Prediction of potential credit default account

Ismail Oussaid, Youssef Ammi

Abstract

The objective of this project is to use personal & financial data of customers such as their age, gender or educational level to scrutinize their relation to the default of some payments in a specific time and predict individual customers' behavior concerning loan repayment using diverse classification algorithms. The comparison of these methods using our imbalanced data will give a very proper model to solve decision problems relating to default payment which will allow banks to make important decisions regarding credit extension. And thus minimizing financial loss for these very banks.

***Keyword**: Classification, Credit Scoring, Prediction, Imbalanced data*

Introduction

Our objective is to predict the default payment variable/label which takes two values: 0 or 1. Let us get an idea about the ratio of "ones" in our dataset, it turns out to be : 0.22 and plotting the distribution of labels (Fig.1) shows how imbalanced is our dataset.

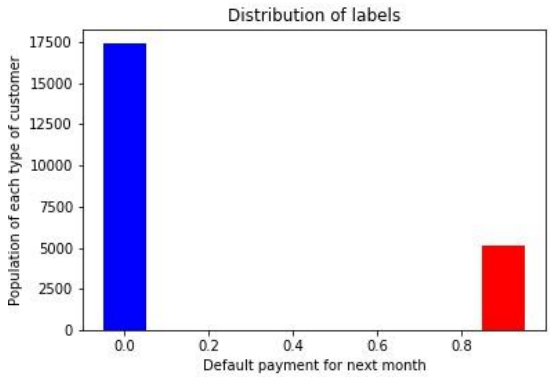

Fig.1 *Distribution of default payment labels*

The challenge appears when machine learning algorithms try to identify the rare case (in our case, one). Due to the disparity of classes in the variables, the algorithm will tend more to categorize into the bigger class : "zero", the majority class. At the same time, it gives the false sense of an accurate model, because we can have a somewhat accurate model simply by predicting the response variable will always belong to the biggest category. Indeed, if we build a predictor giving only 0, we do have an accuracy of 0.78.

The inability to predict the minority class and the false accuracy detracts from the model we build carelessly with regard to the unbalanced nature of our dataset.

First, we look if there are any missing information in the dataset with df.info(), and we see that none of the data is missing.
After that, we check whether there are anomalies by looking closer at each category in the dataset. We realize, calling the describe function, that some values for some categories are unlabeled or undocumented in our dataset description.

Calling the describe function for the categorical attributes shows that some labels of these categories are not defined, not documented, that is an anomaly in our dataset that needs to be dealt with before working on our prediction algorithms

MARRIAGE takes a value which is undefined in our dataset description : 0

EDUCATION takes some undefined values in our dataset : 0,5 and 6.

PAY_0,..,PAY_6 take an undefined value : -2

The 0 value **MARRIAGE** can take will be seen as 'Other' since we have no idea about it. Therefore any 0 value will be transformed into '3', the actual documented 'other' category.

The 0,5,6 values **EDUCATION** can take will be seen as 'Other' since we have no idea about its signification. Therefore any of those values will be transformed into '4', the actual documented 'other' category.

The description of our dataset indicates that the **PAY_n** variables refers to the number of months of delay and that -1 is for a duly paid case. Looking at the other projects working on this dataset, a lot of them agree that an adjustment needs to be made so that 0 refers to "paid duly", and any value -2 is set to 0, because looking at the histogram that makes sense when it comes to sparsity and distribution of classes.

We check to see how our dataset is divided and if there are sparse classes in the categorical attributes : **MARRIAGE**, **EDUCATION**, which can cause overfitting.

Using the command df.value_counts().plot, we realize that some classes are indeed sparse.

Calling the describe function, we detect the following anomalies :

BILL_AMT1,...,BILL_AMT6 all take negative values which doesn't make any sense since they represent the monthly amount of bills, maybe they can refer to the credit of the person.

