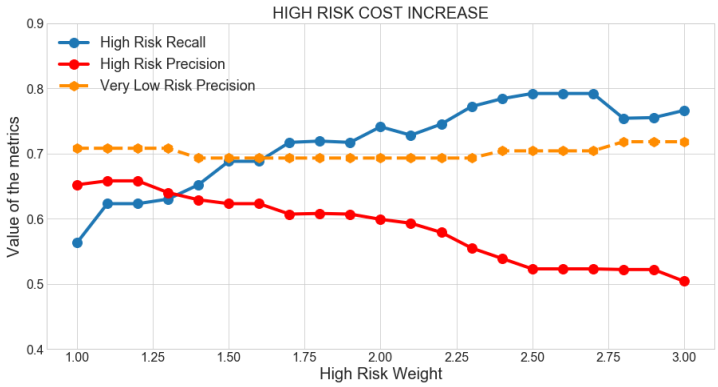


Fig. 16.

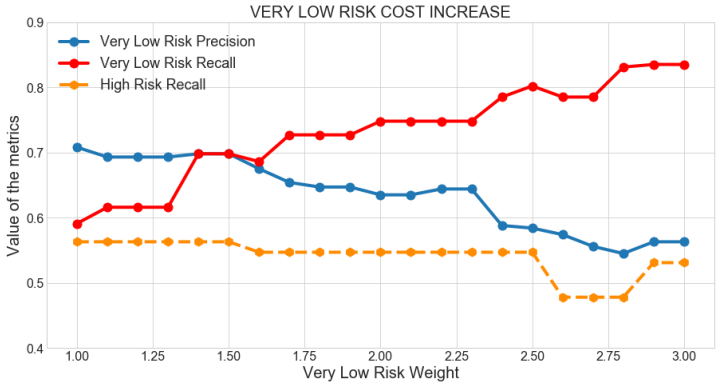


Fig. 17.

As we can see from the images, the weight that maximizes the recall of "HR" is about 2.5, however this value causes a significant decrease in the value of precision, bringing it to almost unacceptable levels. for this reason, the most appropriate value is 2.0, which achieves a significant increase in the recall while maintaining the accuracy around 60%. It is also possible to notice how setting a high difference between the weights of "VLR" and "HR" causes a slight decrease in the precision of "VLR", although for extreme values the pattern is reversed.

As far as "VLR" is concerned, the optimal value is around 1.5, as it obtains a good increase in recall in exchange for an almost imperceptible decrease in precision. Exceeding this value could also negatively affect the value of "HR". To analyze some of the possible permutations of the values of the weights, let's now observe the changes undergone by precision and recall in the case of simultaneous variation of the weights of both classes of risk. In particular, we observe what happens by simultaneously increasing the weights of both classes.

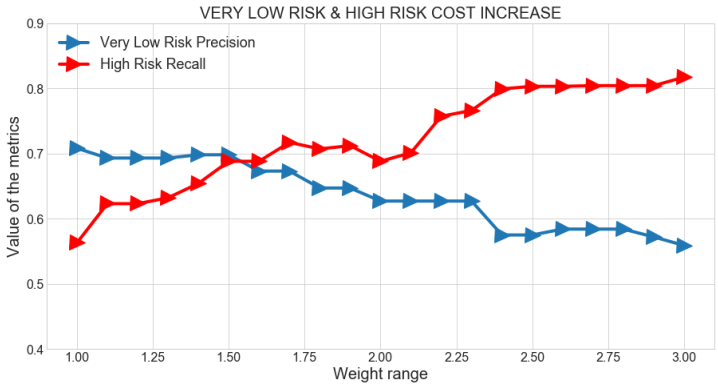


Fig. 18.

We can see how "VLR" is little affected by the various permutations with "HR", while the recall of "HR" is negatively affected by a simultaneous excessive growth of the weight of "VLR". This new analysis again confirms the optimal values previously identified. We will then set the weight of "HR" to 2.0 and the weight of "VLR" to 1.5 because, fixed at that value, "VLR" manages not to be negatively influenced by "HR" and at the same time not to compromise the gains of the latter in terms of recall. With these weights and MinSplit = 115 we can therefore obtain the following values in the test set:

	PRECISION	RECALL	F1-SCORE
VERY LOW RISK	0.70	0.70	0.70
LOW RISK	0.75	0.73	0.74
MEDIUM RISK	0.66	0.60	0.63
HIGH RISK	0.60	0.74	0.66
ACCURACY		0.68064	

Table 4.

