Assignment 1
Advanced Machine Learning
University of Milano-Bicocca
M.Sc. Data Science

Federico Signoretta
ID number: 847343
f.signoretta@campus.unimib.it
October 20, 2019

Prediction of default payments

The assignment consists in the prediction of default payments using a neural network. The dataset contains information on default payments, demographic factors, credit data, history of payment and bill statements of credit card clients in Taiwan from April 2005 to September 2005. The provided data comprises the training set that can be used for the training (and eventually for the validation) and the unlabelled test set.

1 Data Exploration

The dataset *train.csv* consists of 24 columns and $2,700$ rows; instead, the dataset *train.csv* consists of 23 columns and $3,000$ rows: in this case, there is no target variable. The target variable is labeled as *"default.payment.next.month"* and it represents the default payment of a costumer (1 =yes, 0 =no). The others features represent the costumer's qualities and his credit card status for each month. In particular:

- *"MARRIAGE"*: Marital status (1=married, 2=single, 3=others)
- *"EDUCATION"*: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
- *"AGE"*: age in years
- *"SEX"*: gender (1=male, 2=female)
- *"PAY_k"*: repayment status. It takes discrete values $p \in [-2, 8]$. If $p \leq 0$ than the client paid in advance and if $p \geq 0$ than the client paid late. Moreover, $k = 0, 2, 3, 4, 5, 6$ and it represents the month associated to the repayment status.
- *BILL_AMTj*: amount of bill statement, where $j = 1, 2, 3, 4, 5, 6$ and it represents the month associated to the amount of bill statement.
- *PAY_AMTj*: amount of previous payment, where $j = 1, 2, 3, 4, 5, 6$ and it represents the month associated to the amount of previous payment.

The most important observation is the problem about class imbalance: in general, it has a strong impact on the classification models. In this case, the *class-1* has 5,973 records (22.1$\overline{2}$%) and the *class-0* has 21,027 records (77.8$\overline{7}$%).

2 Data Cleaning

In order to have a cleaned dataset, the qualitative variables needed to be binarized. In particular, from "MARRIAGE" it was possible to create a new binary variables called "married": in this case, it was assumed that the characteristics "other" and "single" can be classified as not-married; from "EDUCATION" it was possible to create three new binary variables: "graduate_school", "university" and "high_school" (it was assumed that the characteristics "other"

and "unknown" were the same).
Moreover, the "SEX" variable was translated by one unit in order to have a binary variable where (0=male, 1=female) and the "AGE" variable was binarized, according with the "AGE" distribution: so, it was possible to create new variables based on the following age ranges: in "age_21_28" there are costumers aged between 21 and 28 years-old (25% of the distribution), in "age_29_34" there are costumers aged between 29 and 34 years-old (50% of the distribution) and in "age_35_41" there are costumers aged between 35 and 41 years-old (75% of the distribution).

3 Features selection

Firstly, it was observed the high correlation ($\rho \geq 0.9$) between the variables "PAY_BILLj": in order to avoid multicollinearity, the variables "PAY_BILL2", "PAY_BILL3", "PAY_BILL4" and "PAY_BILL5"were removed.
The dataset was split into training set (70%) with 18,900 and validation set (30%) with 8,100 records and it was used an *ExtraTreesClassifier* algorithm in order to determine the most relevant features. Observing the results, it was decided to keep all feature because all of them explain part of the target.

4 Neural Network

At this point, it was implemented a neural network in order to correctly identify as much as possible the binary classification problem. In particular, it was implemented a fully connected neural network with a hidden layer. More specifically:

1. an **input layer** with number of neurons equal to 23 (number of features) and activation function "RELU";
2. an **hidden layer** with 16 neurons and activation function "RELU";
3. an **output layer** with number of neurons equal to 2 (number of classes) and activation function "sigmoid";

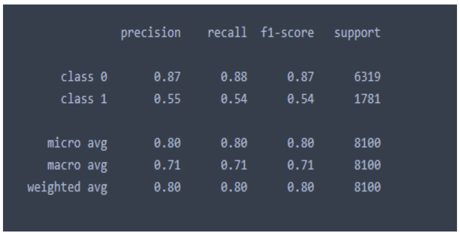
Different combinations of neurons and activation function have been tested, but the values of the model were basically the same. Hence, the choice to use a low number of neurons.
Before training the model, it was needed to configure some important parameters:

- as optimizer it was chosen the "RMSporp" optimizer
- as loss function it was chosen the "binary crossentropy" function
- as list of metrics it was set precision and recall. Of course, it was not considered the accuracy: it would not given useful information about the goodness of the model.

Different optimizers were tested: in particular, between "SGD", "adam" and "RMSporp", the last one performed better than the others. For training the model, it was set a batch size equal to 64 and epochs equal to 75: after several tests, it was observed that by increasing the size of the batch and epochs, the results did not vary much and in some case decreased.
Moreover, the most important parameter used is the "class_weight": through this parameter it was possible to balanced the model in order to give more weight to the rare class (class-1). In particular, it was set a weight equal to 1 for the class-0 and 2.75 for the class-0.

5 Results

Finally, the results of the neural network previously implemented are shown in figure (1) and figure (2).

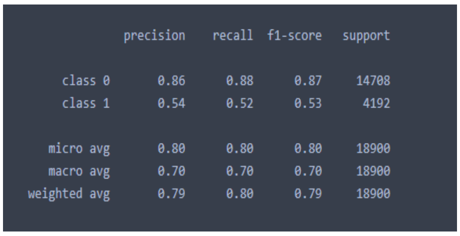


```
              precision    recall  f1-score   support

     class 0       0.87      0.88      0.87      6319
     class 1       0.55      0.54      0.54      1781

   micro avg       0.80      0.80      0.80      8100
   macro avg       0.71      0.71      0.71      8100
weighted avg       0.80      0.80      0.80      8100
```

Figure 1: *Validation Set*



```
              precision    recall  f1-score   support

     class 0       0.86      0.88      0.87     14708
     class 1       0.54      0.52      0.53      4192

   micro avg       0.80      0.80      0.80     18900
   macro avg       0.70      0.70      0.70     18900
weighted avg       0.79      0.80      0.79     18900
```

Figure 2: *Training Set*

From the results, it was possible to observe that the model has about the same performance both training set and validation set. In particular, the *F-measure* on the validation, for the rare class (class-1), is equal to 0.54 and on the training set equal to 0.53. Instead, for the class-0, the *F-measure* is equal to 0.87 both validation and training set.
Applying the model on the test set it was obtained the following results:

- Class 1: 18.6 %
- Class 0: 81.4 %

In conclusion, the percentage allowed of the class-1 is slightly less than the percentage on the training set. So, the neural network provides that 558 costumers - probably - will not pay the next month.