We will also rename the columns '**PAY_0**' as '**PAY_1**', because that makes the attributes names much more intuitive.

Feature analysis

Intuitively, we believe some attributes will be very correlated with our response variables : mainly education, marriage, and sex. We shall verify this and use it for some feature engineering, hence combining them for proposing some new features.

SEX :

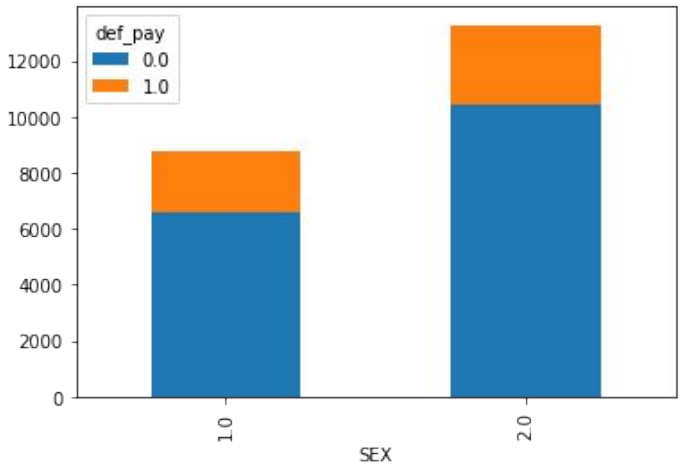

Fig. Distribution of genders

We have seen that 22% of the customers default, this, coupled with our graph above, implies that there are significantly more women than men and that men are therefore most likely going to default the next month.

EDUCATION:

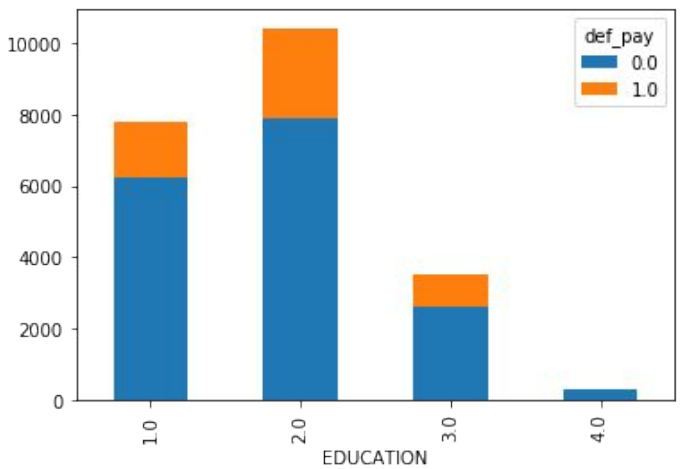

Fig. Distribution of education categories

The plot proposes that the higher the education, the less probable it is to default during the next month. As per the category labeled "*Other*", the documentation implies it would be lower than high school, so this is not necessarily correct. But that's fine since it's the least weighted class.

MARRIAGE:

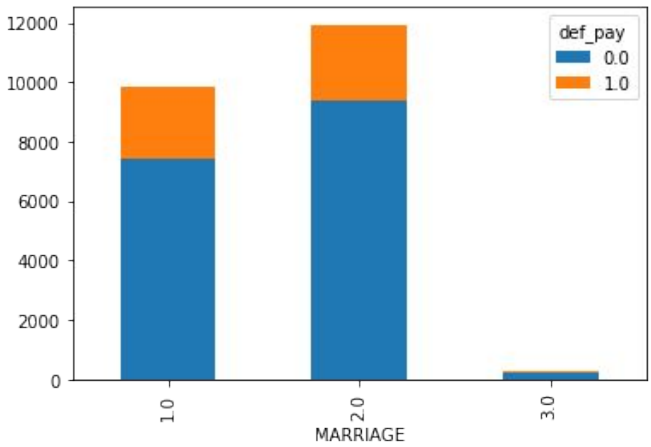

Fig. Distribution of marriage categories

The plot proposes that being married makes you more likely to default, just like in "**EDUCATION**' the category "*Other*" is again numerically negligible so we don't care to interpret its correlation to the response variables.

