Visual Analytics on Credit Cards Defaults

Maria Ludovica Costagliola, Emanuele De Santis

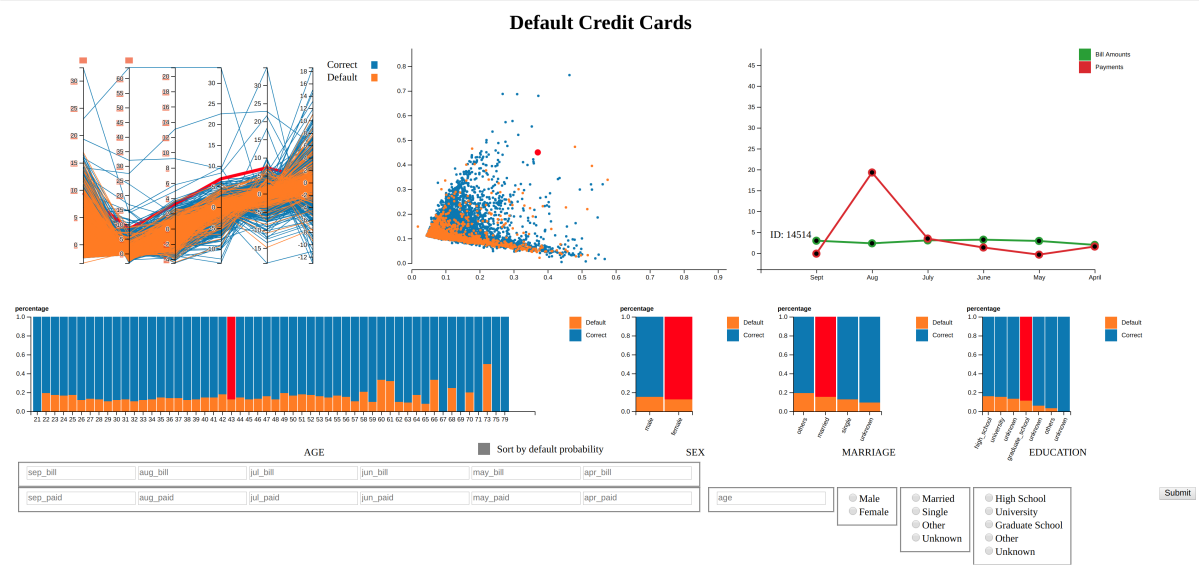

Fig. 1. Sceenshot of the entire interface

Abstract— The project was developed during the Visual Analytics course. It concerns the visualization of credit cards owners data in order to make the bank director knowing the customers that are supposed to not be able to pay the credit card bill in the next month. All data are represented using simple and well-known views that immediately highlights similarities among customers and give to the user an overview on all customers.
The system uses also a machine learning algorithm to classify new customers (manually added using a form) and represents them updating the views.

1 Introduction

After the paper presentation done during the lectures, we decided to focus our attention on a dataset related to bank transactions. Most of the bank transactions datasets are not public available (or they contains few useful information to protect users' privacy), but we were able to find a dataset related to this field.

We were thinking about the need for a bank director to always know how customers with a credit card issued by his financial institution behave. Particularly, we pay attention to the last payments and to the corresponding bank account balances of those customers.

From these data and from some other personal information of the customer (for example age, marriage status, ...), it is possible to identify the ones that probably will not be able to pay the credit card bill in the next month.

The prediction is done by a machine learning algorithm, but the result is useless if it is not combined with an efficient visualization of the whole data. In fact, with this visualization a bank director is able to better understand the result of the machine learning algorithm, considering also the similarity between the result and some preexisting patterns or clusters.

Moreover the bank director may be able to detect default customers even without using any machine learning algorithm, just seeing the information given by the visualization.

2 Dataset

The dataset used in this project is taken from UCI database [1] , it contains about 30000 tuples, each with 24 attributes. Some tuples lack of some values and so they were all removed in order to have a completely useful dataset. A few attributes for each remaining tuple were also removed because they were of no interest for us. We get in this way a more manageable dataset thanks to a lower number of tuples (15337) and of attributes (19). This computation was done by the python function `preprocessing.action()`, that creates a new csv file named `dataset.csv`.

About the used attributes, we can identify four of them that are categorical: age, sex, marriage status, education. These represent personal information about the owner of a particular credit card and they are used for statistical considerations. We have six numerical attributes named *Amount of bill statement*, one for each month from September 2005 to April 2005 and the corresponding six numerical, named *Amount of previous payment*.

The last attribute of each row is the *target*, that is the prediction about the ability to pay on October 2005.

3 Visualization

The visualization that we have developed is constituted by 7 views, each one that shows different aspects of the dataset. All the view are coordinated each other, so clicking on an element of a certain view

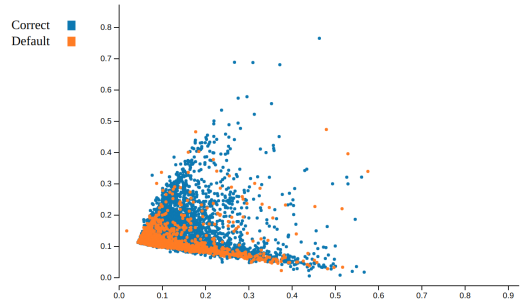

Fig. 2. Scatterplot view of the tool

will highlight some elements in the other views and/or will show other information. Moreover when a new customer is added, all the views change according to the newly added tuple.

The interface was tought to fit in a FullHD (1920x1080) screen, even if it works also if we resize the browser window (the minimun resolution supported is 1024x768).

3.1 Scatterplot

The scatterplot is the main view in the visualization. Each point represents a customer and the color encodes the classification of the customer (default or correct). The color scaling used is the `d3.scaleOrdinal(d3.schemeCategory10)`.

Due to the high dimensionality of the dataset, it was mandatory to apply a dimensionality reduction algorithm. We decided to apply PCA because it is fast and it doesn't require a deep knowledge of the problem, and we then took the first 2 components. This work is done by the python function `pca.action()` using `pandas` to read and parse the csv original dataset and `sklearn` to compute PCA values. The result is saved in `pca.csv`. A 2D scatterplot is the best way to graphically represent the relation between points and to spot possible clusters. In this case it helps the bank director to quickly see that most of the bad customers are grouped togheter in a particular area (in figure 2 in the left-bottom part of the plot).

Each point has associated two event handlers. The first one allows to make visible a tooltip showing the identifier of the customer just by mouseovering over it; this handler also makes the point bigger to have it more visible. The second handler is associated to click events: by clicking on a point, it will highlight the point itself, will trigger the highlighting of the other related elements and will show up the corresponding lines in the linechart.

3.2 Parallel Coordinates

The first view that can be seen in the interface is the parallel coordinates chart. We have six axis, that correspond to the first six principal components given by the PCA algorithm applied on the original dataset. Each line referrs to one single point that is the graphic representation of a customer. Since each customer can have the target attribute equal to either 0 or 1, we have decided to use two colors to distinguish among default and correct users.

4 Conclusion

Acknowledgments

References

[1] I. C. Yeh. UCI machine learning repository, 2016.