Machine Learning Analysis of Credit Card Approval

Paolo Caggiano - Davide Giardini

Contents

Abstract

In the financial world, the credit risk is an important aspect, on the base of which banks decide to grant a credit card or not. This decision is made on the basis of several factors. The goal of this research is the identification of those clients that, based on their financial history, could end up being a bad investment for the bank. in this paper, we want to investigate and understand which are the variables that have an important role in determining if a loan is payed off on time by a client. This objective is carried forward through classification, one of the main techniques utilized in Machine Learning.

1 Introduction

Many financial institutions are providing cashless means for their users like debit and credit cards. Most people rely on them to perform their transaction activities as it is a very easy way of making their payments. A recent study by the Federal Reserve Bank of San Francisco found that in 2021 credit cards were used to make 28 % of all payments [1]. Due to the increasing number of credit card users it has become crucial for banks to differentiate between good and bad costumers. Many financial institutions like national and private banks rely on consumers' information like their basic details, living standards, salary, yearly and monthly returns or their current income source. This complete check and analysis can avoid to the institutions bearing a lot of technical and non-technical losses. Even though decision-making differs from bank to bank, the most common factor considered by financial institutions is the consumer's credit score [2]. Some banks have developed automatic credit approval methods for granting credits or loans to customers[3]. Recently, credit scoring techniques have been expanded to include more applications in different fields. By the start of the 21st century, the use of credit scoring had expanded more and more, especially with new technologies, introducing more advanced techniques and evaluation criteria, such as GINI and the ROC curve. The quality of bank loans is the key determinant of competition, survival and profitability[4].
In this work we have used a data-set coming

from the Kaggle platform [1] to compare the performances of different classification methods in identifying "good" and "bad" clients.

2 The Datasets

The data coming from Kaggle is composed of two datasets. The first one, named "application record" contains all the approved credit cards of the bank. For each observation, numerous information are provided about the owner of the card: their gender, if they own a car, if they own a realty, the number of their children, their total income, their type of income, their education, their family status, their housing type, their age, since how long they have been employed, if they have a mobile phone, work phone, phone or e-mail, their occupation and the number of their family members.
The second dataset, named "credit record", contains, for each credit card, a monthly history of the status of the loan's payment. In particular, the status can assume 8 different levels:

- 0: 1-29 days past due
- 1: 30-59 days past due
- 2: 60-89 days past due
- 3: 90-119 days past due
- 4: 120-149 days past due
- 5: overdue or bad debts, write-offs for more than 150 days
- C: paid off that month
- X: no loan for that month

First of all, we begin by transforming the levels from 1 to 5 into "1", that will identify a bad payment record (status) for that month, while transforming 0, X and C into "0", that will identify a good payment record (status) for that month. Then, we computed the mean of all the newly created status of the same credit card. Since the attribute is binary, computing the mean will result in identifying the percentage of months in which the loan's payment has been delayed.
We now have to differentiate the good and bad credit cards, based on the percentage of months in which the payment is delayed more than 30 days. Our work is developed in such a way to give each bank the possibility of choosing the best threshold based on its aversion to risks. For this research we used a limit of 10%. This means that credit cards that have more than 10% of months in which the loan was paid in delay are considered a "bad" ("1") investment for the bank.
We now proceed to join the two datasets, connecting every credit card to its owner. Then, we aggregate into the same record all the credit cards owned by the same person, computing the average of the credit cards status (bad or good: 1 or 0). We now have to differentiate between a good and bad client based on the number of bad credit cards that he owns. Like before, each bank is able to decide autonomously a specific threshold, while we have fixed for this research a limit of 15%. This means that users that have more than 15% of credit cards considered "bad" are considered a "bad" ("1") client for the bank. The resulting dataset is composed of 9728 observations. Of this clients, 8716 are classified as good costumers, and 1012 as bad costumers. As common in this type of problems, the dataset is unbalanced, this will require the implementation of several techniques to manage the problem.

