EE5183 FinTech Final Project Report- Credit card default prediction (Group 2)

Gregor Karl LIED, David Philip LIN, Dominik HAPPEL, Lewis Marc Edouard MACE, Guillaume DESERMEAUX

unsupervised learning

***Abstract*—This project deals with a recurrent subject of the Fintech area which is the credit card default. More generally, this research tries to answer to the question of the most efficient unsupervised learning algorithm. Our goal is to us a dataset of 30,000 samples to explore the possibilities and performances of different machine learning and deep learning models. To achieve this goal, we played on different hyperparameters an tried different manipulations technique on the training set. We defined several metrics and indicators that are relevant to our problems. We compared the obtain results and conclude on the relative efficiency of each model.**

***Index Terms*—Credit card Default Prediction, Fintech, Risk-monitoring, SVM, Decision tree, K-Nearest neighbors, MLP**

I. Introduction

Credit Card Default happens when clients fail to adhere to the credit card agreement, by not paying the monthly bill. Risk & Credit departments in financial institutions, try to detect the costumers that tend to default in the future, not only to secure money, but even make more money as they could demand higher interest rates for more risky costumers.

Meanwhile, the digitalization is not only changing today's society but also companies' business models, in particular of the financial industry. In general, the large variety of digitalized processes and connected devices (Industry 4.0) generates a huge amount of data which can be used to extract valuable insights.

In this research project, we want to test the performance and usability of different machine learning and deep learning models to analyze private financial data to predict credit card default, by detecting clients that will not be able to pay their debts next month.

In the next part, we are going to describe the dataset and the methods that we used for the experiments. These one are the K-nearest Neighbors, the decision tree, the support vector machine and multi-layer perceptron.

Guillaume André DESERMEAUX (a08522201), Department of mechanical engineering, Nationnal Taiwan university

II. Material and Method

A. The dateset

To test the performance of the different models, we used data from a important bank in Taiwan with credit card holder as the targets. The value that we want to predict is "default payment next month" and it is a binary variable (Yes = 1, No = 0) that tell us if the customer will be in a situation of credit card default the next month. Overall, there are 30,000 samples dataset and 5,240 of them were in situation of credit card default (label = 1). This represent 17.5% of the customers.

The 24 features of the dataset are the following:
- ID: The id of the credit card user
- LIMIT_BAL: The amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
- SEX: the gender of the user (1=male, 2=female)
- EDUCATION : The education level (1 = graduate school; 2 = university; 3 = high school; 4 = others).
- MARRIAGE : The Marital status of the user (1=married; 2=unmarried).
- AGE : The age (year)
- Pay_X (X=1, …,6) : The repayment status from April to September 2005 (-1=pay duly, 1=one-month delay, …, 9=nine-month delay and above)
- BILL_AMTX (X=1, …, 6) : The amount of bill statement in NTD from April to September 2005
- NEXT MONTH DEFAULT : The default situation on the next month (1=yes, 0=no).

The data was randomly divided into 3 groups:
- training set (80%) to train the different models
- validation set (10%) to compute the accuracy and other metrics of the models and validate their performance.
- Testing set (10%) to test the different models and make a prediction.

Then, the data is normalized according the training set and along this formula:

$$zi = \frac{xi - \min(x)}{\max(x) - \min(x)}$$

Guillaume André DESERMEAUX (a08522201), Department of mechanical engineering, Nationnal Taiwan university

In case where a dataset is really unbalance, the prediction of our models can be not optimal. To improve the accuracy and because of the unbalance nature of the data (cf. fig 1), we will propose different approaches to make the two categories even. The first one is called down sampling and consist into randomly cutting off some samples in the biggest category (here label = 0). This technic can be efficient but reduced a lot the dataset size and then can augment the generalization error.

The second one is the SMOTE (Synthetic Minority Oversampling Technique). At the opposite, this on consist into creating new samples for the minority class (here label = 1) using interpolation. To create a new sample, we randomly take 2 sample in the minority class and define the value of the new sample as the averages of the 2 others. This approach seems better because we don't cut any information in the original data and creates new points likely to really exist. But a big drawback is that we lose the meaning of some categorical data like SEX, MARRIAGE and EDUCATION which take float values instead of integer.

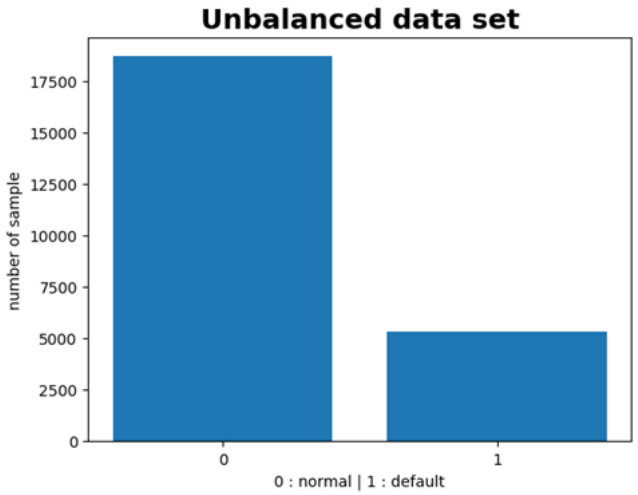

Fig. 1. The repartition of the targets inside of the dataset. Only 17,5% of the samples are defaults.

