Data Mining Project 2017

Team: Lanzichenecchi

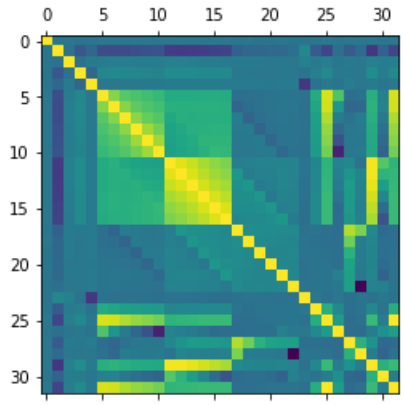
Giovanni Bucci Riccardo De Togni Nicola Ghio Nico Montali

1 Introduction

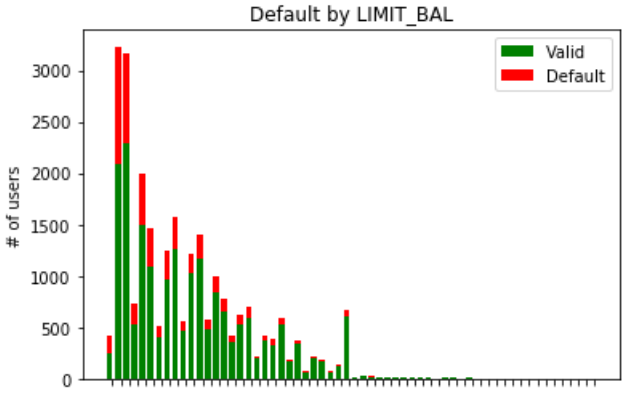
Data mining is the process of discovering patterns in large data sets. Including all operation from data gathering to final results, it is particularly useful when it comes to predict values or behaviours. In this document it is presented the full process applied to a real case for the Data Mining and Text Mining course at Politecnico di Milano. Given a data set of credit card subscribers, provided by Bip, with information about users and a partial payment history, our task was to find a way to predict if users are going to be credible or not basing on this data, and thus allowing proactive risk management actions.

2 Data Exploration

We start by describing our insights of the data set.
First, we noticed that the target class labels are highly unbalanced: 77,9% for class labeled 0 and the remaining 22,1% for class labeled 1, so this is something we must consider during the training of the models.
We also noticed that around 500 samples have all their BILL_AMT_X and PAY_AMT_X attributes set to zero, so we considered them as outliers. Since these outliers seems to have strange behaviour in relation with target we decided to drop them. We performed many analysis on the single features. We report here only the most interesting. The correlation matrix show some interesting facts: the six pay_* columns (from 5 to 10) are correlated somehow with the target(25). Not surprisingly the BILL_AMT columns are correlated one each other.



In the figure below we can see as low limit balance customers tended to default more often than others. The percentage of default customers below 500 euros limit balance is 40%, which doubles the average default percentage.



3 Data Preparation

This process is composed by the following steps:

1. Adjust training and test set, using a common format for strings, numbers, dates, etc. in order to allow the

correct reading of the data, and avoid errors in data elaboration

2. Thanks to the amount of data, it is possible to remove all the outliers mentioned above, without a consistent loss of information

3. Fill in the missing values: we set -1 to the missing values of AGE since it is a numerical attribute, while for SEX, EDUCATION and MARRIAGE we set them to 'NaN'. We preferred this approach since we cannot be sure that they are missing completely at random (MCAR)

4. Given the meaning of the data, we decided to set all the negative PAY_X attribute to -1 so that it could be still meaningful and different from zero or other values, while not deviating too much from the concept represented by this attribute inside the data set

5. Normalize BILL_AMT_X and PAY_AMT_X in each row with respect to the corresponding LIMIT_BAL

6. Delete CUST_COD attribute, useless when it comes to find patterns in data

7. Build new features: for each attribute PAY_DELAY_X, PAY_AMT_X and BILL_AMT_X, we build two new features, that are their row-wise weighted mean and average derivative, called ATTRIBUTE_D and ATTRIBUTE_P where "attribute" stands each time for the name of the selected attribute (example: PAY_AMT_D). This is done in order to express a sort of trend in payments and bill amounts

For the first tests, given the sufficient number of records, the data set has then been split through holdout technique to prepare test set and training set.

4 Modelling and Evaluation

After the splitting of the data set using an 2 thirds - 1 third holdout to separate train test and test set, we calculated a first baseline with a dummy classifier that predicts only the majority class; this resulted in an F1 score of 0.36, giving us a reference point for comparing the performance of our model.
Since this is a classification problem we focused our attention on the main models and for each one of them we performed parameter tuning and feature selection first using holdout, then with cross validation on the training set, giving importance to the weights of the two classes so that the models could see them both with the same importance. This different distribution of the classes could be a problem for the models, so it had to be mitigated, and it was managed differently for every model: using techniques like under or over sampling, or tuning dedicated parameters already present in the classifiers. The best features are selected independently for each model based on the correlation of the single feature with the target. The best subset is chosen with repeated tests.
In particular, we considered the following models: Logistic Regression, Random Forest, SVM, XGBoost, KNearestNeighbors, Multi-Layer Perceptron and AdaBoost. Among these, the statistically best one resulted to be the Random Forest Classifier with an F1 score of about 0.568.

Table 1: Model used and score obtained

Model Used	F1-Score
AdaBoost	0.562
KNearestNeighbors	0.489
Logistic Regression	0.552
Random Forest Classifier	0.568
SVC	0.538
XGBoost	0.567
Multi-Layer Perceptron	0.536

5 Models Ensemble

Not satisfied with the best score of a single model, we tried ensemble approaches using all the previously tuned models, first with voting and then with stacking.
Voting system takes the predictions of each classifier and then, after applying weights to each of them, produce the final output. Weights of the models are initialized basing on the score that each model reached individually, and then tuned trying to maximize the score on the test set, using linear programming optimization. With this approach, we obtain a F1 score on our test set of 0.564.
Stacking, on the other hand, takes as input the predictions of each classifier as before, uses logistic regression in order to combine the different prediction and thus best predict the two classes. With this approach we obtain a f1 score on the test set of 0.549.

6 Conclusions

Given these results, we decided to keep as our final model the Voting Ensemble. The ensemble of a variety of uncorrelated models could lead to a powerful generalization on real data, even if the score on the test set is slightly lower. Using this model we trained it on the whole set and used it to predict the missing labels of the test set given for the project. The predictions on the test

set show an unbalance similar to the training set with 81% of negatives and 19% of positives.