Consumer Credit Card Default Risk Analysis

Lina Gao

May 10th 2021

Introduction

Credit defaulter prediction is crucial for credit card issue companies to mitigate the impact of unexpected financial damage. Rapid development in statistical machine learning provides valuable tools to assess and predict credit default risk. This project will explore multiple machine learning techniques to predict consumer credit default risk using the data set from UCI public data repository. It includes 30000 observations with 25 features comprised of the clients' socioeconomic data and transaction data in the past six months. We will evaluate six different supervised classification algorithms and identify critical factors for their prediction. The workflow is described in figure 1.

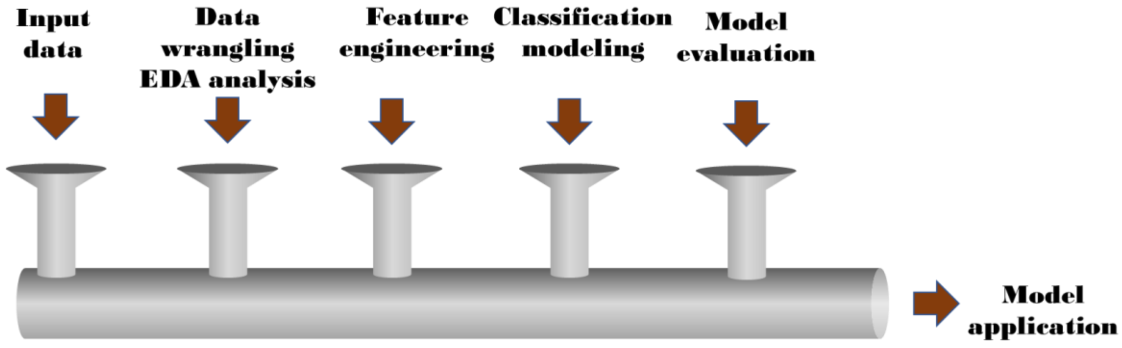


Fig. 1 The workflow of the credit default prediction

Data exploration

1. Bird's-eye view of the data

Data source: UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA

These data were collected in Taiwan from April to September 2005. Their descriptions are listed in table 1.

Table 1. Feature descriptions for credit card default data

Variables	Description
ID	ID of each client
LIMIT_BAL	Amount of given credit in NT dollars (includes individual and family/supplementary credit)
SEX	Gender (1=male, 2=female)
EDUCATION	(1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)

PAY_0, PAY_2, …, PAY_6	Repayment status in September, August, July, June, May, April (-2=no consumption, -1=pay duly, 0=the use of revolving credit, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above)
BILL_AMT1, BILL_AMT2, …, BILL_AMT6	Amount of bill statement in September, August, July, June, May, April, 2005 (NT dollar)
PAY_AMT1, PAY_AMT2, …, PAY_AMT6	Amount of previous payment in September, August, July, June, May, April 2005 (NT dollar)

This data set is 100% numerical without any missing values. It comprises continuous variables, ordinal categorical variables, and nominal categorical variables. LIMIT_BAL, AGE, BILL_AMT1 to BILL_AMT6, and PAY_AMT1 to PAY_AMT6 are continuous variables with the right-skewed distribution. The data is not balanced considering SEX, EDUCATION, MARRIAGE, and DEFAULT. Based on Pandas profile results, we can find the high correlations between historical data, for instance, PAY_AMT1~PAY_AMT6, BILL_AMT1~BILL_AMT6. Therefore, we need to do feature engineering to avoid multicollinearity between predictors and convert categorical variables into dummy variables regarding the models we are interested.

2. Worm's-eye view of the data

The primary question in this section is which factors are essential for the credit default prediction. First, we studied the distribution of clients across different EDUCATION, SEX, MARRIAGE, and DEFAULT status. We found that single and married females are the primary clients regardless of the status of the credit default. They contributed ~60% to the total clients. Female clients with a university degree were the dominant subgroup when using EDUCATION and SEX to segment the data. They represent 29% of the clients. The primary contributors in this data are clients with a higher education degree, either a university degree or graduate degree, without consideration of other predictors. They represent ~80% of all clients disregarding the status of default. The most special subgroup is the clients with education level labeled as 'others.' They are the people with the lowest default rate regardless of gender. Only 8.2 percent got defaulted, compared with the average default rate of 28% for all clients.

