Machine Learning in Predicting Credit Card Default

Yirun Wang aaron.wang@wisc.edu
Fenghang Yao fyao4@wisc.edu
Yunqing Xiao yxiao75@wisc.edu
Shupeng Tang stang72@wisc.edu

Abstract

With the changes in the economy world during the past few years, bank default, which refers to the failure to repay a debt, is becoming a hot topic in the finance industry. Our objective for this project is to use general features of credit card clients to predict whether there will be default payment next month. We used 30,000 sets of information of credit card clients in Taiwan from April 2005 to September 2005 as our dataset and used machine learning as well as deep learning techniques to forecast default payment on next month. In this project, we implemented Gradient Boosting, XGBoost, Logistic Regression and Random Forest machine learning algorithms to train our model and compared the training and test accuracy between each models. We also cover ways including hyperparameter tuning to improve the model, with the goal of finding the best predicting model for the credit card default.

1. Introduction

Nowadays, a lot of people have to rely on credit card to live. However, it is also a big problem for credit card company to predict whether this client would default on a credit card. In fact, the bank and the credit system used various of strategies to prevent and regulate the credit default issue. In the graph above [3], we can see that the overall default rate on Credit Card Loans from all commercial banks are declining in the past 30 years in the US.

From this distinct progress, we are interested in looking in depth about how to identify the risk of credit card default and what factors or characteristics of the clients might lead to the credit card default in order to prevent the issue in advance.

In this project, we would use some features like sex, education, marriage, age, repayment, and billing statement, etc to predict whether this client would default on a credit card in the future. The data for this research is extracted from the UCI Machine Learning Repository[6].We would further use boosting, Logistic Regression, random forest, decision tree and so on to build a complete process to make the outcome more precisely.

1.1. General Takeaways

Additionally, as a add-on of this study, we learn more about how the market evaluate the clients in general based on the characteristics.

Looking on a more board perspective as market risk management, the prediction and accuracy result of the study will be valuable beyond simply defining client as credible and non-credible clients. We'd learn what kind of traits of a client tends to be more "trust-worthy" in aggregate and have more knowledge on the market evaluate in credit card companies.

2. Related Work

In 2009, Yeh and Lien set up a research to examine customers' default payments in Taiwan. In the research, they compared six data mining measures.They first divided 25,000 payment data into two half, one half for training set and another half for test set and they used a standardised dataset to compare the accuracy of six data-mining techniques for predicting credit-card clients' default probability: K-nearest neighbour (KNN) classifiers, logistic regression (LR), discriminant analysis (DA), naive Bayesian (NB) classifiers, artificial neural networks (ANNs), and classification trees (CTs).[7] The study found no substantial difference between the six in terms of error rates according to the training data. However,in the validation set, artificial neural networks has the best performance among six data mining measure, it has the relatively low error rate and highest area ratio (shown in Figure 1).

Table 1
Classification accuracy

Method	Error rate		Area ratio	
	Training	Validation	Training	Validation
K-nearest neighbor	0.18	0.16	0.68	0.45
Logistic regression	0.20	0.18	0.41	0.44
Discriminant analysis	0.29	0.26	0.40	0.43
Naïve Bayesian	0.21	0.21	0.47	0.53
Neural networks	0.19	0.17	0.55	0.54
Classification trees	0.18	0.17	0.48	0.536

Figure 1. Result from Yeh, I-Cheng and Lien, Che-hui's study [7]

The estimated prediction of real probability of default computed by Sorting Smoothing Method (SSM) in Yeh, I-Cheng and Lien, Che-hui's study.

$$P_i = \frac{Y_{i-n} + Y_{i-n+1} + \cdots + Y_{i-1} + Y_i + Y_{i+1} + \cdots + Y_{i+n-1} + Y_{i+n}}{2n+1}$$

Figure 2. Equation from Yeh, I-Cheng and Lien, Che-hui's study [7]

Figure 3 shown the summary of the linear regression between real probability and predictive probability of default from the six data mining techniques. In the figure, we can see that artificial neural networks has the highest Regression ($R^2 = 0.965$). We may use the results or findings from Yeh and Lien's paper to support our predictions.

