Prediction of Credit Card Default using Machine Learning Algorithms

Mohamed SERHIR
Faculty of Engineering Environment and Computing
Coventry University
Coventry, England
Serhirm@uni.coventry.ac.uk

***Abstract*—This paper is intended to predict whether a client will be defaulting the following month by using different machine learning algorithms. In this paper, several data preparation methods were implemented, to clean the data set, encoding the categorical features, covering class imbalanced issues, and feature scaling. For each machine learning algorithm, many parameters have been tested to optimize the models. Then the results and the performances are discussed.**

***Keywords*—machine learning; classification; python; scikit-learn; class imbalance; dimensionality reduction; feature encoding; logistic regression; naive bayes; random forest; k-nearest neighbors classifier.**

I. INTRODUCTION

We are now living in a world where credit cards are increasingly replacing cash, where payments are more and more virtual and accessible. People forget its real value and are unfortunately encouraged to spend money they don't have. It is quite possible to buy products, clothes or simply anything and pay only a few days later, or even at the end of the month. That's what credit cards offer, all the money you spend is loaned money, which you'll probably pay back when your salary comes in. Banks offer this advantage, but the problem is clear: what if a customer cannot repay? What if many customers can no longer payback? What are the consequences for both the bank and the customer? Well, there are usually serious economic consequences, depending on the country, the bank, and the customer, but more globally, can this be avoided? Can a default of payment be avoided?

A decade ago, to expand their economy, Taiwanese banks excessively issued credit cards, especially to unqualified workers. This action resulted in a debt crisis in February 2006 where debt from credit cards reached more than $250 billion [7].

This is a real example, a real crisis where our application can be very useful: Predict if a customer will have a default credit card or not using a machine learning algorithm. To do this case study, I will use a data set from the UCI website, the data is about payment data in 2005 in an important bank in Taiwan, sourced by I-Cheng Yeh [8].

II. LITERATURE REVIEW

A. Data mining

It is a technique or a process that is used to extract usable and useful data (get knowledge) from raw data sets. I-Cheng Yeh and Che-hui Lien used six different data mining methods in 2009 to compare the predictive accuracy of the probability of default with those methods [3]. In their paper, they conclude that there are few differences between the methods for the errors rate but a lot of differences in the area ratio and they conclude that ANN is better to score clients. I-Cheng Yeh is the author of the dataset which we will explore and describe a little later.

B. Classification

That refers to a predictive problem, we want to predict the class of input variables. It belongs to the supervised learning approach, where we provide targets with the input data. In our problem, we are going to predict if a customer will have a credit card default or not. There are many classification algorithms, for this use case models adapted to the data set has been selected, but to conclude it is important to compare the results of different models. [9]

C. K-Nearest Neighbor

This method can be used for regression predictive and classification problems, it tries to classify a data point in a provided data set. To do that the algorithm will compare the new data to the nearest point and the classification is based on the closest points [5].

D. Naïve Bayes

It is an algorithm based on the Bayes Theorem and considers that the predictor variables are independent and have no effect on each other. According to the article [4], the theorem determines the possibility that an event will occur, taking into consideration the probability that another event has already happened. There is however a weakness in the use of this method, the accuracy is related to the hypothesis of conditional independence of the class, there may be dependencies between variables

E. *Logistic Regression*

It is a predictive algorithm based on the concept of probability. To match the predicted values to probabilities, we use a sigmoid function.

F. *Related work*

There is other work on the same topic, regarding the prediction and/or detection of credit card fraud. Most fraudulent transactions are related to stolen cards, the work of [4] and [7] uses algorithms such as Neural Networks, Regression, and Decision Tree learning to predict payment fraud and compare different techniques.

III. The Data Set

The data set used was obtained from the UCI's machine learning repository. It contains 30000 instances and 24 attributes with no missing values. Each instance represents some information about the account of anonymous clients of a Taiwanese bank. This data was used before by I-Cheng Yeh in his case study in 2009 [8], and he published the data set in 2016.