B. The different machine learning and deep learning techniques

We have chosen to use 4 different methods in our research that are the following.

1) K-Nearest neighbor (KNN)

This model comes from a simple assertion that the objects from the same class are close the one from the others. For every point P, it calculates the distances between P and the others in the n-dimensional space. After this, it considers the K closest points and gives to the P the label with the greatest presence. The hyperparameter that can be modified is the K representing the number of neighbors that influence the class choice.

2) Decision trees (DT)

This model creates a decision tree following several binary decision rules. Indeed, each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes. The binary test are chosen to minimize the impurity, which refers to the the measure of variability of the response values of the observations. This model has the advantage to be able to handle categorical data which is a good thing for the data that we have. The hyperparameter of the model that can be tuned is the tree depth.

3) Support vector machine (SVM)

Intuitively, the SVM try to create a hyperplane H: $w^T x+b$ that can separate the data into 2 classes. The distance between the hyperplane and the nearest data points are called the margin. The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set, giving a greater chance of new data being classified correctly. Furthermore, this model uses the kernel trick to resolve nonlinear problem. The data will continue to be mapped into higher dimensions until a hyperplane can be formed to segregate correctly the points. The principal hyperparameters that can be modified are: the kernel function, margin and others kernel-dependent parameters.

4) Multi-layer perceptron (MLP)

MLP is the only deep learning model that we will try in this research. It is composed of several fully connected layers with activation functions. This model is based on the application of weights (that were determined during the training) and activation function at each layer. The model is made of an input and output layers and generally several hidden layers. The parameters that can be modified are the number of hidden layer, the number of units, the activation function and the optimizer.

III. EVALUATION

A. The metrics and comparison methods

The comparison between the different models required us to choose the most adapted test metrics for the problem. Indeed, it is interesting to consider the accuracy (1) to have an overall though about the model efficiency. But to get more precisions into that achievement of the models, we also consider the sensitivity (or recall) and the specificity. A high sensitivity (2) means that the model predicts well the positive class which means the defaults. Furthermore, a high specificity (3) means that the model predicts well the negative class or in other words, the non-default.

$$(1) \quad Accuracy = \frac{True\ positive + True\ negative}{Positive + Negative}$$

$$(2) \quad Sensitivity = \frac{True\ positive}{Positive}$$

$$(3) \quad Specificity = \frac{True\ negative}{Negative}$$

As well, we will trace the gain curves for each model. This graph is interesting because it shows us the real efficiency of a model in terms of people positively predicted in function of the targeted people percentage. The gain curves are obtained after the following process. We make a prediction of the validation test with the model and rank the results from the most to the less likely to default. The we trace a curve with on the X axis, the customer base we want to target with the campaign in the ordered list. The Y axis gives us the answer to what is the percentage of all positive response customers have been found in the targeted sample. This graph shows us how much a model improves the resolution of our problem. Additionally, if the objective of a company is to predict a certain percentage default, this graph gives us how many people need to be carefully monitored.

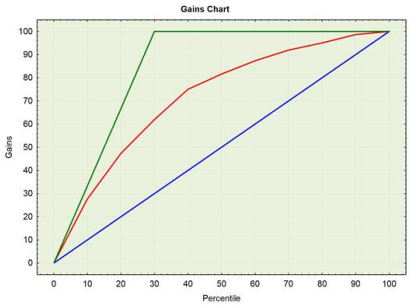

Fig. 2. An example of gain curve (in red). The green curve represents the maximum that can be obtained (100% of targeted people are true). The blue curve represents the probability of the random guess.

Finally, we trace the confusion matrices for each model. This matrix gives us the repartition of True negative/positive and false negative/positive.

B. Hyperparameter tuning

To get the better performances with each model, we try to find the best hyperparameters possible. To compare the different models, we compute the accuracy of the models with different parameters. And thanks to a grid search, we are able to select the parameters that give us the best results.

We found the following result for the non-modified data:

- KNN: K = 11
- Decision tree: tree depth = 4
- SVM: kernel = "RBF", C = 10, gamma=1
- MLP: hidden layer = 3, hidden units = 65, 25 and 45 for the 1st, 2nd and 3rd layer, activation function = Relu, optimizer = Adam

In the following part, all the models are trained with the optimal hyperparameters. We can expect the more elaborate models to perform better in the prediction, especially the Neural network one.