3 Data Preparation

In this step we use different techniques to prepare the dataset for the classification. First of all we remove the attributes that we think are useless in predicting a customer's status. Along with the ownership of a work/mobile phone and of an email, we decide to remove also the gender to avoid discrimination in the classification process.

3.1 Handling Missing Values

The provided dataset presented 3001 missing values, all in the "occupation type" attribute. We soon notice that in the majority of cases this occurs because retired users have no "occupation type". We therefore add the "pensioner" level to all those users that show the "pensioner" quality in the "income type" column. Doing this, the number of missing value is lowered to 1286.
For this remaining missing values we decide to use a global constant equal to "not specified". We make this decision since we think that the other techniques are not convenient in our case: deleting the records would lead to a very substantial decrease in the dimensionality of the dataset, while the mode replacement

[1] https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction

would cause a substantial increase of the dimensionality of the most frequent occupation type.

3.2 Preprocessing

Given that many classification algorithms don't support categorical variables, we use the *one-hot-encoding* procedure to create one dummy column for every level of the categorical variables, namely: income type, education type, family status, housing type and occupation type. Lastly, to avoid different scales of the data, we normalize the continuous attributes, namely: income, days of birth and days employed.

4 Classification Algorithms

In our research we use five classification algorithms.

- *Logistic Regression*
  Logistic Regression is a classification technique used in machine learning that uses a logistic function to model the dependent variable. The dependent variable is dichotomous in nature. As a result, this technique is used while dealing with binary data.

- *Random Forest*
  Random forest consists of a large number of individual decision trees that operate together. Each individual tree in the random forest outputs a class prediction. The class with the most votes becomes our model's prediction.

- *MultiLayer Perceptron*
  A MultiLayer Perceptron (MLP) is an artificial networks composed of input neurons, hidden neurons and output neurons, that communicate unidirectionally, from the input attributes to the Class Attribute.

- *Naive Bayes*
  Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

- *Naive Bayes Tree*
  Naive Bayes Tree uses decision tree as the general structure and deploys naive Bayesian classifiers at leaves.

5 Performance measures

To evaluate the performances of the five algorithms, we take into account six different tools and measures.

5.1 Confusion Matrix

In the classification problem, a very useful tool in performance evaluation is the confusion matrix, since most of the metrics can be obtained analytically from it. This matrix is made up of rows and columns where the real and expected classes are indicated; in this way it is possible to evaluate the four combinations deriving from the classification, i.e: True Negatives, False Positives, False Negatives and True Positives.

5.2 Accuracy

Accuracy is one of the most known metrics in the field. It refers to the fraction of instances correctly classified. Analytically:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}; \qquad (1)$$

However it is not a reliable measure in the cases of unbalanced class.

5.3 Recall

Recall is equal to:

$$Recall = \frac{TP}{P}, \qquad (2)$$

it indicates the fraction of instances of positive class that are correctly identified.
This measure is the one on which we are going to focus in our analysis: since not identifying a bad costumer (and approving his credit card) costs to the bank more than not identifying a good costumer (and not approving his credit card), the algorithms that will be able to identify the most percentage of bad costumers will be considered the best. Of course, this does not mean that it will be the only measure important in the analysis, given that an unacceptable algorithm could achieve the maximum score of Recall simply without not approving a credit card to anyone.

5.4 Precision

Precision is equal to:

$$Precision = \frac{TP}{TP + FP}; \qquad (3)$$

it represents the fraction of instances which are classified as positive and that result to be effectively positive.
This measure is perfect to be put side by side with the Recall, as it controls how many of the clients to which the algorithm didn't approve the credit card were in fact bad costumers. In this way, it resolves the problem just described: the unacceptable algorithm that could achieve maximum recall by classifying everyone as bad will get a very low precision score.

5.5 F-score

F-score is equal to:

$$F - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}; \quad (4)$$

in other words, it is an harmonic mean between Precision and Recall. This measure is perfect as it gives us a measure of the balance between recall and precision.