TABLE I. Data Set Features

Key	Description	Type	Value Range
LIMIT_BAL	Amount of the given credit	Numerical	10000 to 1000000
SEX	Gender	Categorical	0 to 1
EDUCATION	Education level	Categorical	0 to 3
MARRIAGE	Marital status	Categorical	0 to 1
AGE	Age in years	Numerical	21 to 79
PAY_1	Repayment status in September	Categorical	-2 to 8
PAY_2	Repayment status in August	Categorical	-2 to 8
PAY_3	Repayment status in July	Categorical	-2 to 8
PAY_4	Repayment status in June	Categorical	-2 to 8
PAY_5	Repayment status in Mai	Categorical	-2 to 8
PAY_6	Repayment status in April	Categorical	-2 to 8
BILL_AMT_1	Amt bill statement in September	Numerical	-165580 to 964511
BILL_AMT_2	Amt bill statement in August	Numerical	-69777 to 983931
BILL_AMT_3	Amt bill July in September	Numerical	-157264 to 1664089
BILL_AMT_4	Amt bill statement in June	Numerical	-170000 to 891586
BILL_AMT_5	Amt bill statement in Mai	Numerical	-81334 to 927171
BILL_AMT_6	Amt bill statement in April	Numerical	-339603 to 961664
PAY_AMT_1	Amt paid in September	Numerical	0 to 873552
PAY_AMT_2	Amt paid in August	Numerical	0 to 1684259
PAY_AMT_3	Amt paid in July	Numerical	0 to 896040
PAY_AMT_4	Amt paid in June	Numerical	0 to 621000
PAY_AMT_5	Amt paid in Mai	Numerical	0 to 426529

All amounts are in NT Dollar and the data is for the year 2005. The categorical features are already ordinal encoded, but there are some ambiguous categories that I will handle in the data preparation part. Table I represent the data set after some small improvements regarding the key name and the categorical value range. The features can be divided into two groups, personal information that can show us who is the customer and personal history that can show us what the customer did in the past. The first feature represents the limit of credit including the individual consumer credit and his/her family. Then there is some information about the customer: the sex, age, education, and marital status. And 3 important features: “Pay”, “Bill amount” and “Pay amount”, each represents stores a history from September 2006 to April 2006. The features are named by _1…_6, where 1 is for September 3 for July, and 6 for April. “Pay” is categorical, it can be -2 for no credit, -1 if the customer has duly paid his credit, 0 if there is the use of revolving credit and a number between 1 and 9 that represent the number of months delay (where 9 means 9 months or more). “Bill amount” represents the total amount of bill statement paid in the past. And “Pay amount” stores the number of previous payments.

The target is DPMN for “Default Payment Next Month”, which the goal is to predict if it will be no or yes (0 or 1).

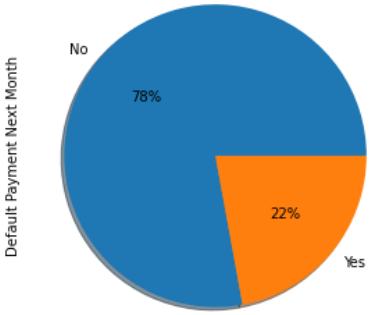


Fig. 1. Distribution of classes in the target column

The data does not show a large imbalance between classes, but it may cause some classification problems, as most machine learning methods are better when classes have an equal number of values. It is not hard to get a high accuracy by predicting the majority class, but for the minority class, it can have more errors because the model is designed to maximize the overall accuracy and reduce errors. It is why in this case, 3 types of data will be created then tested and compared.

