Model selection for credit risk

Matteo Bandiera, Samuele Fonio, Luca Macis

December 2021

Abstract

The aim of this paper is the analysis of which are the characteristics of an individual that most accurately predict the capability of being a solvable creditor. We used three different classification techniques, namely: logistic regression, k-nearest neighbors and support vector machines. These techniques were chosen based on the nature of the problem (for logistic and knn) and on the shape of the data (SVM). Therefore, since these methods are quite different, this article can also be seen as a comparison between strong classifier, capable of just attribute values 0 and 1 and soft classifier, which gives us also informations on probabilities on the binary classification.

Data explanation and EDA

This dataset was taken from a Taiwanese bank facing the credit card crisis of 2006. The dataset has 30000 observations and 24 features. We had to do some cleansing resulting in 29601 observations and 24 features. Here are the variables:
- *LIMIT_BAL* (num): amount of the given credit (NT dollar);
- *Gender* (cat): costumer's sex (1=male;2=female);
- *Education* (cat): costumer's education (1=graduate school;2=University;3=High school;4=Others);
- *Marriage* (cat): customer's marital status (1=married;2=single;3=others);
- *Age* (num): customer's age;
- *payment_sept,...,payment5_april* (num): number of arrears of monthly payments;
- *bill_statement_sept,...,bill_statement_april* (num): amount of bill statement of that month;
- *prev_payment_sept,...,prev_payment_april* (num): amount of previous payment of that month;
- *credit_default* (binary): result of credit default (Yes=1,No=0);

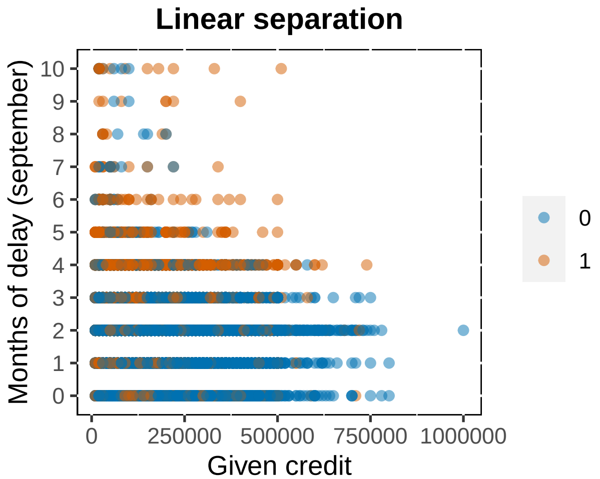


Figure 1

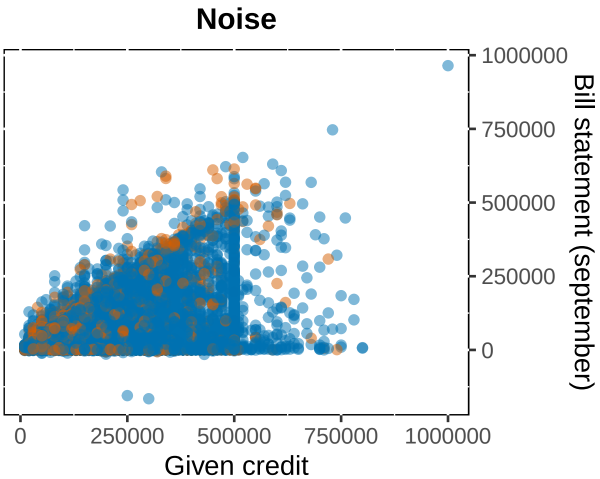


Figure 2

As we can see in Figure 1 the more the months of delay the greater the number of defaults. In Figure 2 we show how continuous variable cannot achieve a satisfying separation in the data. This kind of "noise" is the first problem we had to deal with addressing this task, since KNN and SVM may suffer from this.