Compared to the previous tree we get an increase of 0.20 in the recall of "HR" and 0.03 in the recall of "VLR". The precision of "VLR" has remained unchanged while that of "HR" has decreased by 0.08, which is an acceptable decrease considering the remarkable gain in recall. As far as the other classes are concerned, there is a slight increase in their recall, accompanied by relatively low decreases (maximum 0.1) in precision. Accuracy, on the other hand, remains substantially unchanged.
As far as the structure of the Decision Tree is concerned, it has remained substantially unchanged, having undergone only slight variations in the leaves located at greater depths and in the values of some splits.

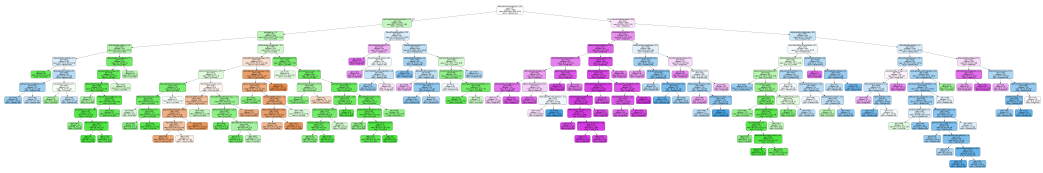


Fig. 19. Weighted Decision Tree

3 Explainability

In order to try to further reduce the complexity of the model, we decided to try to simplify it by reducing the number of splits. This was done by modifying the value of the "Min Split" parameter

of the tree, forcing it to finish its analysis earlier. In order to identify the best value, we have observed the variation in the values of accuracy and the key metrics as this parameter changes:

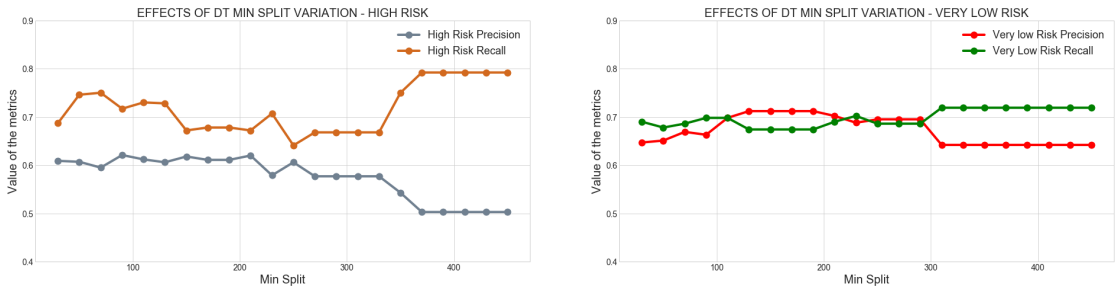


Fig. 20. Effects of MinSplit Variation on Key Metrics

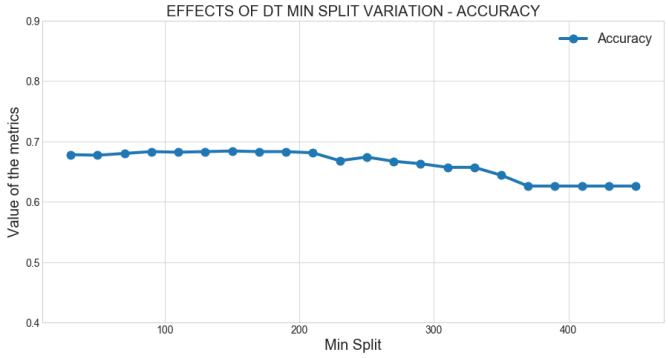


Fig. 21. Effects of MinSplit Variation on Accuracy

The graphs identify 400 as the ideal "MinSplit" value for simplification, a value beyond which the metrics no longer vary significantly. Once this parameter has been replaced, a tree with considerably reduced dimensions is obtained:

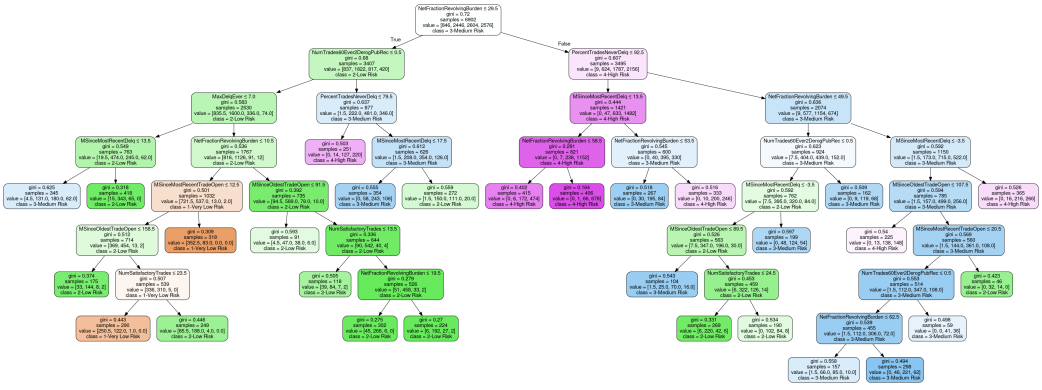


Fig. 22. Simplified Decision Tree

Let's now look at the values obtained from this tree, compared with those obtained from the previous one:

	PRECISION		RECALL		F1-SCORE	
	Min.Split 400	Min.Split 115	Min.Split 400	Min.Split 115	Min.Split 400	Min.Split 115
VERY LOW RISK	0.64	0.70	0.72	0.70	0.68	0.70
LOW RISK	0.73	0.75	0.70	0.73	0.71	0.74
MEDIUM RISK	0.63	0.66	0.46	0.60	0.53	0.63
HIGH RISK	0.50	0.60	0.79	0.74	0.62	0.66

ACCURACY	Min.Split 400	0.62622	Min.Split 115	0.68064

We can therefore observe a minimum loss of accuracy (equal to 0.055) while, with regard to key metrics, there is a slight increase in the recall of "High Risk" in exchange for a slight loss in the Precision of "VeryLowRisk". The new tree is therefore a generally acceptable alternative for those who are looking for maximum explainability. Let us now proceed to a more in-depth analysis of this last tree.

Global Explainability

Let's analyze the behavior of our tree: the first variable on which the tree is based is "NetFractionRevolvingBurden", thus giving a fundamental importance to the amount of any debt, because those who have a medium-high value will almost certainly not be "VLR". Based on the previous split, the tree is divided into 2 sections, having the left one composed of 2 macro-branches and the right one of 3.

The left section will check the number of trades delayed by more than 60 days and if different from 0 will classify these customers "Medium Risk", with a few exceptions of "HR" and "LR".
If, on the other hand, the client has never paid at least 60 days late, the tree will use the duration of the worst delinquency as an additional classification parameter. If the customer has not committed any delinquency, it will be classified as "VLR" with some exceptions of "LR".
If, however, the maximum duration is 30 days, implying that the number of delinquencies is different from 0, the months that have passed since the last delinquency will be observed; if more than 14 months have passed, the customer will be classified as "Low Risk", while if less than 14 months have passed, it will be classified as "Medium Risk"

Turning now to the right section, the first check is made on the percentage of trades concluded without any delinquency. If this is less than 93%, we will go down in the macro-branch that will contain almost all the "HR" clients and a few hundred "MR".
Otherwise, the tree will then perform another check on the "Revolving Burden" to see if the debt settles at a medium or high level. If the "Revolving Burden" is medium, we fall into the second macro-branch, which is the one with the most complex tree composition, having a heterogeneous mix of "LR" and "MR".
If, however, the debt is high, the tree will go to check the months passed since the last delinquency; if these subjects have not committed delinquencies, we will go down into the macro-branch, where multiple checks will be carried out that will lead to their classification mainly as "MR" and occasionally as "HR". Failed all the previous controls, the remaining subjects will all be classified as "HR".

We can conclude that this simplified model manages to achieve acceptable results, however the leaves of its 'tree are much more impure than the previous one and some splits are far too simplistic even for the most important classes. For example, almost all "HR" subjects are classified essentially only through a control on the "Revolving Burden".

For this reason, if one needs a more detailed inspection on some risk classes (as for the "High Risks" mentioned above) it is advisable to also observe the tree previously developed, since in the left section it has an extra macro-branch, useful for the distinction between leaves "VLR" and "LR", while the first right macro-branch does a much deeper analysis at the level of numbers of splits, able to identify multiple much purer leaves of class "HR" and able to properly separate a good part of the "MR" in appropriate leaves.

Local Explainability

Regarding local explainability, this will be, for the sake of completeness and correctness, carried out on the best performing tree. We have therefore analyzed 6 random instances of our dataset: 2 correctly predicted instances (1 "VLR" and 1 "HR") and 4 erroneously predicted instances (2 "VLR" and 2 "HR"). In particular:

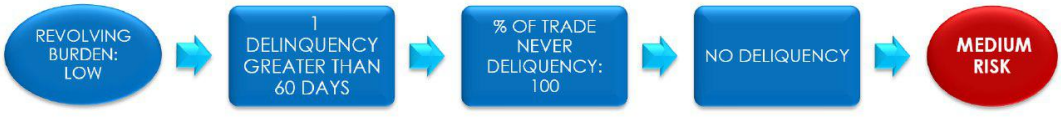
Instance 1 (Real Class: VLR vs Predicted Class: VLR):

We can easily confirm the logic of the tree's decision. A longtime customer who is not in a debt situation, who has never committed a delay in payment and who has never created situations that would require an inquiry against him is clearly a "VLR" subject.