IV. Data Preparation

A. *Feature encoding*

Machine learning models can only deal with numerical values, and in the most data set, we can find numerical and categorical features. Feature encoding is one of the most important In the data set used for this use case, as well as in many data sets, there are two types of features, the numerical features containing numerical values, which are sometimes quantified in different measures, but also categorical data, which are often expressed in words. There are also features containing only text, but we will not explore this because our study contains only numerical and categorical values. Most machine learning models require numbers to work properly, which is why encoding is a very important part of the data pre-processing.

The categories in the data set were already encoded in numbers, but some small improvements were required: the data isn’t binary encoded. Even if after encoding our data only contains numerical values, machine learning algorithms can give more weight to some categories (i.e. the education level

can take a value between 0 and 3 and the model can classify the category 0 or 3 in a different way. In this paper, the sex category was encoded as 0 for male and 1 for female, marriage category as 0 for married and 1 for not married because the author of the data set didn't provide information about the first value, and single means not married. To simplify the classification, it's better to have only two categories (0 and 1). *Fig. 2.* shows the modifications regarding the first three categorical features.

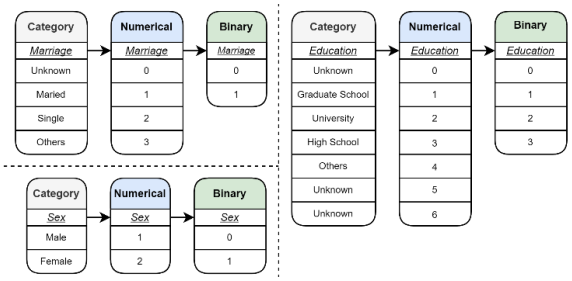


Fig. 2. Binary encoding for the first 3 categorical features.

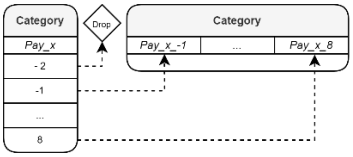


Fig. 3. One Hot Encoding & Removing Dummy Variable

The last categorical column, "Pay 1...6" was encoded using a python library (Pandas) with the get dummies function that converts the categorical variable into an indicator variable, as shown in *Fig. 3*.

B. Normalisation / Standardisation

After having carried out a binary encoding, it is also important to put all our values on the same scale, for these two very well-known techniques exist: normalization and standardization. Standardization is used to scale the data to have an average of 0 and a standard deviation of 1, while normalization is used to scale values between 0 and 1.
In this use case, MinMaxScaler was used for resizing the dataset independently at the feature level.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1) \qquad x_{scaled} = \frac{x - \mu}{\sigma} \quad (2)$$

C. Class Imbalanced

As shown in *Fig. 1.*, in this paper, there are 23364 samples in the majority class (no default payment next month: 0) and 6636 samples in the minority class (default payment next month: 1). To solve the class imbalanced issue and to get a more accurate result, three different techniques were used, some functions have been created to get the majority class and the minority class to create the train sets resampled using different techniques. The code is available in the appendix.

1) Random undersampling

It is used to randomly remove 13322 samples from the majority class to get 5339 samples for each class. Only the train set was under-sampled, the test set is used for testing the models and do not needs to be resampled.

2) Random oversampling

It is used to randomly duplicate 13322 samples from the minority class to get 18661 samples for each class. As for the under-sampling, only the train set has been duplicated.

3) Smotenc

Synthetic Minority Over-sampling Technique is used to create synthetic data by creating new data points that are mid-way between two neighbors. This is an oversampling method, but opposite to random oversampling smote will not duplicate values.

D. Dimensionality Reduction

In machine learning, a high number of features may negatively affect the efficiency of a model. To simplify and speed up our learning process, it is possible to reduce the total number of variables in a data set by exchanging a high number of variables with controlled accuracy. In this paper, feature extractions methods were used, Principal Component Analysis and Linear Discriminant Analysis. After some tests in Python, reducing dimensionality saves several tenths of minutes and much more especially when searching for the best parameters for an algorithm with a very complex computation such as KNN.