Summary of linear regression between real probability and predictive probability of default

Method	Regression Coefficient	Regression Intercept	Regression R^2
K-nearest neighbor	0.770	0.0522	0.876
Logistic regression	1.233	−0.0523	0.794
Discriminant Analysis	0.837	−0.1530	0.659
Naïve Bayesian	0.502	0.0901	0.899
Neural networks	0.998	0.0145	0.965
Classification trees	1.111	−0.0276	0.278

Figure 3. Result from Yeh, I-Cheng and Lien, Che-hui's study [7]

3. Proposed Method

3.1. Problem Description

This project is going to predict the credit performance of the clients based on the information such as the age, sex, martial status, and past credit history such as bill amount and previous payment statement by month. It is a complicated and multi-variate problem which has various parameter that will impact the final evaluation and prediction of the models.

We explore this task using data from credit companies in Taiwan which contains the information of the clients for the past six month/ In order to make prediction, we fit different kinds of machine learning models to predict the performance of the client.

We use different strategies and methods to solve this problem. First, we use a linear regression with all features and parameters of the baseline model. From that, we can try different machine learning models and methods to train the data such as Random Forest, Decision tree, XGBoost, Gradient Boosting. These models are all being trained on the same training sets which is comparable to each other about accuracy and performance..

3.2. Data Preprocess

Our dataset consisted of 30,000 credit card clients from Taiwan with information of their age, sex, education, payment history and bill statements. A preview of the dataset is shown below:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2
0	1	20000.0	2	2	1	24	2	2	-1	-1	...	0.0	0.0	0.0	0.0	689.0
1	2	120000.0	2	2	2	26	-1	2	0	0	...	3272.0	3455.0	3261.0	0.0	1000.0
2	3	90000.0	2	2	2	34	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	1500.0
3	4	50000.0	2	2	1	37	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	2019.0
4	5	50000.0	1	2	1	57	-1	0	-1	0	...	20940.0	19146.0	19131.0	2000.0	36681.0

Figure 4. Preview of Our Data set

Since there were no outliers and no missing values, and the categorical output variable is already labeled in binary form in the original dataset, we did not manipulate the original dataset.

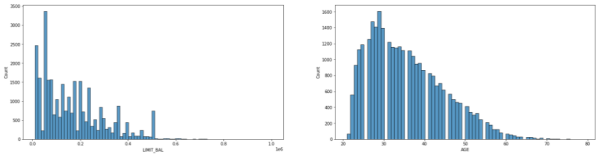

Figure 5. Plot of Continues Variables in Our Data Set

After some initial exploratory data analysis, we constructed a histogram of continues variables in our data set, which is shown in 5 . We observed that we have more number of clients having limiting balance between 0 to 200000 currency, and we have more number of clients from age bracket of 20 to 40, which indicates that the clients are from mostly young to mid aged groups.

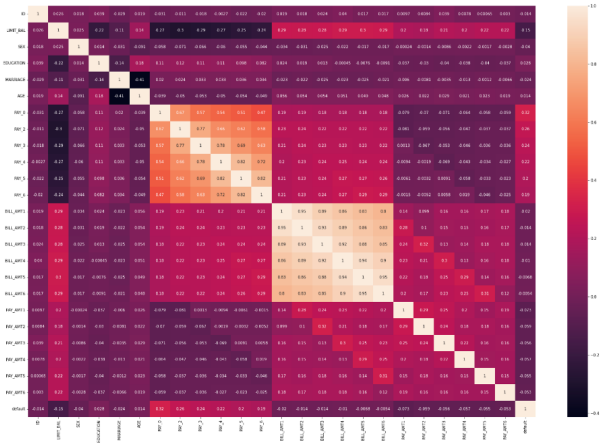

Figure 6. Plot of Continues Variables in Our Data Set

In order to have a better understanding of our data set, we also constructed a correlation matrix, which is shown in figure 6. We can see that next month default prediction is

dependent on repayment status of past six months of all the features given to us. But there is multicollinearity between the Repayment Status features. This could potentially increase error of our models, and we would discuss this later in our experiment section.

3.3. XGBoost

After processing the data, we start to implement common machine learning models to train our data. Firstly, we choose to implement XGBoost algorithm which is an newly implementation of Gradient Boosted Decision Trees algorithm. Just like the name it stands for, extreme gradient boosting, XGBoost uses several tricks and approximations to make the sequential training process faster.

XGBoost is an algorithm that performs a cycle that will continuously generates new models and then combine all of them into an ensemble. In order to find the best model, we calculate the errors every time we build a new model and use the model to predict the errors. New models will fix the errors in the previous models and then be added to the ensemble models. Therefore , the ensemble models will be improved throughout the cycle and finally we will have the best model.[5]

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

Real value (label) known from the training data-set

Can be seen as f(x + Δx) where x = $\hat{y_i}^{(t-1)}$

Figure 7. XGBoost objective function analysis.

