Model selection for credit risk

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 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 soft classifier, capable of just attribute values 0 and 1 and strong classifier, which gives us also informations on probabilities on the binary classification .

Data explanation and EDA

This dataset (http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients) was taken from a bank in Taiwan which wanted to monitor the possibility of credit default since, for facing a huge economical crisis, it started to permit a lot of credits. The dataset was quite big: 30000 observations for 24 features. We had to remove some of them since the description we found was not coherent with the data (e.g. more classes for categorical features had no explanation) resulting in 29601 observations and 24 features (we removed the ID number because useless). Here are the variables:
- *LIMIT_BAL* (numeric): Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit ;
- *Gender* (categorical): costumer's sex (1=male;2=female) ;
- *Education* (categorical): costumer's education (1=graduate school;2=University;3=High school;4=Others);
- *Marriage* (categorical): customer's marital status (1=married;2=single;3=others);
- *Age* (numeric): customer's age ;
- *payment_sept,. . . ,payment5_april* (numerical): History of past payments. The measurement scale for the repayment status is the delay in months with respect to the expected payment of that month;
- *bill_statement_sept,. . . ,bill_statement_april* (numerical):Amount of bill statement of that month;
- *prev_payment_sept,. . . ,prev_payment_april* (numerical):Amount of previous payment of that month;
- *credit_default* (binary): Result of credit default (Yes=1,No=0);

Note: All numerical values are in NT dollar.

As exploratory data analysis we decided to put these graphs in order to describe the principal features of the dataset: linear separation for some variables and "noise" for other.

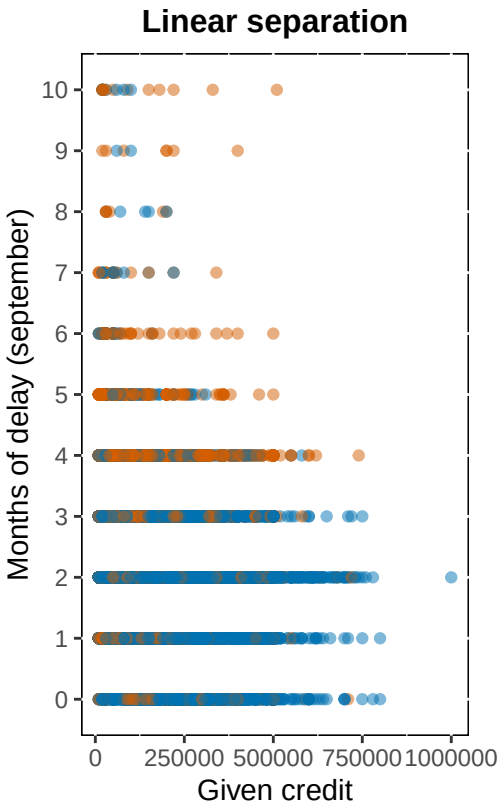


Figure 1

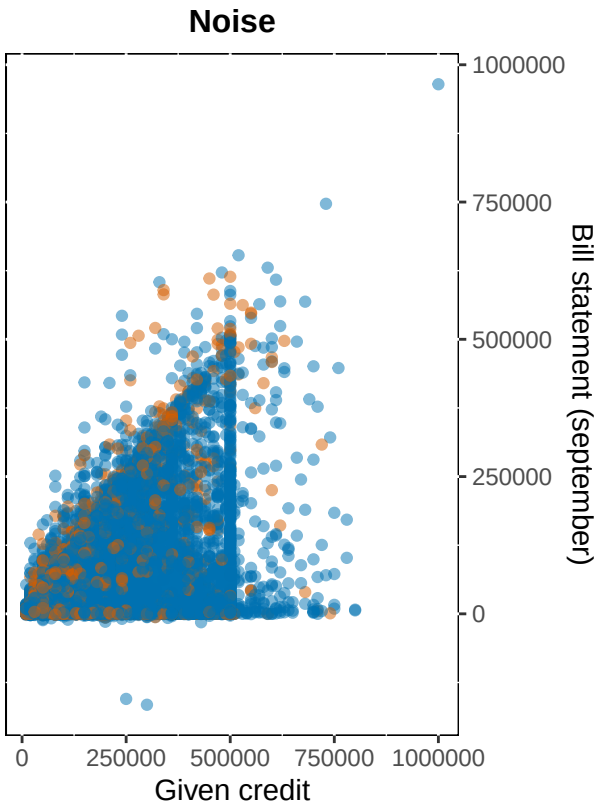


Figure 2

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 can not 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 data set 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:

```
##              Balanced Accuracy  Accuracy Sensitivity Specificity        F1
## performances1         0.6110954 0.8144643   0.9782514   0.2439394 0.8912225
```

```
## performances2          0.6080651 0.8131125   0.9782514   0.2378788 0.8905167
## performances3          0.6041214 0.8002704   0.9582427   0.2500000 0.8817290
```

	x
Balanced Accuracy	0.6080651
Accuracy	0.8131125
Sensitivity	0.9782514
Specificity	0.2378788
F1	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 takes 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 has as a natural goal the one of minimizing errors in classification, and in particular in classifying as solveble 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.