Using statistics test, we tested the effect of the predictors on DEFAULT in this data. Chi-square test gave us the persistently positive result for SEX, MARRIAGE, and EDUCATION. Female tends to have lower default rate than male. Clients with higher education degrees (university and graduate) are generally less likely to get credit default than clients with a high school degree. More Married clients received credit default than single clients. Checking the effect of DEFAULT on LIMIT_BAL by t-test and Mann-Whitney-U test, we found the default group tended to have significantly lower LIMIT_BAL than the non-default group. This phenomenon is more noticeable when we only check the clients with LIMIT_BAL higher than 740k. Almost none of them received default credit, as shown in figure 2. AGE also has a significant effect on DEFAULT based on t-test results. There are three sets of predictors from the historical transaction in this dataset,

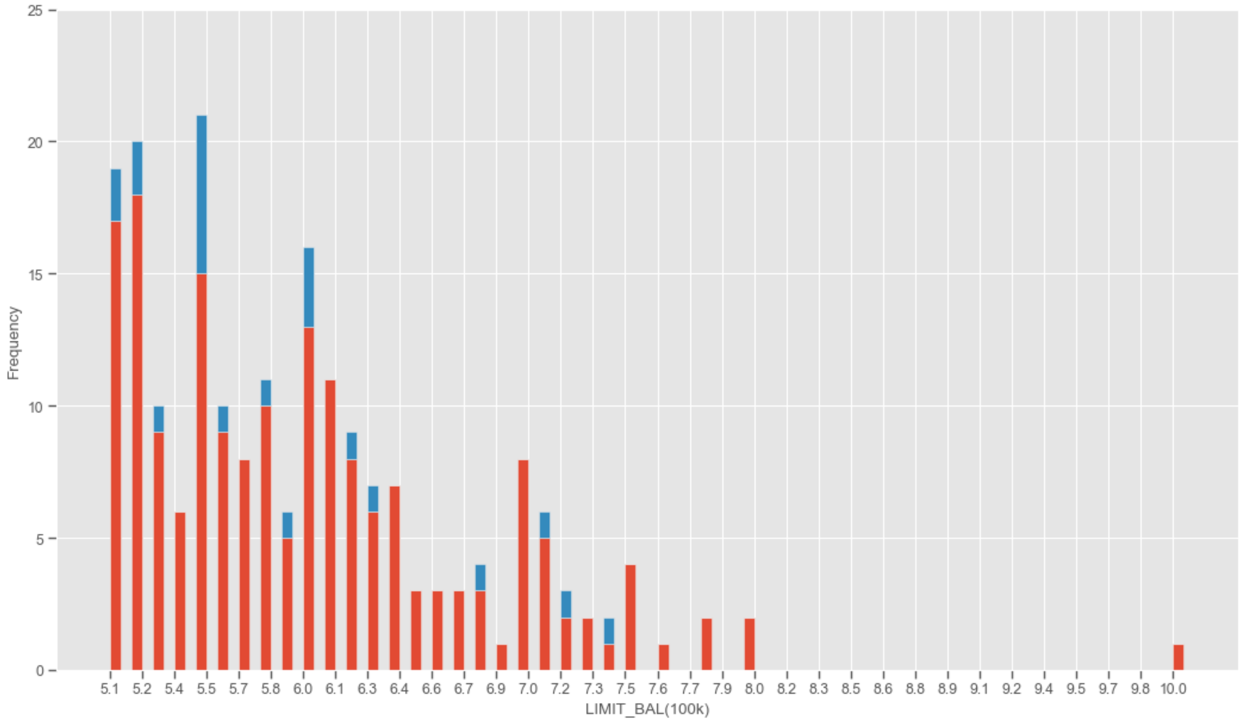


Fig.2 The stacked histogram of LIMIT_BAL colored by DEFAULT

BILL_AMT1 to BILL_AMT6, PAY_AMT1 to PAT_AMT6, and PAY_1 to PAY_6. The significant effect of default status on these features is confirmed by both the Mann-Whitney U test and t-test. Using the probability density plot and the cumulative probability plot, we also found their similarity and differentiations. The clients with credit default have lower values of BILL_AMT and PAY_AMT than the ones without default. But the defaulters tend to have higher PAY_1 to PAY_6 values than the non-defaulters. The similarity of their shape within each set also indicated their high correlations, for instance, BILL_AMT in figure 3.

Moreover, we tested the effect of SEX, MARRIAGE, and EDUCATION on LIMIT_BAL with the Mann-Whitney-U test, Kruskal-Wallis tests, ANOVA, or t-test. We found a significant difference of LIMIT_BAL between male vs. female and single vs. married using both Mann-Whitney U test and t-test. Female clients have a higher median level of LIMIT_BAL than males. The same situation appeared for single clients as married ones. Checking the effect of EDUCATION on LIMIT_BAL by Kruskal-Wallis tests and ANOVA, we observed the significant difference of LIMIT_BAL across different EDUCATION levels. Clients with higher education tend to have more likelihood with higher LIMIT_BAL than the ones with low education.