At a glance we can notice how solvent customer are the majority, in fact we have 22996 solvent clients and 6605 insolvent ones, make it a quite unbalanced dataset to deal with. For all the models we divided the dataset in training and test sets (with proportion 9 to 1). We will use cross-validation only when needed (Lasso and Ridge, otherwise we will just use different measures of performances on test sets. As measures we used (using TP as true positives, FP as false positives, and N for negatives). $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, $balanced\ accuracy = (\frac{TP}{pos} + \frac{TN}{neg})/2$ (average recall), $F1 - score = \frac{TP}{TP+\frac{1}{2}(FP+FN)}$, $sensitivity = \frac{TP}{pos}$,$specificity = \frac{TN}{neg}$.

Logistic Regression

As regard the nature of the problem, this approach may represent the best choice because, for each unclassified customer, we will have a probability to be insolvent. Having access to probabilities it allows us to have bigger leeway in managing the model. We start with fitting the model with all the variables and classifying the insolvent customers (credit_default=1) taking as a threshold of 0.5. Then we use anova test to get a reduced version of the model and one with interactions. These are the results we have obtained in performance for the best model:

Balanced Accuracy	Accuracy	Sensitivity	Specificity	F1
0.6080651	0.8131125	0.9782514	0.2378788	0.8905167

As we can see the accuracy is quite good, but we know that in unbalanced conditions we need to use different measures that take into account both the false positives (false solvent customers) and false negatives (false insolvent customers). In this model we have a very low specificity, i.e. there are a lot of false positives. On the other hand sensitivity is high, i.e. there are almost no false negatives. This unbalanced situation between sensitivity and specificity is reflected in the balanced accuracy (it can obviously improve) and in the F1 score (high enough since we have a low value of false positives). Now, since the bank natural goal is to minimize errors in classification, in particular the error of classifying as solvable a credit which will not be repaid, we have to change measure of performance to address the real world concerns. To tackle this task we can change the probability threshold to decide if a customer is insolvent or not. To find the best threshold for the probability we used the probability cutoff as a tuning parameter. As a graphical reference we plotted the evolution of threshold on specificity and sensitivity.

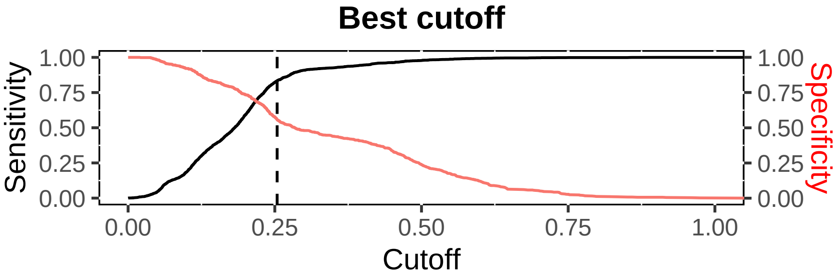


Figure 3

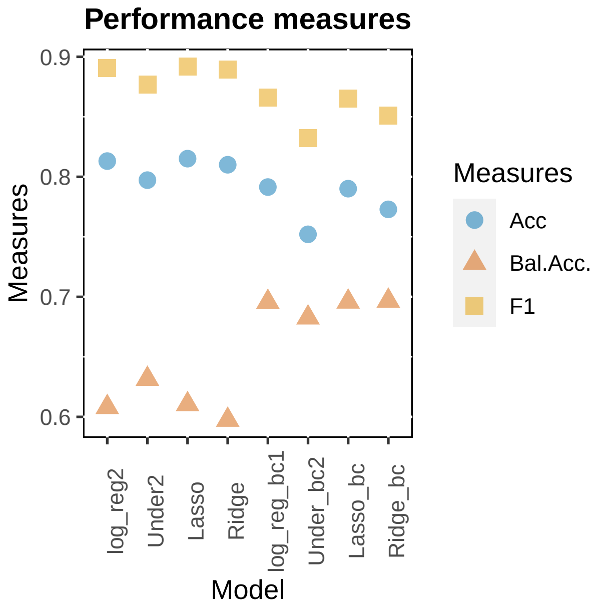
Changing the probability according to this method we resulted in the following performances:

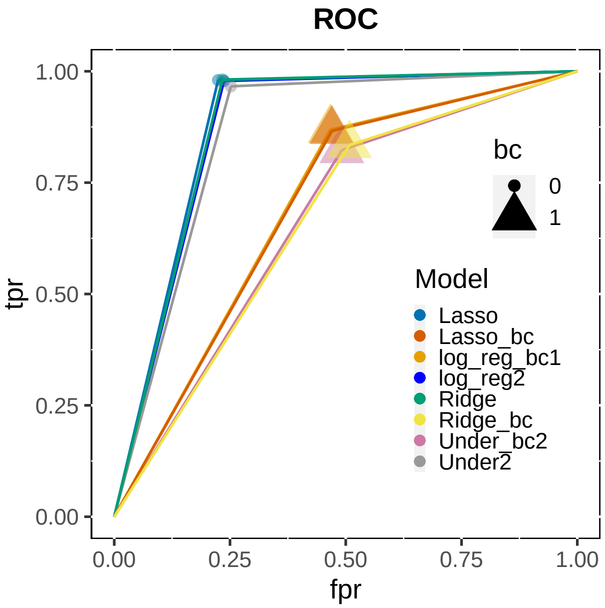
Balanced Accuracy	Accuracy	Sensitivity	Specificity	F1
0.6956829	0.7914836	0.8686385	0.5227273	0.8661895

As we can see specificity and balanced accuracy improved, but the cost is that now we have more false positives, resulting in a worse accuracy and sensitivity. The disclaimer can be the F1-score, worse than before: to balance the FN (from 503 to 291) at the denominator we added a lot of FP (from 50 to 384).

Having more knowledge on economical theory could eventually lead us to tailor the model weighting the probability cutoff according to the effective on specific parameters. Unfortunately this is unfeasible.

However, we kept searching for improvements. Since we noticed that we had an overfitting problem in the logistic models (pchisq value for deviance were 1), we thought that solving it could result in better performances. To achieve this reduction in overfitting we used three techniques of regularization: undersampling, Lasso and Ridge regressions. Lasso and Ridge were performed on the whole dataset. Then with undersampling we reduced the number of solvent customers to reach a good pchisq value and to balance them with respect to the insolvent customers. For each model we applied then the best cutoff. We got the following results:
(In general, “1” in the name of the model means “logistic model with all variables”; “2” means “simplified logistic model”; “bc” means “best cutoff”)





As we can see solving overfitting leads us to a better performance, in particular the best is performed by Lasso. For what concerns the models with best cutoff the best one is logistic regression with all the parameters.
If we look at the ROC plot we can see that in general the models with default thresholds perform better than the others. The latter ones could be of some interest for the bank. We will discuss this in the conclusions.

KNN

Since KNN is a model based on distance of the observations, we have to treat our dataset to use the model in a proper way. First of all, we will not include the features “sex”, “marriage” and “education” because they are unordered factors with no proper meaning for the KNN model. We also decided to normalize the remaining features to avoid one of them being more influential than the others without a valid reason.

Once we normalized the dataset, we can divide it in training set and test set with the same proportion we used for the Logistic.

Now we can start with our KNN prediction. We do not only want to build the model based on best “K”, we also want to find the best distance kernel to tune our model on. To achieve this purpose, we will do a Grid Search. The Grid Search will evaluate each “K” with every possible distance kernel (rectangular, triangular, Epanechnikov, biweight, triweight, cos, inversion, Gaussian). To evaluate each model, we will use the following measures: balanced accuracy, accuracy, F1 score, sensitivity and specificity. Grid Search will select the best model based on the balanced accuracy performances.

We can see the parameters found by our Grid Search are k=43 and kernel weighted distance is inversion=$\frac{1}{|d|}$ (with d=distance between points). These are the performances on our test set:

Since Knn is a soft classifier we tuned the probability threshold to improve even more the balanced accuracy as we did in the Logistic. These are the results:

	bal. acc.	accuracy	sensitivity	specificity	F1 Score
Performances	0.6706076	0.8310811	0.9510099	0.3902054	0.8984978

	bal. acc.	accuracy	sensitivity	specificity	F1 Score
Performances	0.70669	0.7675676	0.813064	0.600316	0.8461538