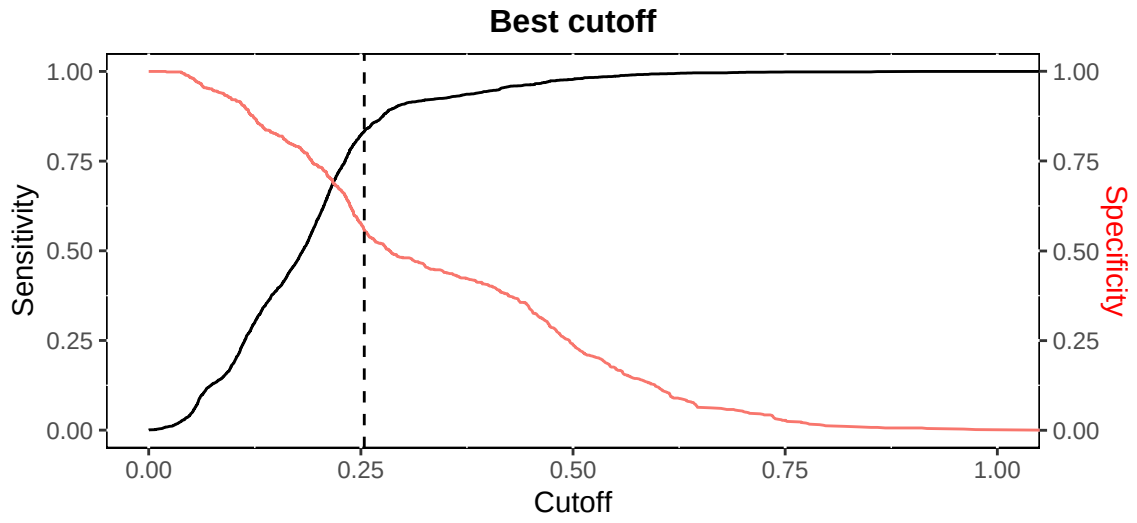


Figure 3

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

```
##                    Balanced Accuracy  Accuracy Sensitivity Specificity        F1
## performances_bc1           0.6956829 0.7914836   0.8686385   0.5227273 0.8661895
## performances_bc2           0.6960309 0.7718824   0.8329709   0.5590909 0.8501665
## performances_bc3           0.6511527 0.7600541   0.8477599   0.4545455 0.8459201
```

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,

	x
Balanced Accuracy	0.6956829
Accuracy	0.7914836
Sensitivity	0.8686385
Specificity	0.5227273
F1	0.8661895

worse than before: to balance the FN (from 499 to 208) at the denominator we added a lot of FP (from 50 to 724).

Having more knowledge on economical theory could eventually led us to taylor 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. First of all, 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, then we used Lasso and Ridge.

For each model we applied then the best cutoff. We got the following results:

```
df_test[c(1,3,5,7),]#----> scegliamo il modello log_reg_under2
```

```
##                 Accuracy Balanced.Accuracy        F1
## log_reg2       0.8131125         0.6080651 0.8905167
## log_reg_under2 0.7972456         0.6315964 0.8769159
## log_reg_lasso  0.8151403         0.6104502 0.8917475
## log_reg_ridge  0.8100710         0.5974663 0.8892393
```

```
df_test[c(2,4,6,8),]#----> scegliamo il modello log_reg_bc1
```