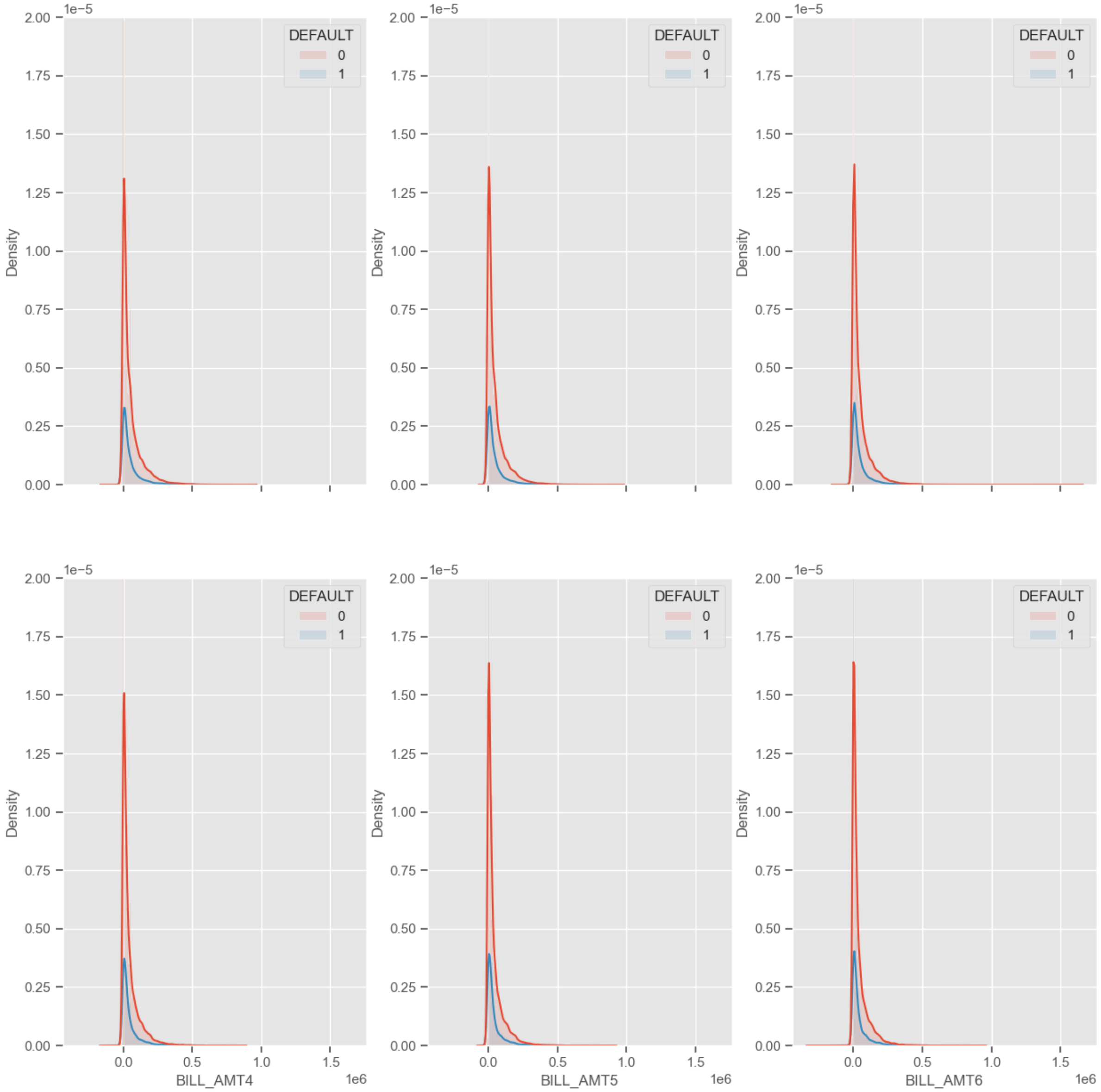


Fig.3 The probability density distribution of BILL_AMT1 to BILL_AMT6 colored by DEFAULT

In summary, all predictors are essential for the prediction of the target variable, DEFAULT. Some predictors are correlated, as indicated in figure 4. We need to do some feature engineering to mitigate the side effect of multicollinearity between these predictors based on the requirements of the modeling assumption and convert categorical variables into dummy variables as well.