Feature engineering :

MARRSEX :

We have just seen that men are most likely to default, and marriage makes you more likely to default as well, so why not create a new feature which will categorize "married men" : we do that by considering a variable called **MARRSEX** which combines **SEX** and **MARRIAGE** by taking their product.

C_n :

Next we will create a new feature which gives an idea about how far is the bill from the limit allowed, as intuitively this gives an idea about the ability to pay back or not.

AGESEX :

Intuitively, the probability of defaulting is low in your 30's (because in general people don't have many kids in that age), and then goes up as age increases, this motivates us to create a new category combining age and sex, or more precisely combining sex and the interval of age each client belongs to. The new feature is called **SE_AG**, and gives a very good feature in terms of correlation to the response variables, the clients are somewhat evenly distributed with respect to this feature, which makes our set more and more balanced.

EXP :

Next, we create a feature concerning the expenses of each client relatively to his limit, as this gives an idea about the risks he is taking, a person spending close to the limit allowed is more likely to fail in paying back.

Model Tuning

Tuning :

As for our algorithms we shall use : Ensemble methods , which are learning algorithms that construct a set of classifiers and then classify by taking a (weighted) vote of the predictions. For each sampling strategy, we shall tune the following algorithms :"Random Forest", "Adaboost", "Gradient Boost", "Logistic Regression" & ""Naives Bayes" and give the accuracy of our tuned model on the whole training set. We use the Gridsearch method to tune the hyperparameters of our predictive algorithms. The three first ones were the best so we decided to focus on them.

The GridSearch uses every possible combination of the hyperparameters and selects the one giving the higher accuracy.

Sampling :

The fact that our dataset is unbalanced encourages us to use some sampling techniques to tune the parameters of our algorithms.

We can **upsample** the minority class : using sampling with replacement and we can **undersample** the majority class : removing some observations of the majority class

The upsample might increase the variance since it replicates the minority class. It usually is better than downsampling, indeed the downsample can deprive the algorithm from important information and increase bias, but training on a downsampled training set can be much faster in training,fitting and predicting.

In a bias/variance trade-off approach, we can use the **SMOTE** approach for sampling : synthetic sampling.

For each sampling strategy, we shall tune the following algorithms :"Random Forest" (RF), "Adaboost" (ADA), "Gradient Boost" (XG) , and give the accuracy of our tuned model on the whole training set.

The results are as follows :

TRAINING

Accuracy	*RF*	*ADA*	*XG*
NORMAL	0.81	0.727	0.795
UPSAMPLE	0.949	0.725	0.858
DOWNSAMPLE	0.718	0.722	0.695
SMOTE	0.864	0.856	0.847

Fig. Training comparison for the three algorithms

TESTING

After tuning and training our algorithms, it is time to test them on the testing dataset. The results are as follows :

Accuracy	*RF*	*ADA*	*XG*
NORMAL	0.814	0.731	0.8
UPSAMPLE	0.808	0.685	0.721
DOWNSAMPLE	0.77	0.779	0.734
SMOTE	0.805	0.815	0.806

Fig. Test comparison for the three algorithms

For the RandomForest classification algorithm, as expected the upsampling approach fails to generalize its excellent performance to new data, whereas the downsampling approach improves the generalization error. As per the best performance, the normal sampling and SMOTE sampling strategies have the same accuracy but the SMOTE sampling takes less time, so it turns out to be the best one. The trade-off between bias and variance is once again the best approach.

As for the ADABOOST and Extreme Gradient Boosting approaches, the SMOTE sampling is the most performant when it comes to both training and generalizing error.

The expectation about downsampling training the fastest algorithms is met as well.