1) PCA

PCA is an unsupervised algorithm that looks for variables with the most variance and because it's unsupervised it ignores class labels. PCA was implemented here in Python by using the scikit-learn library and the cumulative explained variance was displayed with matplotlib library in *Fig. 4.*

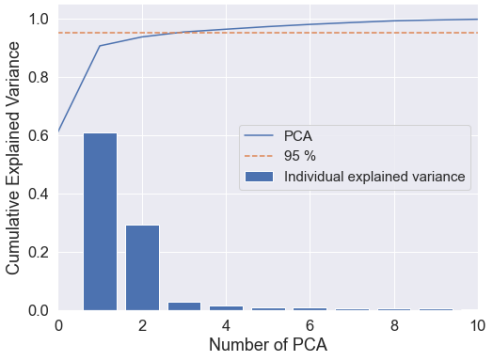


Fig. 4. Cumulative & Individual Explained Variance

The results indicate that the first 2 variables have the highest variance. The cumulated variance of the first 2 variables reaches 95% which is an acceptable loss of information to save time and simplify the training of the model.

2) LDA

LDA is a supervised technique that attempts to maximize the separation of the categories. This method is mostly used in classifications problems where you have a categorical output variable. It allows two main things to be done: binary classification and multi-class classification.

V. Classification Techniques

A. K-Nearest Neighbors Classifier (KNN)

K-NN is one of the most popular algorithms in machine learning. It compares each instance to a k number of nearest training instance by calculating the distance between each point. In this paper, various parameters were tested to obtain an optimal result. The function "GridSearchKNN" has been created to search for the best accuracy by testing a different number of neighbors between 1 and 30, different metrics for the calculation of distances between each point (Euclidian, Manhattan, Minkowski or Chebyshev) and with two different weights (uniform and distance).

B. Random Forest (RF)

This method consists of creating decision trees using the samples. The prediction of each new sample is made by voting for the best solution. The result is the average of the result of each tree. The function "GridSearchLR" was created for testing different parameters: the number of estimators (5, 10, 100, 200) and the max features (log 2, sqrt or auto).

C. Naïve Bayes (NB)

Naive Bayesian classification is a type of simple probabilistic Bayesian classification based on Bayes' theorem with a strong independence of assumptions. This classifier has several properties: the dissociation of the conditional class probability laws between the different characteristics results in the fact that each probability law can be estimated independently as a one-dimensional probability law.

D. Logistic Regression (LR)

LR is another popular algorithm used for classification problems. Different from K-NN, this algorithm predicts the probability for an instance to belong to a specific class. In this paper, binary regression was used. This method needs a large data set to give a more real and accurate probability, we have a data set of 30000 rows with 2 independents targets. The function "GridSearchLR" was created for testing 10, 100, and 1000 max iterations.

VI. Discussion & Results

All algorithms, representations, and functions have been coded in python using Jupyter notebook. Libraries widely used in the world of data science were also used, such as Pandas, matplotlib and scikit learn. The imported libraries, the code created, and the results can be found in the appendix. The execution of the algorithms was performed on a laptop with an Intel i5 CPU having two physical cores clocked at 2.5GHz and 24 GB of RAM. The power of a computer is very important for machine learning problems, at least 16 GB of RAM to avoid a "Ran Out of Memory Error" if a notebook is used. Scikit learn provides a multicore option (n_jobs), having more clock speed and cores can make a difference in terms of time spent.

Each machine learning model was trained with different parameter as explained in *Part V.*, Pipelines and Grid Search were also used to obtain the best results. Using these methods can take a lot of time but for this paper, it was within reasonable limits. However, for a larger data set, using Random Grid Search is preferable, it works in the same way but instead of testing every single parameter, it selects random parameters. The model evaluation metrics used for the research of best parameters were the best accuracy (recall, precision, and f1 are more important for our use case and will be discussed) and with a 5 fold for the cross-validation. The cross-validation is used to "validate" the accuracy, it will take 5 different parts of the training set and calculate 5 times the accuracy then the mean of the 5 results is conserved. Cross-validation allows to avoid overfitting cases as much as possible and to give a score closer to reality, because it's sometimes possible by luck during the calculation of the score that the algorithm selects samples that are easy to predict, so we want to confirm this score on several games.