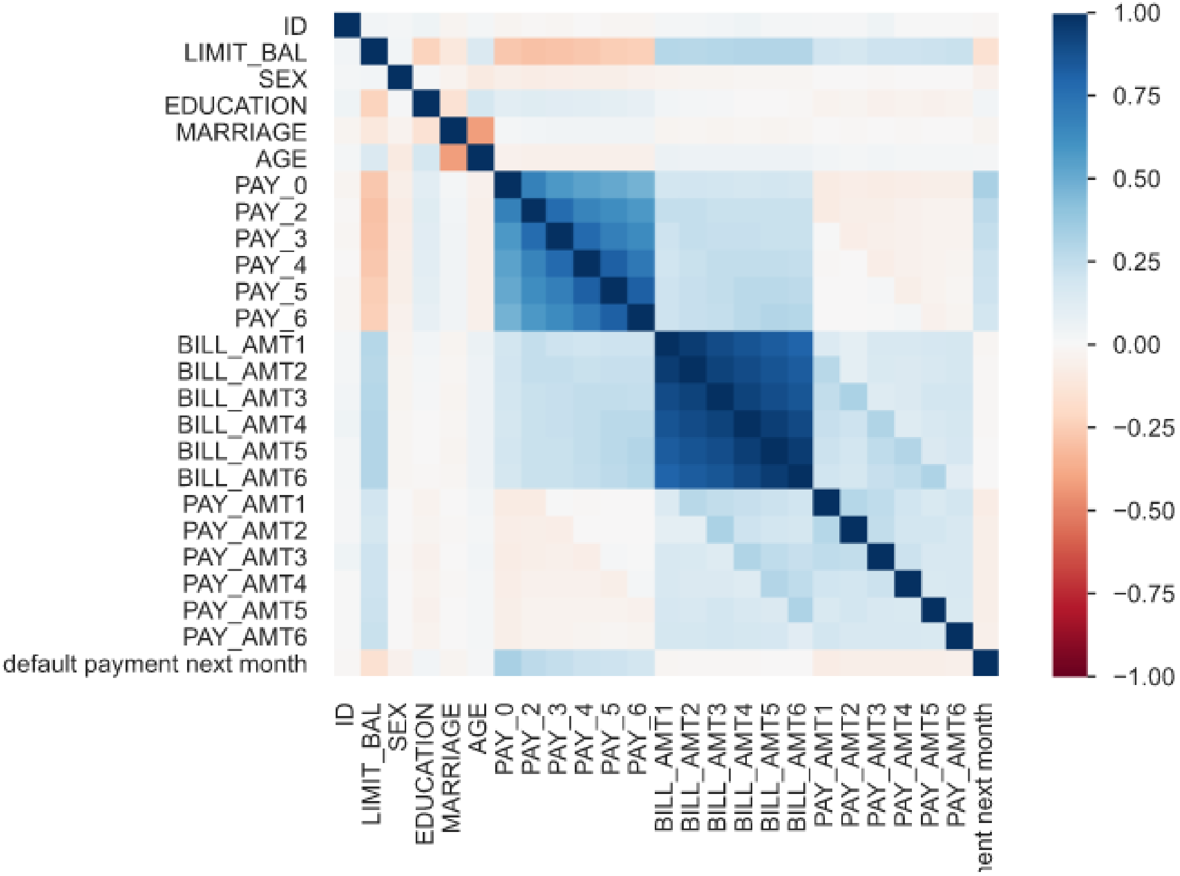


Fig. 4 The correlation between the predictors in the original data set

Feature engineering and data pro-processing

In this data set, we found account-level data, including historical statement data and demographic data, not the credit bureau data. Because of the multicollinearity between the historical data, we decided to construct several new features convenient for our data analysis process. They are

1. $pay\_sum = \sum_{i=1}^{6} PAY\_i$
   It is the accumulated payment status. The lower, the better. (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above).
2. $mean\_utilization\_ratio = \sum_{i=1}^{6} BILL_{AMTi} /(6 \times LIMIT\_BAL)$
   It is the average utilization ratio of the clients. It can help us isolate the effect of LIMIT_BAL and check the average percentage of clients' expenses over credit limit in the past six months.
3. $6\_month\_loss\_given\_default = \sum_{i=1}^{6} BILL\_AMTi - \sum_{i=1}^{6} PAY\_AMTi$

This feature evaluates the total loss of the credit card company if the clients are given credit default after six months

4. $mean\_payment\_ratio = \sum_{i=1}^{6} PAY\_AMTi \ /(6 \times LIMIT\_BAL)$
   It can help us evaluate the average ratio of payment over the credit limit of the clients
5. $bill\_trend = (\sum_{i=1}^{3} BILL\_AMTi - \sum_{i=4}^{6} BILL\_AMTi \ )/(3 \times LIMIT\_BAL)$
   We can use this feature to monitor the trend of client's expenses. Was it increased, decreased, or did no show any trend at all over the past six months?
6. $pay\_trend = (\sum_{i=1}^{3} PAY\_AMTi - \sum_{i=4}^{6} PAY\_AMTi \ )/(3 \times LIMIT\_BAL)$
   Just like bill_trend, we constructed pay_trend to assess the clients' payment status.
7. $bill\_sum = \sum_{i=1}^{6} BILL\_AMTi$
8. $payment\_sum = \sum_{i=1}^{6} PAY\_AMTi$

The above eight features with the original features, EDUCATION, MARRIAGE, AGE, LIMIT_BAL, and SEX, are included in the data set, df_sum, which will compare with the original data set for model development. Unlike the original data set, there was no obvious collinearity in the continuous features of this data set, as shown in figure 5.