Furthermore, we also use the Hyperopt library for Bayesian hyperparameter optimization. During the tuning, we try to perform different kinds of hyperparameter and extend the range of the hyperparamter to find out the best value. Random search is one of the algorithms implemented for Hyper-Parameter Optimization. In random search, we keep iterating the searching process to find a better position and every iteration is dependent on the prior iteration.

3.4. Random Forest

The second algorithm we used to train our data set is random forest. Random forest is a supervised learning algorithm, it constructs multiple decision trees on different samples, and it takes the majority vote for classification and merges results together to compute a more accurate prediction.

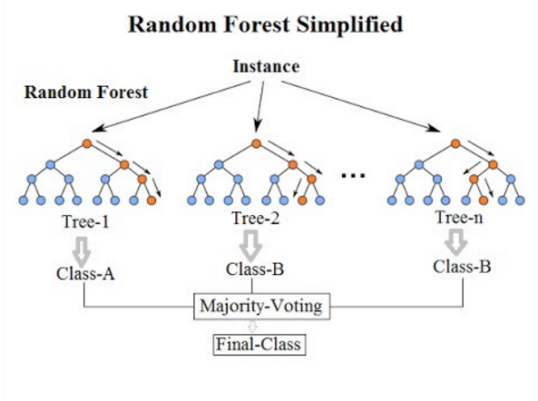

Figure 8. Random Forest diagram [1]

3.4.1 Decision tree

Decision tree learning is a basic predictive model for statistic and machine learning, and it is also the basic logic behind random forest measure. The logic behind the decision tree is using basic decision rules derived from data features to predict the value of a target variable [4].Decision tree model is popularly used not only because it is easy for people to understand and to interpret but also can handle both numerical and categorical data. However, decision tree model also has limitations: it is very unstable, a small change in training set would lead to very different final prediction; it has low bias but very high variance, so it is easily to get over-fitting, which means decision tree could generate too complicated trees that fail to generalize the data well[4]. So in this project, instead of using decision tree, we chose random forest to train our data.

3.4.2 Bagging

Bagging is a technique aimed for increasing the accuracy and stability of machine learning algorithms. It can lower variance but also help to avoid over-fitting [1].To generate varied samples, bagging employs a bootstrapping sampling approach to create diverse samples.Set training data $X = x_1, x_2, x_3, \ldots$ with their result $Y = y_1, y_2, y_3, \ldots$, bagging repeatedly and randomly selected samples in the training set with replacement, and fit trees in these samples. In this way, bagging lower the variance without increase the bias [1].

3.4.3 From Bagging to Random Forest

Except the original bagging algorithm, Random forest algorithms also consists another type of bagging:feature bagging, which applied a modified tree learning method to randomly select a subset of features.In this way,if several features have strong influence on the prediction, these features will be selected many more times in trees [1].

3.5. Gradient Boosting

Different from using a random forest algorithm, we fit the data using a gradient Boosting model in order to compare with our baseline model. Gradient boosting is a unique machine learning technique which is often used in classification and regression models.

This method combine multiple weak leaner such as decision trees into a strong leaner. To achieve this, gradient boosting model optimize a differentiable loss function.

This model can be used to predict as an ensemble form of weak prediction models like decision trees. While a decision tree is a weak learner, the result of gradient boosting model is often called gradient-boosted trees.

3.5.1 Learning Rate

Learning rate, which is often refereed as , shows the speed of the model learning. As each tree modifies the general model, the magnitude of the modification is defined and controlled by learning rate.[2]

Learning rate is an indicator that controls the amount of contribution from each model to the general ensemble prediction results. The smaller the rates, the more decision trees that needs in the ensemble. Thus, we aim to explore different learning rates using a log scale from 0.0001 to 1.0.

3.6. Logistic Regression

The last model we chose is Logistic regression. We intended to let Logistic Regression be our baseline model, because it is simple, low- cost to train, and efficient. Logistic regression is a classification model that models the probability of a discrete outcome. It is simple and efficient method for binary classification problems. It primarily uses a logistic function (shown in 9) to model the output variable instead of fitting a hyperplane, and it does not require its input and output variables to have a linear relationship. The main difference between logistic regression and linear

$$P(t) = \frac{1}{1 + e^{-t}}$$

Figure 9. Logistic Function

regression is that logistic regression does not try to fit a hyperplane, but instead utilizes the logistic function to make the output of linear equation between 0 and 1.