SVM

Our last model in analysis is the Support Vector Machine. As for the KNN, we want to find the best hyperparameters, based on the best Balanced Accuracy, with a Grid Search. Since SVM complexity is O(n_features * n^2 objects), we need to reduce the dataset and try SVM on a much smaller portion. The intractability problem of SVM training and how to best reduce the training test impacting the least possible the pattern of the Support Vector is another topic that should to be treated separately. For this analysis we choose the common practice of randomly picking 10% of the original dataset (2961 observations, 2665 for training and 296 for testing). The hyperparameters we would like to tune are: degree, cost and gamma, considered that the kernel chosen is polynomial. The degree of the polynomial could be between 1 and 3, the cost between 0 and 20 with step 0.1 and gamma between 0 and 2 also with step 0.1. With bigger values of gamma the hyperplane and its support vectors are allowed to change their shapes according to the observations near the margin.

The Grid Search result is: polyonomial Kernel of degree 1; cost: 10.5; gamma: 1.2.

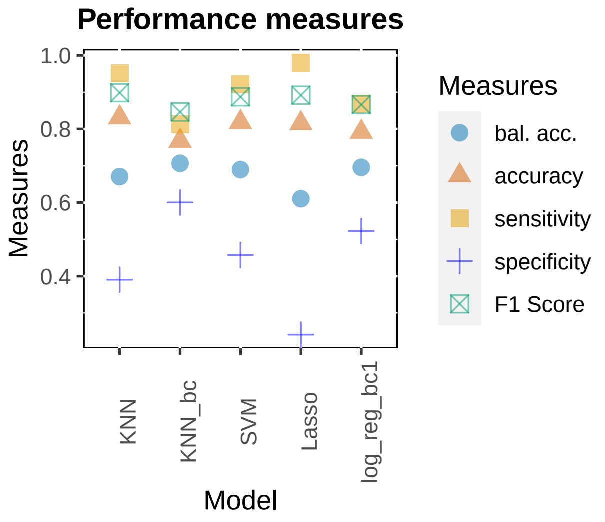
To check if our dataset reduction compromised the efficiency of the model we decided to try it on the original bigger test set (2959 observations), where no data were omitted. These are the resulting performances:

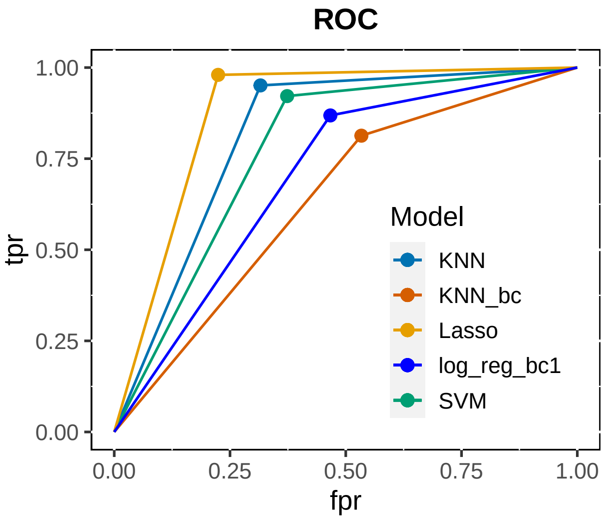
	bal. acc.	accuracy	sensitivity	specificity	F1 Score
Performances	0.6896404	0.8181818	0.9217051	0.4575758	0.8873534

From the result on the original test set with no data omission we can see that the data reduction did not compromise the reliability of the SVM Model.

Final comparison and conclusions

In this analysis we developed three different classifiers to predict solvent and insolvent customers. In the following plots we want to compare the different performances among these models.





Altough there is not an unambiguous way to choose the best one, we would like to give our suggestion: If the bank is interested in identifying the largest number of possible insolvent customers, then the KNN classifier with the best cutoff probability is the one to go, being aware that the price to pay is losing some solvent customer. Indeed, if the goal is to misclassify the least possible, then KNN classifier is the best choice. Finally, the bank might choose the SVM classifier if it wants good results in classification taking into account that there is no room for improvement. From the ROC plot we can see that also Lasso has good performances and, as we said, there is room for improvement. In conclusion, we remember that the Logistic and KNN are soft classifiers, so we are able to change the probability cutoff to improve performance handly.