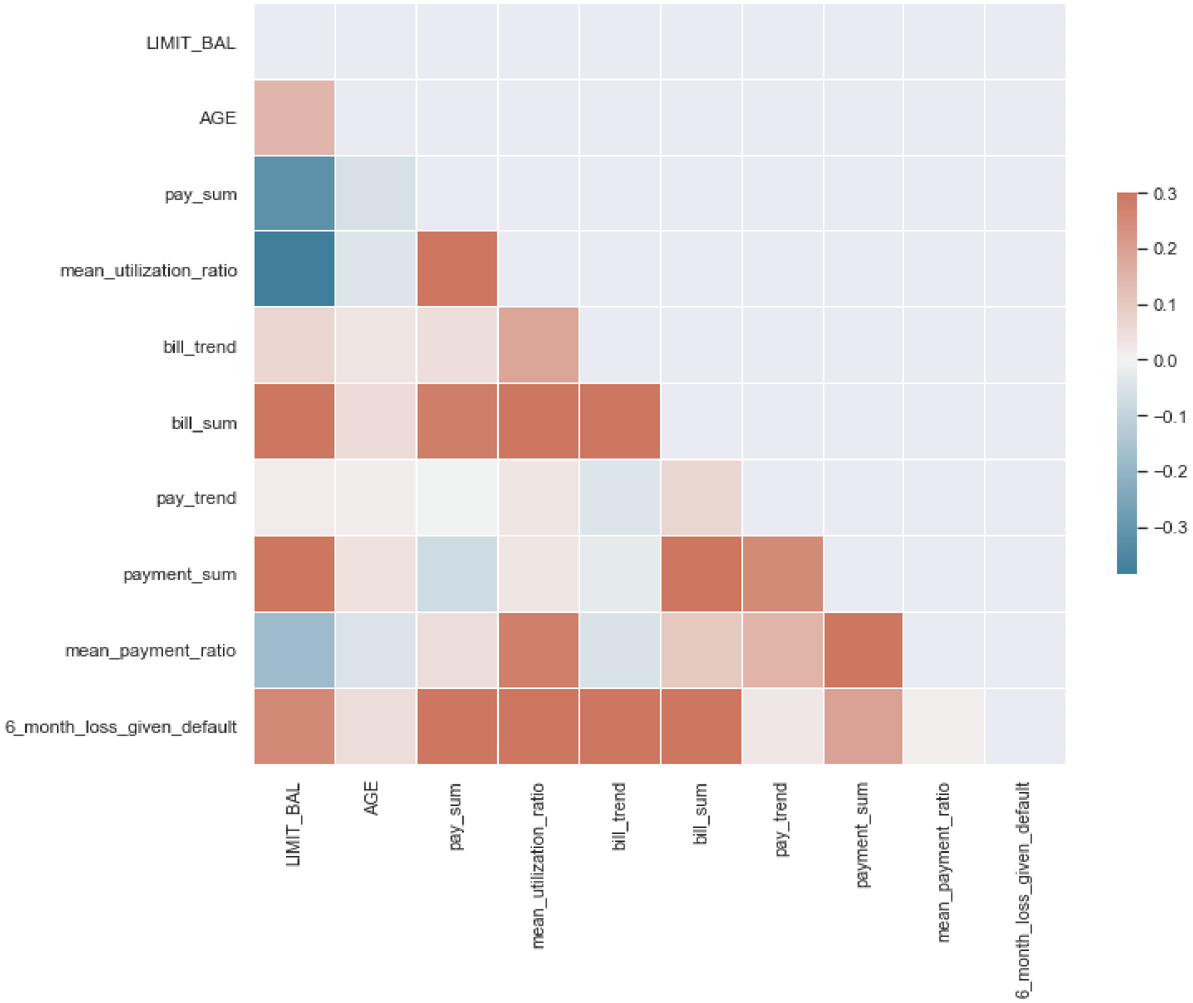


Fig. 5 The correlation of the constructed features, AGE and LIMIT_BAL in df_sum

Besides, we converted MARRIAGE, SEX, and EDUCATION into dummy variables. To avoid collinearity, we chose to drop the redundant columns during the transformation. As for AGE and LIMIT_BAL, we decided to keep them as they were. Although there are ~80% of clients without default credit card vs. ~20 default in our data, it is not very severe imbalance data. We didn’t apply oversample or undersample techniques. Considering the wide variety of feature scales in this project, we used a standard scaler to rescale them before modeling.