5.6 AUC

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values. The Area Under the Curve (AUC) is the measure of the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve. In general, a value for AUC that is bigger than 0.5 indicates a situation where the classifier performs better than a random guessing.

6 Hold-out results

As a first approach, we utilize the hold-out procedure. We divide the dataset into training(67%) and test(33%) set, using a stratified sampling based on the attribute status. We then proceed with the learning and testing phase of all of the five algorithms.

Algorithm	Recall	Precision	F score	Accuracy	AUC
Logistic	0.024	0.571	0.046	0.897	0.525
Random Forest	0.003	0.250	0.006	0.895	0.507
MLP	0.599	0.107	0.182	0.440	0.509
NB	0.036	0.145	0.058	0.878	0.509
NB Tree	0.0	-	-	0.896	0.548

Table 1: Hold out results

The data in table 1 shows very high accuracy across nearly all the classification algorithms, and very low scores on all of the other measures. This is an evident problem of class unbalance inside the dataset: given that the bad clients represent 10.4% of the total records the algorithms tend to classify all of the clients as good. This is known as the "Zero Rule": in an unbalanced dataset the classifiers achieve very high accuracy simply by classifying all of the data as the most frequently occurring class, in our case "good". All the algorithms that achieve high accuracy show in fact low recall scores, demonstrating low abilities in identifying the bad costumers.
The only classifier that doesn't seem influenced by this problem is the MultiLayer Perceptron, that already achieves better results in Recall while losing some accuracy.

7 Cross-Validation

In order to get a more effective estimates of the classifier's performance measures we will use for the next analysis the K Fold cross validation to divide the data into k subsets. In this way the holdout method is repeated k times, such that each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. The error estimation is averaged over all k trials to get total effectiveness of our model. This method is used to tune models so as to optimize the bias-variance tradeoff. In our case, we decide to use the three folds cross validation (k=3).

8 Unbalanced Dataset

To resolve the problem of unbalanced dataset we proceed in two ways: by oversampling, or by adopting the cost matrix.
Oversampling consists in creating new artificial observation of the minority class, until the dataset becomes balanced. In particular, we used the technique SMOTE (Synthetic Minority Over-sampling Technique) [5], that works by creating synthetic rows by extrapolating between a real object of a given class and one of its nearest neighbors (of the same class). It then picks a point along the line between these two objects and determines the attributes of the new object based on this randomly chosen point.
The cost matrix, instead, is an array of numbers organized in columns and rows, each specifying a cost for each outcome in the confusion matrix, therefore assigning a cost to right and wrong predictions. This matrix allows for the usage of a cost-sensitive learning model. To build the cost matrix we used a common technique in the field: we pondered the costs according

to the ratio of the minority class to the total. We therefore gave a cost of 10.4 to the False Positives (the good costumers classified as bad costumers) and a cost of 100 to the False Negatives (the bad costumers classified as good costumers). This makes sense, given that approving a credit card to a bad client, will likely end up in more costs for the bank compared to the lost profit caused by not approving a credit cart to a good client. Lastly, we assigned a profit of 1 (cost of -1) to the correct identification of True Negatives. The resulting cost matrix is the following:

	Predicted good	Predicted bad
Good	0.0	10.4
Bad	100	-1

Table 2: Cost Matrix

8.1 Oversampling

The results of the five algorithms on an oversampled dataset created with the SMOTE techniques are summarized in table 3.

Algorithm	Recall	Precision	F score	Accuracy	AUC
Logistic	0.402	0.114	0.178	0.613	0.528
Random Forest	0.059	0.017	0.076	0.851	0.531
MLP	0.569	0.108	0.181	0.466	0.505
NB	0.625	0.103	0.177	0.396	0.498
NB Tree	0.024	0.112	0.039	0.879	0.512

Table 3: Oversampling results

The Naive Bayes algorithm is the one that achieves a better recall score. The algorithm is able to identify 632 of the 1012 bad clients (62.5%). This result, though, is achieved by classifying a lot of the costumers as bad costumers, as the precision measure points out. A low score on precision, in fact, means that few of the clients classified as "bad" by the algorithm are in fact "bad".