IV. Discussion

Here are the results that we obtained:

Model	Type of model	Accuracy (%)	Sensitivity (%)	Specificity (%)
K Nearest Neighbors	*K = 11*	80	31	94
Decision Tree	*Unbalanced Classes*	82.8	39	92
	Balanced Classes	73	60	73
	SMOTE	81	44	91
SVM	*RBF Kernel Unbalanced Classes*	83	34	96
	RBF Kernel Balanced Classes	78	55	85
MLP	*Unbalanced Classes*	81	24	98
	Balanced Classes	73	63	76

Fig. 3. Accuracy, sensitivity and specificity that we obtain with our different models. We tried down sampling and oversampling methods on some of them

We surprisingly see that the best accuracy is not given by the MLP bur the SVM and decision tree with unbalance class. Furthermore, the best sensitivity that we get is around 60% with the decision tree and down sampled dataset.

The results with the modified datasets are not better in terms of accuracy. However, we see an improvement in the sensitivity for all the models. But the specificity is also being reduced a lot so we cannot consider the techniques to modify the dataset improve the results.

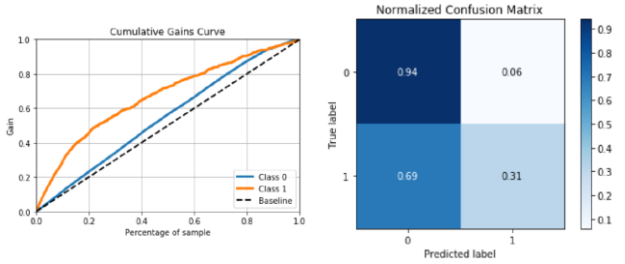

Fig. 4. Gain curve and confusion matrix for the KNN model

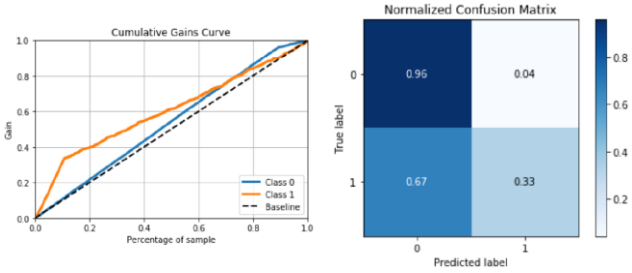

Fig. 5. Gain curve and confusion matrix for the Decision tree model

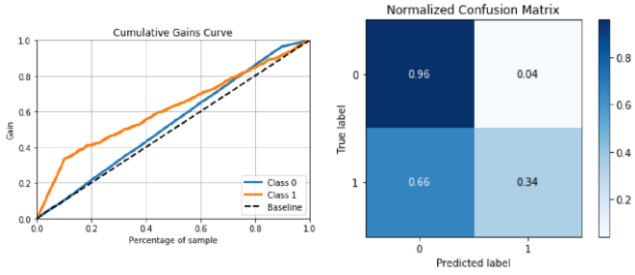

Fig. 6. Gain curve and confusion matrix for the SVM model

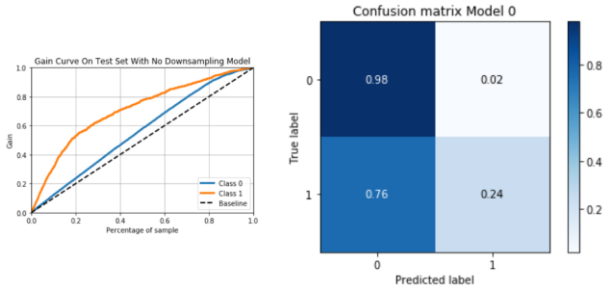

Fig. 7. Gain curve and confusion matrix for the MLP model

The confusion matrices show us that the models perform well on the prediction of the prediction of the negative class. However, the prediction of the positive class is not so good and reach only 24% to 31% of correctness.

On the lift curves, we particularly pay attention to the Class 1 curve. We can see that the lift curve for the KNN and decision tree have a sharp edge and for more than 10% of ranked samples, the slope drops brutally. It also seems like the prediction of the MLP is a lot more likely to predict the customer in a default situation with a small portion of targeted customer. We can then conclude as expected, that the Multi-layer perceptron performs better in the resolution of our problem.

REFERENCES

[1] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2):2473–2480, 2009.

[2] Jazmin Quezada, University of Texas at El Paso Forecasting Crashes, Credit Card Default, and Imputation Analysis on Missing Values by the use of Neural Networks, 2019

[3] Q. Wang, Y. Hu, J. Li, “Community-based feature selection for credit card prediction”, Jan. 2018, [Conference paper]

Work repartition:

Dominik: KNN

Guillaume: Decision tree

David: SVM

Gregor: Data analysis & NN