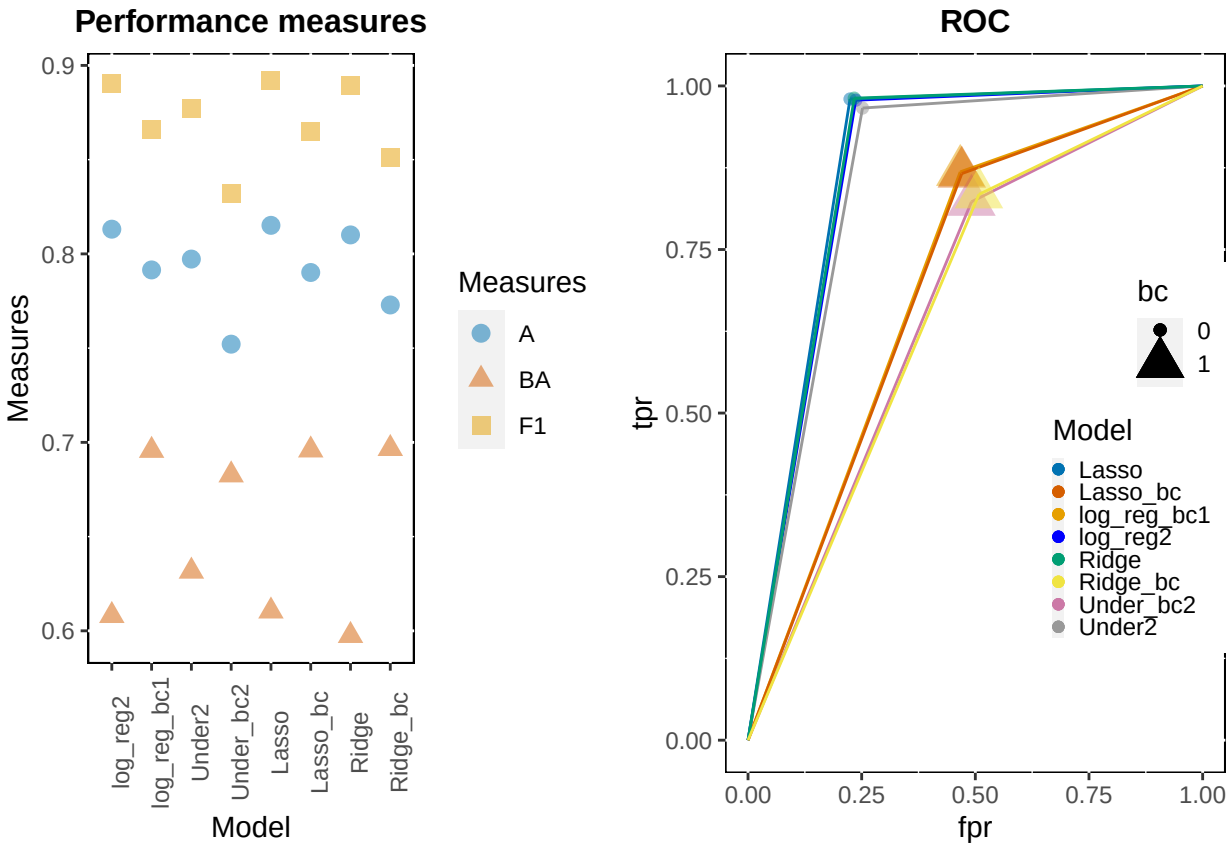
```
##                    Accuracy Balanced.Accuracy        F1
## log_reg_bc1       0.7914836         0.6956829 0.8661895
## log_reg_under_bc2 0.7521041         0.6826758 0.8322981
## log_reg_lasso_bc  0.7901318         0.6958931 0.8650880
## log_reg_ridge_bc  0.7728962         0.6966833 0.8509317
```

```
## [1] 8
```

```
## [1] 0.9782514 0.8686385 0.9662231 0.8229273 0.9799913 0.8660287 0.9812962
## [8] 0.8342758
```

```
## [1] 0.2369668 0.4667697 0.2519084 0.4914773 0.2243902 0.4702290 0.2336957
## [8] 0.5080000
```

```
## [1] "log_reg2"    "log_reg_bc1" "Under2"      "Under_bc2"   "Lasso"
## [6] "Lasso_bc"    "Ridge"       "Ridge_bc"
```



As we can see solving the overfitting of the logistic regression led us to better performances.So we can see from this graph that the best model is the under sampled logistic reduced model with interactions.

?? Furthermore, in the ROC plot the ones with the optimized cutoff perform worse, which can be another parameter to take into account.

To conclude, the best model that we are going to compare with the others is the Lasso, since its test performances are a little bit better with respect to the other models.??

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 that one of them will be more influent 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. 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 here:

	Best K	Best distance
Results	43	inv

And the performances found in the training set with those parameters:

These are the performances on our test set:

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

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

Results with best cutoff:

SVM MODEL

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 to impact the less possible the pattern of the Support Vector is all another topic that need to be treated separately in another occasion. For this analysis we chose the common practice to randomly pick 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.