Imbalance data set problems have been demonstrated earlier in this paper; it is why all algorithms were tested with 4 data sets:

- The original imbalanced data set binary encoded and divided into training and testing sets. (1)
- A new training set sourced from the (1) but down-sampled.
- A new training set sourced from the (1) but up-sampled.
- A new training set sourced from the (1) but up-sampled using smotenc.

Confusion matrices will also be used for showing and explaining the results. In this paper, the size of the confusion matrices is 2 x 2 because it's a binary classification problem. To generate a confusion matrix, the confusion matrix method from scikit learn was used. A function was also created to plot the result of this matrix and another function was created to calculates the different scores: accuracy, precision, recall, and f1.

$$accuracy = \frac{(tp+tn)}{(tp+tn+fp+fn)} (3), precision = \frac{tp}{(tp+fp)} (4),\ recall = \frac{tp}{(tp+fn)} (5),\ f1 = \frac{(2*precision*recall)}{(precision+recall)} (6)$$

The equations (3), (4), (5), and (6) show how the different scores were calculated. Accuracy is the most important score because it shows how the model is accurate but in our case, it's

not a good indicator. Our models can exceed 80% accuracy, but due to the complexity of the problem of predicting a category, with a majority and a minority category as samples, it is naturally easier to get very good results on the majority category and consequently to increase the total accuracy. However, the precision (3) remains very poor in this use case, models have much more difficulty predicting target 1 (will have a default payment) than target 0 (which is most cases). The precision is therefore used to indicate an error ratio concerning the prediction of the positive class. The recall (5) score is also interesting, it is used to know the ratio of positive predictions over the class yes.

As the goal is to predict whether a person is going to be in default of payment, it is not very bad if a person is wrongly predicted, but it is important to minimize false negatives. The best model will therefore be a model with a good overall score and a very good precision score.

A. *K-NN*

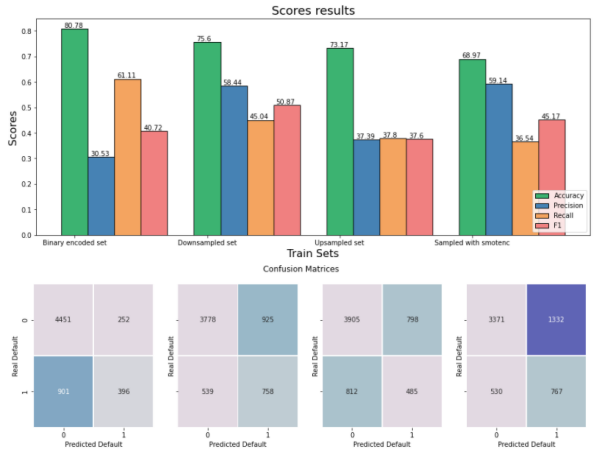
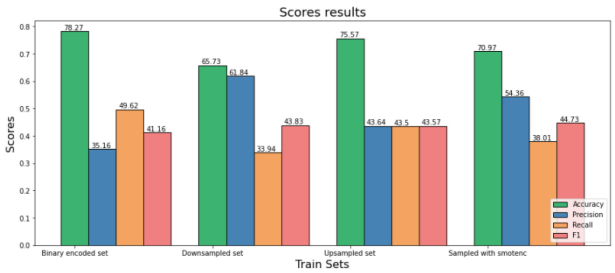


Fig. 5. K-NN Scores Results & Confusion Matrices Using PCA & Minmax

As expected, with an imbalanced data set, the accuracy is very poor, the number of false negatives is 3 times higher than the number of true positives. When trying to resample we get good results for the down-sampling technique and some moderate but balanced results for the up sample. It is with the smotenc technique that a result close to the objective is obtained, with the minimum of false negatives, the down-sampling technique comes second.