We can see that the algorithms are more accurate for the upsampling as expected, because as we said : the upsampling tends to overfit. But we don't expect it to be accurate on the testing dataset. The downsampling is less variant but has more bias. This is why we believe the SMOTE sampling approach is the best one as it represents a bias/variance trade-off.

The SMOT has a lower precision on the training set but we expect it to fit better on the testing dataset.

Time (s)	*RF*	*ADA*	*XG*
NORMAL	43.1	0.999	13.5
UPSAMPLE	15.7	45.1	34.8
DOWNSAMPLE	10.6	9.6	7.3
SMOTE	10.1	55.7	29

Fig. Time comparison for the three algorithms

Here, we can see that the three algorithm do not take much time to execute on a test set, which we found surprising.

The face it that ADA SMOTE is the slowest one but has the best accuracy so one can say that the cost of this accuracy is to wait longer, but considering that the execution lasts less than one minute, it is reasonable to take this couple sampling-model to work with for such cases.

Before deciding to work with those classifiers, we have trained Logistic Regression, SVM with different kernel, etc. even Neural Network (except Naive Bayes) and it happens that those models were not capable, considering out feature engineering, to be better than the previous better model.

Positioning our work with respect to what have been done :

At first sight, it seems that the accuracy of our algorithms is not impressive, but looking at what has been done all over the internet and in our references, such as : (2015). Benchmarking state-of-the-art classification algorithms for credit scoring. Indeed this is because of the unbalanced dataset, we have tried to attack this problem by adopting some specific sampling strategies when tuning our models : Downsampling, Upsampling and SMOTE, as this has not been done in the papers we referred to. But the unbalance of the dataset is still too much to overcome by simply sampling in a smart way.

In “The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients.” , in pages 2476,2477 , we can see that only the artificial neural networks and naive bayesian approaches outperform what we did.

It turns out that the work that have been done in this field does indeed show the supremacy of Artificial Neural Networks for this type of predictions, as it has been shown in “Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation” and in “Desai, V. S., Crook, J. N., & Overstreet, G. A. A. (1996). Comparison of neural networks and linear scoring models in the credit union environment”.

Further analysis

At the end, we have plotted the importance columns for each model’s best performance and we have noticed that PAY_1, AgeBin, EDUCATION & SEX are the most important features. Indeed, they are key factors since we have felt this at the very beginning because if one has a great degree, one should have a bigger salary which helps him repay the credit. Plus, as we know, salaries happen not to be the same for men and women in general so it clearly makes difference in credit repayment. Furthermore, the age is a very important actor because young people (in average) earn less than older people. And finally, PAY_1 was the only surprise for us, it is decisive and it proves the repayment status of the first month is crucial.

Furthermore, we could have made batter results if we had digged more to get more uncorrelated features. Indeed, as you can see in the code we have provided (correlation matrix), some features are very correlated so

this has a negative impact on our results, for sure.

Conclusion

Finally, after a very methodological research, the creation of intuitive features and the removal of some useless features, we can prove that XGBOOST with a SMOTE sampling is very efficient method. In fact, only Neural Networks happens to be better and more efficient. But this is true with one specific features engineering that is mentioned in our references. Furthermore, one shall not neglect how close Random Forest with normal sampling (cross validation) is from the better algorithm. So both solutions are tremendous and should be considered to solve such problems.

References

- Desai, V. S., Crook, J. N., & Overstreet, G. A. A. (1996). Comparison of neural networks and linear scoring models in the credit union environment. European Journal of Operational Research, pages 24–37.
- Cheng Yeh, Che-hui Lien (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, Volume 36, Issue 2 Part 1, pages 2473-2480.
- Stefan Lessmann, Bart Baesens, Hsin-Vonn Seow, Lyn C.Thomas (2015). Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society, Volume 247, Issue 1, pages 124-136
- Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A review. Journal of the Royal Statistical Society, Series A – Statistics in Society, pages 523–541.
- Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. Management Science, pages 312–329.