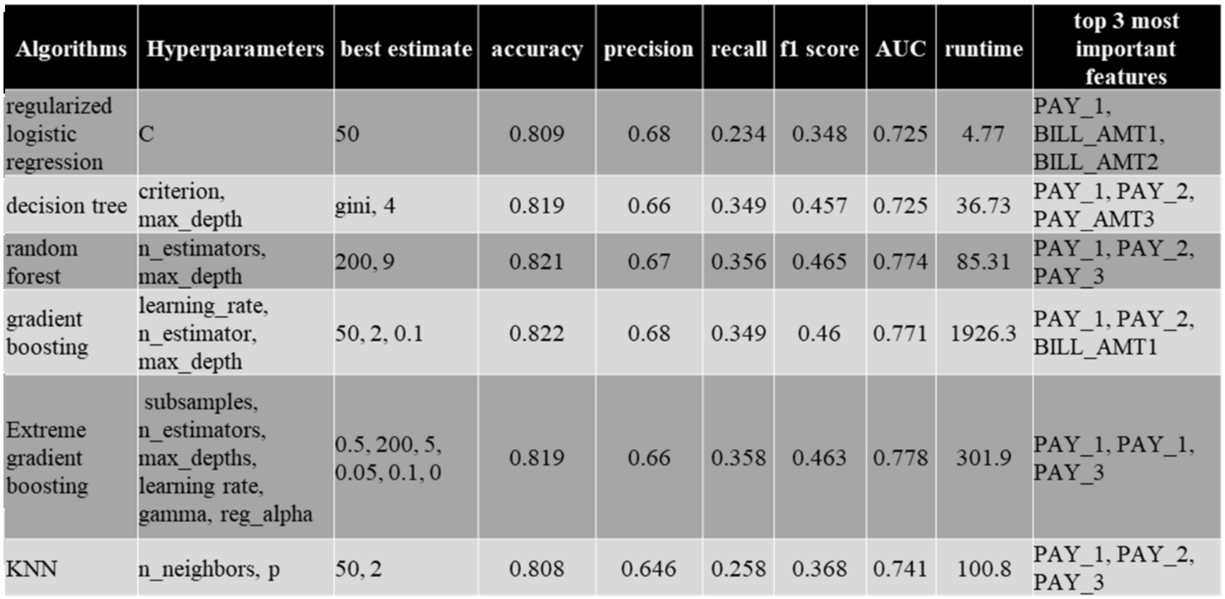
Model development:

I. modeling using the original data

We applied six classification algorithms for the prediction of credit default using the complete original data. They are regularized logistic regression (RLR), decision tree (DT), random forest (RF), gradient boosting (GB), extreme gradient boosting (EGB), and K-nearest neighbor (KNN). Binary logistic regression is the simplest traditional classification technique. But it cannot perform very well when there is collinearity between predictors. It is also vulnerable to overfitting. Thus, we chose RLR. The hyperparameter we focus on is C, which is the inverse of regularization strength. Thus, small values of C mean strong regularization. DT is another most popular and widely used machine learning algorithm because of its special strength in interpretability and less requirement for data pre-processing. In this project, we chose criterion and max_depth to optimize. Criterion is the function to measure the quality of a split. In Scikit_learn, there are two options offered, "gini" and "entropy." Both are used to evaluate the information gain. Max depth is the longest length of the path from the root of the tree to a leaf. KNN is a non-parametric method used for classification. We chose two hyperparameters to tune in this project, n_neighbors, and p. When p = 1, we are using Manhattan distance (l1). When p =2, we switch to Euclidean distance (l2). In general practice, people choose the number of neighbors by calculating sqrt(N), where N stands for the number of samples in the training set.

As an ensemble algorithm, RF is regarded as the combination of decision tree and bagging. Therefore, it inherits the advantages of DT but overcomes the overfitting issues in DT by bagging and can help us get more accurate results. The hyperparameters we chose to tune are n_estimators and max_depth. n_estimators is the number of trees in the RF model. Gradient boosting is a combination of DT and boosting. Different from bagging, a parallel training process, boosting is a sequential training procedure. Boosting can provide a weighted average of their estimates. It also means more accuracy but time-consuming. Three critical hyperparameters need our attention, learning_rate, n_estimator, and max_depth. People created EGB to enhance the speed of traditional gradient boosting. This algorithm applies a unique split-finding algorithm to train the tree. It also combines with regularization to reduce overfitting. Thus, it is expected to be a more rapid and accurate version of gradient boosting. The trade-off is we need to tune more hyperparameters for modeling, subsample, n_estimators, max_depth, learning_rate, gamma and reg_alpha. Gamma is the minimum loss reduction required to make a further partition on the node of the tree. A large gamma means a more conservative algorithm. The subsample is the sample ratio of the training instances during the random sampling step in every boosting iteration. We chose random search with cross validation to tune relevant hyperparameters in these algorithms considering the computation speed. The corresponding results and their prediction evaluation were listed in table 2.

Table 2. Hyperparameter optimization and modelling results



Algorithms	Hyperparameters	best estimate	accuracy	precision	recall	f1 score	AUC	runtime	top 3 most important features
regularized logistic regression	C	50	0.809	0.68	0.234	0.348	0.725	4.77	PAY_1, BILL_AMT1, BILL_AMT2
decision tree	criterion, max_depth	gini, 4	0.819	0.66	0.349	0.457	0.725	36.73	PAY_1, PAY_2, PAY_AMT3
random forest	n_estimators, max_depth	200, 9	0.821	0.67	0.356	0.465	0.774	85.31	PAY_1, PAY_2, PAY_3
gradient boosting	learning_rate, n_estimator, max_depth	50, 2, 0.1	0.822	0.68	0.349	0.46	0.771	1926.3	PAY_1, PAY_2, BILL_AMT1
Extreme gradient boosting	subsamples, n_estimators, max_depths, learning rate, gamma, reg_alpha	0.5, 200, 5, 0.05, 0.1, 0	0.819	0.66	0.358	0.463	0.778	301.9	PAY_1, PAY_1, PAY_3
KNN	n_neighbors, p	50, 2	0.808	0.646	0.258	0.368	0.741	100.8	PAY_1, PAY_2, PAY_3