B. *Random Forest*



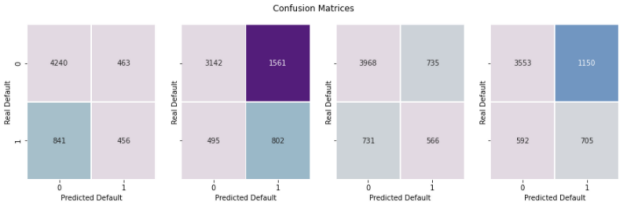


Fig. 6. RF Scores Results & Confusion Matrices Using PCA & Minmax

Results overall are quite like the K-NN algorithm, with this time a better precision score with the down-sampling technique but with nearly 40% more false positives than with smotenc. It is however preferable to monitor 1561 clients and to have missed 495 defaulters than to monitor fewer clients and end up with 592 defaulters. So, using a down-sampled training set is better got Random Forest.

C. *Naive Bayes*

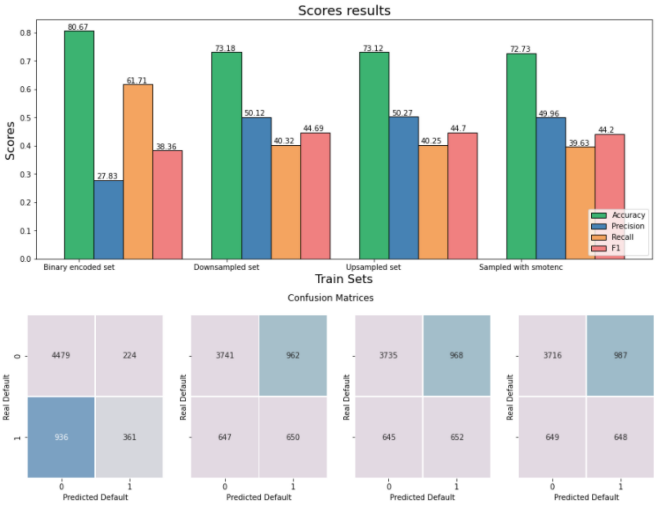


Fig. 7. NB Score Results & Confusion Matrices Using PCA & Minmax

Slightly poorer results, but once again, the most cost-effective resampling methods are down-sampling and smotenc.

D. *Logistique Regression*

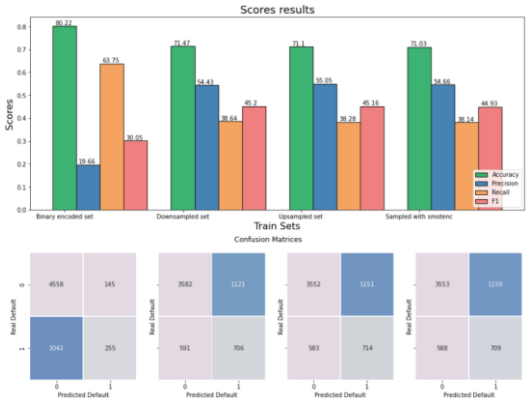


Fig. 8. LR Scores Results & Confusion Matrices Using PCA & Minmax

The results here are very similar between the resampled training sets, but down-sampling and smotenc up-sampling are slightly better.

E. Models comparison

To compare the different models, only the down-sampled and up-sampled (with smotenc) training sets will be used. Because the training set with no resampling does not represent the reality and the up sampling does not show very good results.

TABLE II. MODELS COMPARISON

Classification model	Accuracy		Precision		F1 score	
	Ds	Smote	Ds	Smote	Ds	Smote
K-NN	76%	69%	58%	59%	51%	45%
RF	66%	71%	62%	54%	44%	45%
NB	73%	73%	50%	50%	45%	44%
LR	72%	71%	54%	55%	45%	45%

According to Table II. And the different results, K-NN algorithm, with min-max scaling applied and a good preprocessing with an under-sampling of the training set, is the best option.