4. Experiments

4.1. Dataset

Our data set contains data of detailed default payments, demographic information of the client, general credit data, history of payment, and bill statements of credit card clients from April 2005 to September 2005 in Taiwan. The above figure shows an overview of what our data looks like:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AM
0	1	20000.0	2	2	1	24	2	2	-1	-1	...	0.0	0.0	0.0	0.0	689.0	
1	2	120000.0	2	2	2	26	-1	2	0	0	...	3272.0	3455.0	3261.0	0.0	1000.0	100
2	3	90000.0	2	2	2	34	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	1500.0	100
3	4	50000.0	2	2	1	37	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	2019.0	120
4	5	50000.0	1	2	1	57	-1	0	-1	0	...	20940.0	19146.0	19131.0	2000.0	36681.0	1000

5 rows × 25 columns

Figure 10. A General Overview of the Dataset

The data set consists information such as personal information of the client such as sex, marriage, educational level, and the credit histories of the clients like repayment status, amount of bill statement, and amount of previous payment by month. The original data can be found by this link .

4.2. XGBoost

We implent two different models using XGBoost. The first model uses all the features as predictors and the second one only uses features that has absolute correlation greater than 0.1.

As for the first model, We use all the 23 features excluding ID as our predictors and perform XGBoost algorithm. Then, hyperopt library is implemented for Bayesian hyperparameter optimization. In order to find the best value for each hyperparameter, we used hyperopt optimization algorithms to extend the range of hyperpamaters including *n estimators, learning rate, lambda, alpha, gamma, reg alpha, reg lambda, colsample bytree, min child weight.*

Hyperparameter	Range
n estimators	hp.choice[30, 50, 100, 300])
learning rate	hp.choice[0.01]
lambda	hp.loguniform[1e-8, 1.0]
alpha	hp.loguniform[1e-8, 1.0]
gamma	hp.uniform[1,9]
reg alpha	hp.quniform[40,180,1]
reg lambda	hp.uniform[0,1]
colsample bytree	hp.uniform[0.5,1]
min child weight	hp.quniform[0, 10, 1]

Table 1. Range of hyperparameters

In the model using only uses features that has absolute correlation greater than 0.1, we first use correlation function to calculate its correlation with default payment and then select the features that has absolute values greater than

0.1. In the same way, we implement the XGBoost algorithm as well as Bayesian hyperparameter optimization to find the best value for hyperparameters. The range of the hyperparameters is the same as former model.

4.3. Random Forest

In random forest model, firstly we want to see the correlation between "default.payment.next.month" and other features. So we draw a heatmap (shown in figure 7) to see the correlation result. The numerical result listed in figure 8. We can see that there are positive and negative correlation with "default.payment.next.month", positive correlation means the feature and the result move to the same direction, that is when the feature value increase the correlated value (default payment next moth result) increase, vice versa. And we can see that Pay0 has the largest correlation (0.325) with the result, and others have less significant impact on result. Therefore, we choose all features together as our predictors to train our data.

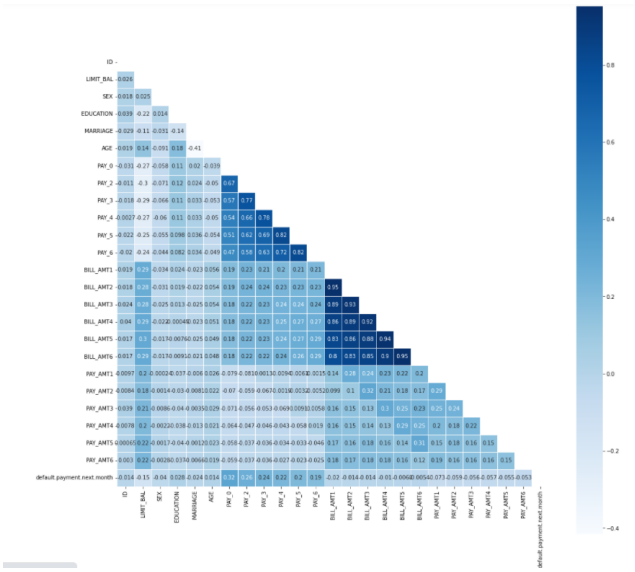

Figure 11. Correlation plot

Then, we first set X as the predictor set which only drop the "default.payment.next.month" feature, set y as our prediction set which includes all "default.payment.next.month" results (1 for has default payment next month and 0 for no default payment next month). Then we split the data set into two subsets: train set and test set. And we set 20% of the data set as our test set.