8.2 Cost Sensitive Learning

The results of the five algorithms applied instead with the inclusion of the cost matrix, are summarized in table 4.

As we can see from the table, the MLP achieves the highest Recall, that is equal to 0.843. However if we take into account the F-score, the algorithms which perform better are the Logistic and NB Tree.

Algorithm	Recall	Precision	F score	Accuracy	AUC
Logistic	0.606	0.113	0.190	0.464	0.526
Random Forest	0.626	0.111	0.188	0.439	0.522
MLP	0.843	0.106	0.189	0.245	0.5
NB	0.532	0.109	0.180	0.498	0.513
NB Tree	0.584	0.114	0.190	0.483	0.528

Table 4: Cost Sensitive Learning results

9 Feature Selection

Feature selection is the process of detecting relevant features and removing irrelevant, redundant, or noisy data. In other words, feauture selection involves evaluating the relationship between each input variable and the target variable and selecting those that show the strongest relationship. In literature different approaches are used: *brute force*, *wrapper* and *filter*.

For what regards the latter we can distinguish between univariate and multivariate filters. In the first case we can mention : t-test, ANOVA, mutual relationship. While, for the multivariate filters, measures like the Correlation Feature Selection, the Relief and the blanket are commonly used. In our cases the goal is to verify if the results change on the basis of the variables we select.

9.1 Oversampling - FS

In combination with the previously described SMOTE technique for oversampling the dataset, we implemented a multivariate filter to apply feature selection. To do this, we use the CfsSubsetEval method of the Weka AttributeSelectedClassifier node. The method selects the following variables: car and realty ownership, income, the "working" income type, the "Secondary Special" education type, the "Civil Marriage" and "married" family status, the "House Appartment" housing type and the "managers", "core staff", "sales staff", "laborer" and "Not Specified" occupation type.

With these variables, the algorithms gave the following results:

Algorithm	Recall	Precision	F score	Accuracy	AUC
Logistic	0. 378	0.11	0.170	0.616	0.514
Random Forest	0.140	0.114	0.126	0.797	0.504
MLP	0.319	0.105	0.158	0.646	0.498
NB	0.444	0.104	0.168	0.543	0.494
NB Tree	0.035	0.123	0.054	0.874	0.505