Instance 2 (Real Class: VLR vs Predicted Class: LR):

The decision of our tree is certainly influenced by a particular combination of temporal values: we have in fact that the subject has received an inquiry in the last month and, although in general this is not a clear negative signal, if we consider the recent creation of the last trade, this can be misleading in the classification phase. Despite the apparent misclassification, we still believe that there is no extremely problematic situation, as that branch of the tree contains no values other than "VLR" or "LR" and a customer of this type is clearly eligible to receive an approval of his credit application.

Instance 3 (Real Class:VLR vs Predicted Class: MR):



This classification error most likely results from the presence of noise in the dataset. In fact, we find conflicting information in the splits.

Initially it appears that this customer has paid 1 time late by at least 60 days, but in the next 2 splits is among those who have never made a delay in payment. As a result of this, the decision of the tree to increase the risk class due to conflicting information is considered correct and we leave it to the operator in charge to carry out a more in-depth analysis if deemed necessary.

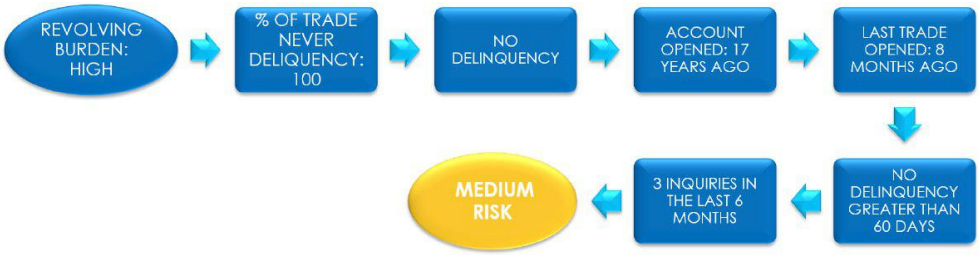
Instance 4 (Real Class:HR vs Predicted Class: HR):

We can observe one of the instances correctly predicted but still not manifestly obvious.

We are dealing with a subject who has been a client of the bank for almost 2 decades who has a percentage of satisfied trades which, although it does not allow him to pass the first test, allows him to pass the second less restrictive internal test, and a value of "Revolving Burden" which makes him fail even a second internal test, but only for a short amount. All of this is combined with the fact that the client, although he committed his last delinquency years ago, has a high number of total trades which, compared to the percentage of successfull trades in the image, indicates that the client has committed a sufficiently high number of actual delinquencies.

It is therefore a case influenced by limit values, but in a banking context a subject so indebted and with such a bad credit history needs a lot of time to recover credibility and can rightly be considered a high risk subject.

Instance 5 (Real Class:HR vs Predicted Class: MR):



It is possible to observe that the extreme value of revolving burden has been balanced, from the point of view of the classifier, from the fact that it is a very active account that was opened almost 20 years ago by a person who has not committed any delinquency in the trades undertaken. The high number of inquiries can therefore be due to the suspicions that such a situation can often arise in the bank. We therefore believe that the choice of the tree to classify this client as a medium risk is at least justifiable.

Instance 6 (Real Class:HR vs Predicted Class: LR):

This instance is very similar to the previous one, however the absence of inquiries by the bank in the last 6 months has prompted the model to classify it as "LR" instead of "MR". We therefore believe that this kind of situation should be assessed more carefully by the appointed employee.

After analyzing all previous cases, we can therefore say that the model provides acceptable explanations or justifications for any of its choices.

3.1 Conclusion

Having carried out all the necessary analyses, we believe that our model is reliable. A bank manager can therefore trust the model because it has good performance and, even in the case of an "error", the deviation from the real class is generally slight and can be justified by several aspects of the credit history of the customer himself, which, depending on the case, could even suggest a shift in the actual class of risk. The only cases that are not fully justifiable are due to really extreme features of the subject or noise in the dataset.

As already mentioned, a general analysis can be carried out on the reduced tree but, with regard to the explanation to be provided to the individual customer, the best thing to do is to analyze the slightly more complex tree, which provides a more accurate explanation of the reasons behind the refusal or acceptance of credit.