4.4. Gradient Boosting

In this model, we use different values of the learning rate to test the prediction model modification from 0.0001 to 1.0. In order to compare the performance of each

default.payment.next.month	1.000000
PAY_0	0.324794
PAY_2	0.263551
PAY_3	0.235253
PAY_4	0.216614
PAY_5	0.204149
PAY_6	0.186866
EDUCATION	0.028006
AGE	0.013890
BILL_AMT6	-0.005372
BILL_AMT5	-0.006760
BILL_AMT4	-0.010156
ID	-0.013952
BILL_AMT3	-0.014076
BILL_AMT2	-0.014193
BILL_AMT1	-0.019644
MARRIAGE	-0.024339
SEX	-0.039961
PAY_AMT6	-0.053183
PAY_AMT5	-0.055124
PAY_AMT3	-0.056250
PAY_AMT4	-0.056827
PAY_AMT2	-0.058579
PAY_AMT1	-0.072929
LIMIT_BAL	-0.153520

Name: default.payment.next.month, dtype: float64

Figure 12. Correlation numerical result

configured learning rate, we use gradient boosting model to explore and compares the results of each values and their influences on the overall performance of the models.

The set of learning rate we used to explore is [0.0001, 0.001, 0.01, 0.1, 1.0]. For model building, we use the class RepeatedStratifiedKFold from scikit-learn with n splits of 10 and n repeat of 3.

The task here is classify a individual's credit background prediction with all of our parameters as predictors since we want to evaluate the clients' information from a general angle.

We observe the mean accuracy for each configured learning rate throughout the times and generate a comparison between them about the performances and accuracy.

By GradientBoostingClassifier, we used the learning rate that has the best performance which is 0.1 with max depth of 5.

4.5. Logistic Regression

In the logistic regression model, we implemented two models. The first model is trained with all features and is considered as our baseline model. The second model is then trained without the features with high multicollinearity based on the correlation plots we drew in previous sections. The reason we wanted to revome the variables is that multicollinearity refers to unusually high correlations between two variables. As it increases, the standard error of the model also increases, and model accuracy is compromised.

For both models we first dropped the ID feature, then we tested different random state numbers and finally chose random state = 10 since it led to the best results. We also split the training and testing data to 80 percent and 20 percent. Using scikit-Learn, we trained each model respectively and obtained their accuracy scores.

4.6. Software

The models running in our project were carried out using Jupyter Notebooks running Python 3.8 or later. We mainly used libraries include pandas and numpy for data processing, matplotlib for plotting graphics, and scikit-learn and mlxtend for building machine learning models.

5. Results and Discussion

5.1. XGBoost

For the first model using all the features as our predictors, we achieve the test accuracy of 81.322%. Then, by setting the range for hyperparameters and implementing Bayesian hyperparameter optimization, we find that the model achieves its best accuracy 81.956% with the value of hyperparameters in Table 2.

Hyperparameter	Best Value
n estimators	50
learning rate	0.01
lambda	1.00
alpha	1.96
gamma	1.78
reg alpha	170.0
reg lambda	0.25
colsample bytree	0.90
min child weight	0.0

Table 2. Best Value for XGBoost hyperparamters in the first model

For the second model that using only selected features as our predictors, the test accuracy of the model is 81.378%. Again, using the same way by Hyperopt hyperparameters tuning, the test accuracy increases to 82.089%. The best value for the hyperparameters are shown in Table 3.

Hyperparameter	Best Value
n estimators	30
learning rate	0.01
lambda	1.38
alpha	1.88
gamma	3.39
reg alpha	44.0
reg lambda	0.88
colsample bytree	0.66
min child weight	7.0

Table 3. Best Value for XGBoost hyperparamters in the second model

5.2. Random Forest

In random forest model, we select samples from training set using bootstrap. And,we used all parameters as our predictors, set $n\_estimator = 100$, $random\_state = 2$,and we get 82% accuracy on test data set which means we get 82% correct predictions over total predictions using random forest model.

5.3. Gradient Boosting

5.3.1 Learning Rate Comparison

After running and evaluating the performance of different learning rate from 0.0001 to 1.0, we can see that the larger the learning rate, the better the performance is on the data set. From this result, we can expect that we can advance and improve the performance on ensembles of smaller learning rates by adding more trees into it.

Also, we can improve the speed of model's learning process by using fewer trees with a larger learning rate.