Table 5: Feature Selection - Oversampling results

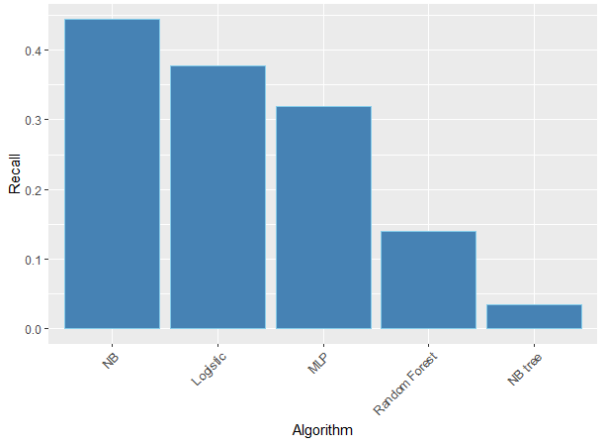

Figure 1: Recall results with Feature Selection-Oversampling

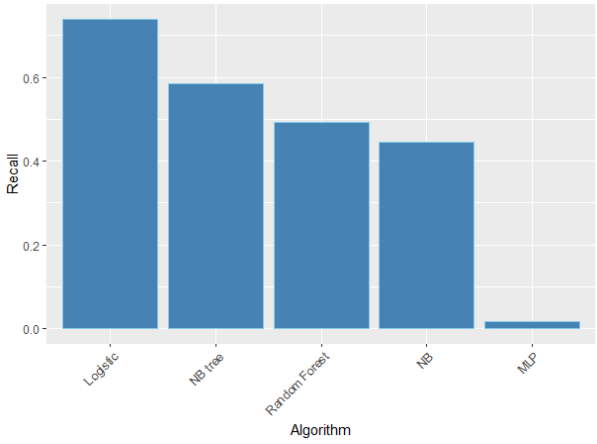

Figure 2: Recall results with Feature Selection-Cost Sensitive Learning

The Random Forest classifier performs better with this approach than in the hold out method. Whereas MLP and Naive Bayes achieves worse results, while the Logistic and NB Tree algorithms remains unchanged.

9.2 Cost Sensitive Learning - FS

For the Cost Sensitive Learning we use an univariate filter approach to perform the feature selection. We use the mutual information measure to identify those variables that are related with the status of the client. The variables selected are: the clients' number of children and of family members, the clients' income and age and if the client is single.
With these variables, the algorithms gave the following results:

Algorithm	Recall	Precision	F score	Accuracy	AUC
Logistic	0.738	0.11	0.192	0.353	0.523
Random Forest	0.493	0.114	0.185	0.548	0.524
MLP	0.016	0.186	0.029	0.89	0.541
NB	0.445	0.111	0.178	0.573	0.517
NB Tree	0.584	0.114	0.190	0.483	0.528

Table 6: Feature Selection - Cost Sensitive Learning results

While the MultiLayer Perceptron takes a big loss in its performances, the other algorithms' F-scores don't change much. Therefore, for these methods, a feature selection approach could improve time performances of the learners without having a significant impact on their results.

10 Conclusions and future developments

Firstly, the increase in all of the F-score measures of all of the five algorithms over the hold-out method have shown that both oversampling and the adoption of the cost-matrix are a good way to address the problem of the unbalanced dataset. Indeed, the high accuracy that the hold-out method achieved is likely not an indicator of the classifiers' good performances, but an effect of the measure's bias in favour of an all-negative classification.
Then, we also demonstrated how the Cost Sensitive Learning tend to achieve better results over the oversampling approach. Furthermore, this type of learning could be further improved using a real cost matrix, coming from the bank's data.
For what regards the different algorithms, the Logistic and Naive Bayes classifiers have shown good performances across all of the different approaches implemented. The Random Forest and the Naive Bayes Tree classifiers, instead, tend to give better results if paired with a Cost Sensitive learner. Lastly, the Multilayer Perceptron, achieved very good results using all of the available variables, especially paired with the Cost Sensitive Learner, while taking a big loss in performances when used together with a Feature Selector.
During our analysis, we were able to see how much difference a change in the threshold of classification in good and bad customer could make on the results of the algorithms. Classifying a client as a good or bad investment based on the percentage of month that saw a delay in

the payment of the load and the percentage of "bad" credit cards that he owns is a decision that has to be made based on past data. For this reason, we believe that the learning phase could highly benefit from a future research that could locate a threshold based on real world's banks' adversity on risks.

References

[1] Kelsey Coyle, Laura Kim, and Shaun O'Brien. "2021 Findings from the Diary of Consumer Payment Choice". In: *Federal Reserve Bank of San Francisco, Cash Product Office* (2021).

[2] Umabhanu Tanikella. "Credit Card Approval Verification Model". PhD thesis. California State University San Marcos, 2020.

[3] B Subashini and K Chitra. "Enhanced system for revealing fraudulence in credit card approval". In: *Int. J. Eng. Res. Technol* 2.8 (2013), pp. 936–949.

[4] Hussein A Abdou and John Pointon. "Credit scoring, statistical techniques and evaluation criteria: a review of the literature". In: *Intelligent systems in accounting, finance and management* 18.2-3 (2011), pp. 59–88.

[5] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.