VII. CONCLUSION

Long pre-processing work is very important before using a machine learning algorithm. It is also important to test many parameters and to look for objectives according to the problem. The results show the importance of resampling the data set, something that has not been considered in other papers. Also, a size reduction was performed, which resulted in fewer features with a controlled loss of information.

In this paper, a solution to a real problem has been put forward. In a world where online banking continues to grow, machine learning remains to be an essential method for detecting fraud (which avoids the need for a refund and investigations) but above all to be sure that the customer will be able to pay his credit without disappearing. Other applications can be resolved with machine learning in this field, but only the two biggest sources of problems, the most expensive ones have been mentioned. Speaking of costs, the application of this problem can be carried out in any bank, the cost of the engineers who will process the data, prepare it and then select the most suitable model will certainly be lower than the cost resulting from the time lost in carrying out the administrative procedures during a high number of payment defaults. As a reminder, the most important objective remains to keep the highest possible accuracy to catch the maximum of defaulter even if more innocent persons are suspected of having a default payment the following month.

The previous contribution compared machine learning algorithms using different data mining methods. In this paper, more than ten years after, the benefits of python and the growing use of open-source libraries have been demonstrated. Indeed, scikit learn has started to be popular in 2015 and allows the users to focus on data preprocessing and model selection rather than testing many parameters.

Professional considerations & ethics

The data set was completely anonymized, any data privacy policy was violated. This same data set was used to analyze, evaluate, and predict a feature using different machine learning algorithms and no to categorize people. One of the biggest challenges in machine learning and decision-making is to avoid any type of bias. When a lot of specific information about minority people is in a data set, it's important to analyze if that does affect the model and the prediction. In this paper, only the age and maybe the limit amount of credit can represent a category of people, but the age wasn't an important feature. Moreover, limit credit can be the same for everyone, rich, poor, worker… No salary was represented in the data set, for example.

VIII. REFERENCES

[1] Al-Khatib, A. (2012). Electronic Payment Fraud Detection Techniques, http://www.wcsit.org/media/pub/2012/vol.2.no.4/Electronic%20Payment%20Fraud%20Detection%20Techniques.pdf

[2] Allen, M. (2020). SMOTENC – Python for healthcare modelling and data science. Retrieved from https://pythonhealthcare.org/tag/smotenc/

[3] Balu, B. (2020). Naive Bayes Classifiers. Medium. Retrieved from https://medium.com/towards-artificial-intelligence/naive-bayes-classifiers-79ac2c805f3f

[4] Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602-613. https://doi.org/10.1016/j.dss.2010.08.008

[5] Kotsiantis, S., Zaharakis, I., & Pintelas, P. (2006). Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, *26*(3), 159-190. doi: 10.1007/s10462-007-9052-3

[6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., & Grisel, O. et al. (2020). Scikit-learn: Machine Learning in Python. Jmlr.csail.mit.edu.https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

[7] WANG, E. (2020). Taiwan's Credit Card Crisis. Sevenpillarsinstitute.org. Retrieved from https://sevenpillarsinstitute.org/case-studies/taiwans-credit-card-crisis/

[8] Yeh, I. (2020). UCI Machine Learning Repository: default of credit card clients Data Set. Archive.ics.uci.edu. Retrieved from https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[9] Yeh, I., & Lien, C. (2009). The comparisons of data mining techniques for the predictive accuracy of the probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480. https://www.sciencedirect.com/science/article/abs/pii/S0957417407006719

IX. APPENDIX

The entire notebook can be found in this drive: https://drive.google.com/drive/folders/1hGE1HnvpflgwEW0-n-hmyrOBUGRPChI9?usp=sharing