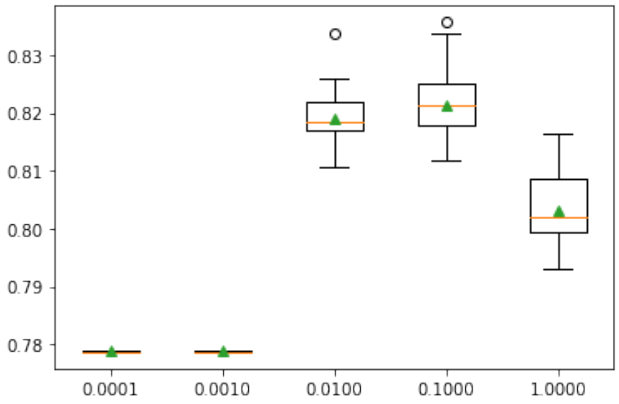

Figure 13. A Comparison of Learning Rate Performance

5.3.2 Gradient boosting results

For the original model using all the features as our predictors and learning rate of 0.0001, we achieve the test accuracy of 77.9%. Then, by setting the learning rate of 0.1 which has the best performance overall, we find that the model achieves its best accuracy 82%.

Also, because we have separate the data, we utilize Mean absolute percentage error (MAPE) function as an error metric model to further evaluate the algorithm.

From the results, the mean absolute percentage error (MAPE) of this model is 10.472%, which means that 10.472% is the average difference between the predicted and actual value.[8]

5.4. Logistic Regression

Our first logistic regression model trained with all features yielded an accuracy of 78% (shown in 14). Then the second model which dropped all the unimportant features

and features with high multicollinearity achieved an accuracy of 79.0% (shown in 15). This shows that the accuracy of our model increased by 1 percent and removing features with high collinearity could positively affect the accuracy of our models.

	precision	recall	f1-score	support
0	1.00	0.78	0.88	5998
1	0.00	0.00	0.00	2
accuracy			0.78	6000
macro avg	0.50	0.39	0.44	6000
weighted avg	1.00	0.78	0.88	6000

Figure 14. Classification report of first LG model

	precision	recall	f1-score	support
0	1.00	0.79	0.88	3000
1	0.00	0.00	0.00	0
accuracy			0.79	3000
macro avg	0.50	0.39	0.44	3000
weighted avg	1.00	0.79	0.88	3000

Figure 15. Classification report of second LG model

6. Conclusions

In this project, we sought to train and fit a machine learning model that can best predict if a credit card client is going to default his/her payment next month based on the client's various features such as sex, education, payment history and bill statements.

After experimenting four different models, we found that the XGBoost model has the best performance overall. It not only has the highest accuracy (82.09%) in the four models, but also has many other advantages such as allowing users to run cross-validation after each iteration and it's faster than gradient boosting.

A potential future development of our research could be training a model that utilizes more personal information such as ethnic group, nationality, and less bank account information such as previous payment history. Because when a credit card company is evaluating the risk of default of a new customer, the company doesn't necessarily have access to the client's bank account information. If a model could accurately predict the risk of default with minimum bank account information, it would be quite interesting and practical.

7. Acknowledgements

We would like to thank UCI Machine Learning for providing the Default of Credit Card Clients Data set and the guidance of Professor Sebastian Raschka in implementing machine learning algorithms and writing final reports.

8. Contributions

Yirun Wang found the Default of Credit Card Clients data set on Kaggle and did feature selection to select the best predictor features. As for models, Yirun Wang coded the XGBoost model and tuned the model using Bayesian hyperparameter optimization. Yirun Wang is also in charge of abstract, XGBoost part and acknowledgements.

Yunqing Xiao is in charge of the introduction, gradient boosting, problem description, and dataset explanation for this report.

Fenghang Yao is in charge of writing the related work, random forest part which includes coding and writing related reports.

Shupeng Tang is in charge of coding of the logistic regression model and plots for data preprocess. As for the report, Shupeng Tang covered the data preprocess, logistic regression and conclusion.

References

[1] Random forest, Nov 2021.

[2] Gaurav. An introduction to gradient boosting decision trees. Jun 2021.

[3] F. R. B. of St. Louis. Delinquency rate on credit card loans, all commercial banks, Oct 2021.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] C. Tian and G. Carlos. Xgboost: A scalable tree boosting system. *Machine Learning*, 978(1):785–794, 2016.

[6] I.-C. Yeh. default of credit card clients data set.

[7] I.-C. Yeh and C.-h. Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients, Jan 2008.

[8] Zach. mean absolute percentage error (mape).