These are the best values for the hyperparameters found by the Grid Search:

	Kernel	Degree	Cost	Gamma
Results	polynomial	1	10.5	1.2

And the performances in the training set with those parameters:

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 was omitted. This is the resulting confusion matrix:

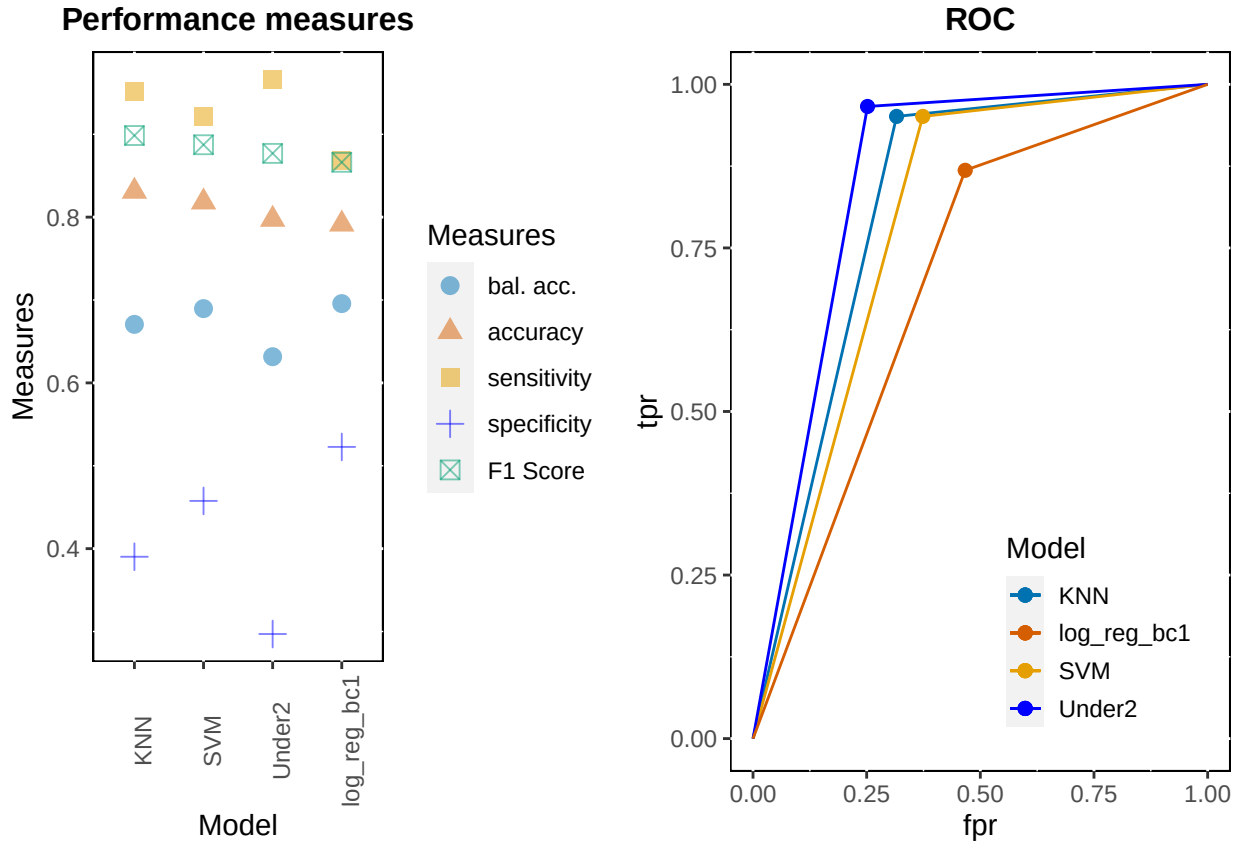
And these are the resulting performances:

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.

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

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 between these models.



There is not an unambiguous way to prefer one above the others.

Based on these plots, it is up to the bank to choose the classifier to use. If it is in the bank's interest to identify the largest number of possible insolvent customers, The Lasso classifier with the modified cutoff probability is the best one, being aware that a large number of potential solvent customers will be lost. Indeed, if the bank want the best rate of predicted, regardless of whether they are solvent or insolvent, the bank would choose the KNN classifier, being aware that is the worst in predicting the possible insolvent customers. Finally, the bank could choose the SVM classifier if it wants a good balance between overall prediction and solvent and insolvent rate prediction.

In conclusion, we remember that the Logistic is a soft classifier, so we are able to change the probability cutoff to improve performance in reference to the need of the moment.

(. . . Also the Knn could provide provide probabilities associated to each new unseen observations. . .)

Indeed the SVM is considered a strong classifier, it cannot provide probability of belonging to a class. The SVM model presented in this analysis is the best possible and it cannot change its classification

based on probability of pertinence.