Because we didn’t apply any sampling technique to balance our data before modeling, accuracy was not the best metrics for model evaluation. We need consider all those metrics, especially AUC and f1 score. Since f1 score is calculated by the harmonic mean of precision and recall (2×precision×recall / (precision + recall)), it basically combines precision and recall into one metric. AUC is the area under ROC curve. We can use it to balance the trade off between precision and recall. A good AUC score means the model is developed with good recall without sacrificing the precision very much. Thus, AUC and f1 score are the primary index for us to determine the best model. Observing the ROC/AUC curves of all models in Fig.6 and the evaluation results in table 2, we can identify EGB as the best choice for us. Its' run time is less than the traditional gradient boosting, and its' AUC score is the highest among all models. Its f1 score is almost the same as random forest. The importance of features in the original data set, PAY_1, and PAY_2, are consistently listed as the most crucial regardless which algorithms we applied.

II. modeling using the constructed features

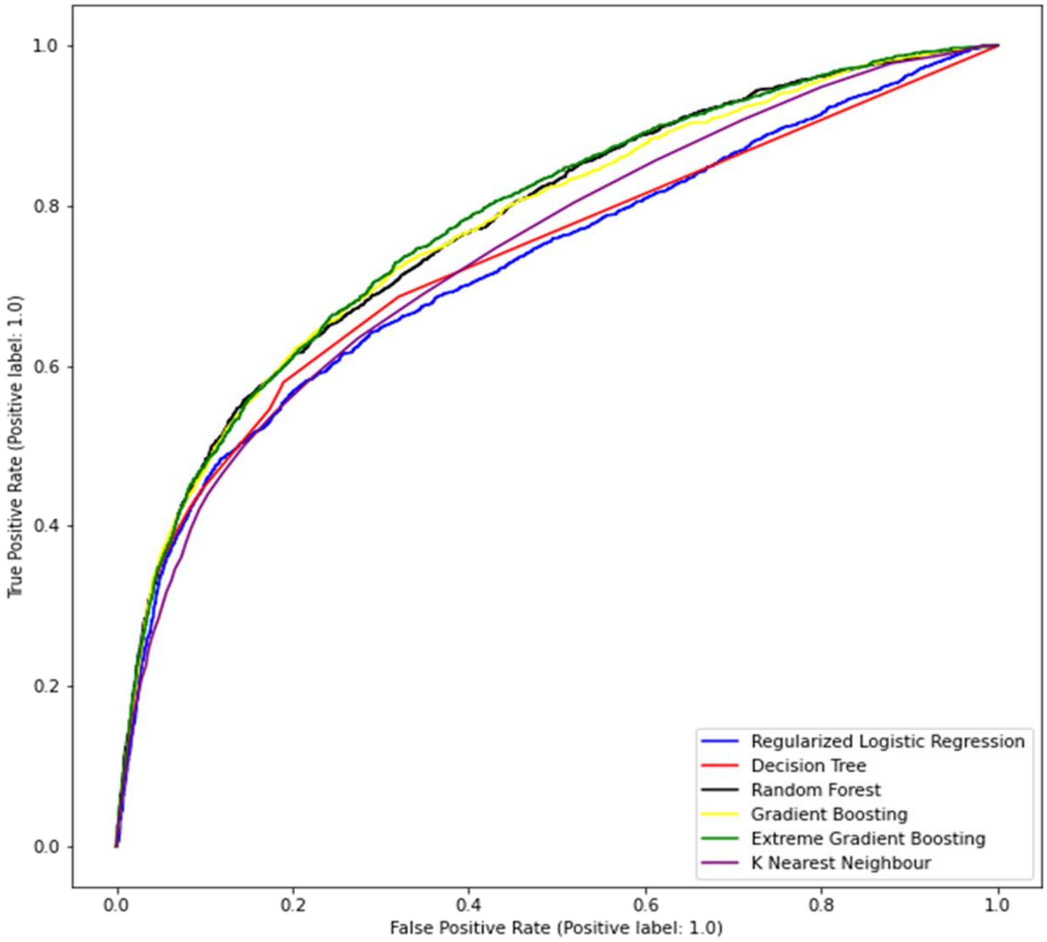


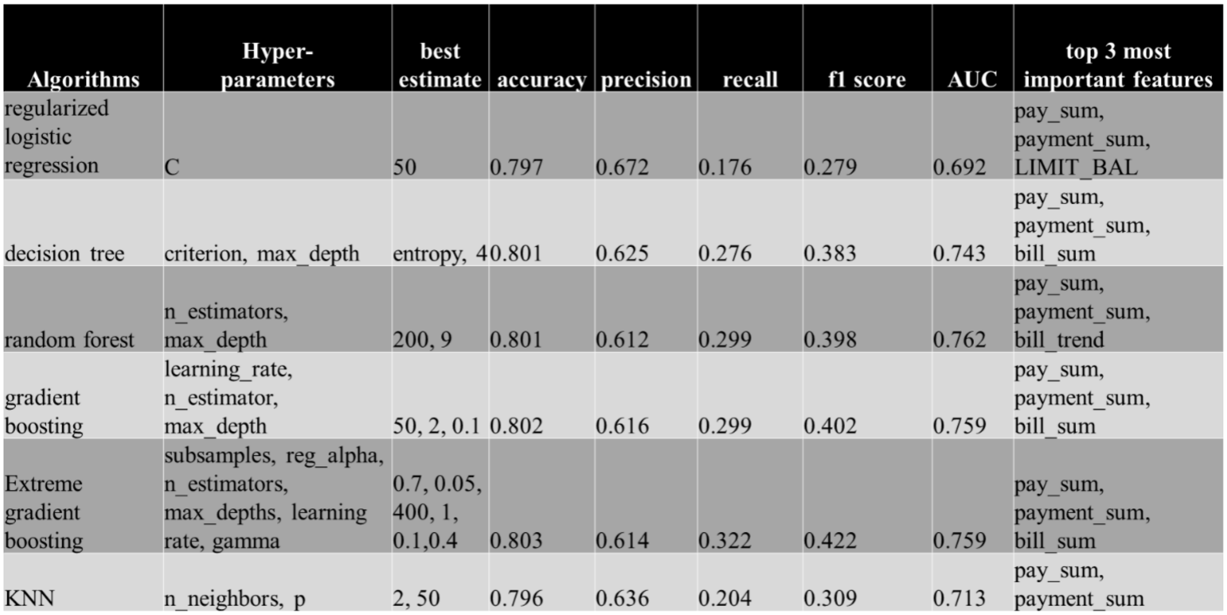
Fig. 6 ROC/AUC curves of all studied algorithms using the original data

We evaluated those six classification algorithms using the engineered features in the same way as the original data. The results were in table 3. Comparing Tables 2 and 3, we can see XGB is persistently the best performer. Using the reduced dimension data set, df_sum, we can get comparable results as the original data set because both their AUC and f1 score values are similar. For instance, the AUC values are 0.778 and 0.463 when using the original data set and EGB algorithm. They turned into 0.759 and 0.422 using the same algorithm with the reduced data set. Another insight we obtained from these models is that pay_sum is often regarded as the top critical feature in the data regardless of socioeconomic status.

Conclusion:

1. Regardless of the socioeconomic status of the clients, payment status is the most critical feature for credit default prediction. It is described as PAY_1 to PAY_6 in the original data or pay_sum in the reduced data set.
2. Features related to a socioeconomic status like age, marriage, and education significantly affect the default in credit risk assessment.
3. XGB is the most attractive algorithm for predicting credit default risk compared with RLR, DT, RF, GB, and KNN.

Table 3. Hyperparameter optimization and modelling results using the engineered features



Algorithms	Hyper-parameters	best estimate	accuracy	precision	recall	f1 score	AUC	top 3 most important features
regularized logistic regression	C	50	0.797	0.672	0.176	0.279	0.692	pay_sum, payment_sum, LIMIT_BAL
decision tree	criterion, max_depth	entropy, 4	0.801	0.625	0.276	0.383	0.743	pay_sum, payment_sum, bill_sum
random forest	n_estimators, max_depth	200, 9	0.801	0.612	0.299	0.398	0.762	pay_sum, payment_sum, bill_trend
gradient boosting	learning_rate, n_estimator, max_depth	50, 2, 0.1	0.802	0.616	0.299	0.402	0.759	pay_sum, payment_sum, bill_sum
Extreme gradient boosting	subsamples, reg_alpha, n_estimators, max_depths, learning rate, gamma	0.7, 0.05, 400, 1, 0.1,0.4	0.803	0.614	0.322	0.422	0.759	pay_sum, payment_sum, bill_sum
KNN	n_neighbors, p	2, 50	0.796	0.636	0.204	0.309	0.713	pay_sum, payment_sum

Limitation:

1. Data bias: Because this data set is from Taiwan instead of the US, it has limited application reference for consumer credit prediction in the US.
2. Features: We didn't have credit bureau data in this project. We also applied historical data, not the information recently.
3. All these analyses are only applied to the existing clients of the credit card company, not for the prospective ones.

Future work:

1. Considering the popularity of deep learning model, we can explore its application in this data set and compare it with the ensemble algorithms.
2. Develop a modeling pipeline to extract information more efficiently, providing an automated and faster solution for making credit decisions on time.