

Credit Card Default Prediction

ISOM 3360 Project Report

Spring 2020

Liu, Weiyang 20413306

Zhang, Chuang 20412479

Zhang, Zimeng 20492120

Zhou, Xinrui 20493318



Table of Contents

Project Introduction	1
Business Problem.....	1
Significance to Solve the Target Problem	1
Data Understanding	1
Data Source.....	1
Data Description	1
Data Attributes	1
Data Label	2
Data Type	2
Descriptive Data Table for Attributes	2
Missing Values	2
Skewness and Outliers	2
Class Imbalance	4
Label Imbalance	4
Model Building	4
Model 1: Decision Tree.....	4
Model 2: Logistic regression	5
Model 3: K-Means Clustering	5
Model 4: Naïve Bayes	6
Model 5: K Nearest Neighbor	7
Model 6: Ensemble Methods.....	8
Performance Evaluation	9
Project Conclusion.....	10

Project Introduction

Business Problem

Nowadays, the use of credit card becomes an integral part of modern economies. Still, default on credit card payment is considered to be a crucial problem globally. Default on credit card payment is failure to make debt payment by the due date. This may happen due to a sudden change in a person's income source such as sudden job loss, inability to work, or some health issues. It may also be deliberate, when people intentionally use the credit card, knowing they could not afford to repay loans later, until the bank stopped their cards.

Credit card companies have stringent risk requirements to minimize the default risk. To assess the creditability of card holders, they usually collect personal information such as sex, age, marital status, education level, income and so on. They also monitor the usage of each credit card and generate data including card balance, monthly bill, repayment amount and default status. Traditional programming measures are not capable of finding data patterns, and it is also inefficient and costly to use human heuristics to determine the risk of default with large datasets. Therefore, for our project, we use the credit card data from Taiwan in 2005 as an analysis example and aim to implement machine learning techniques to predict the likelihood of default of each individual card user.

Significance to Solve the Target Problem

Card issuers will run into great troubles dealing with the default account. They have to urge people to repay or even take legal action to enforce payment. If people are not solvent, banks will bear the losses itself. According to figures from the Bank of England, the default rate, which is calculated by the central bank based on a balance of responses from lenders, is 22.9%, and caused huge losses to banks.

Our project aims to help credit card companies to detect the potential default account earlier based on the features data we have, so they can take preventive actions to minimize the losses. In addition, based on the prediction, credit card companies can also issue credit card smarter to people with accounts with lower default risk and also help the customers by providing necessary suggestions to avoid default.

Data Understanding

Data Source

UCI Machine Learning Repository: <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>

This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

Data Description

Number of records: 30,000

Number of attributes: 24

Data Attributes

- LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family / supplementary credit)
- SEX: (1=male, 2=female)
- EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5,6 =unknown)
- MARRIAGE: Marital status (1=married, 2=single, 3=others)
- AGE: Age in years
- PAY_0 - PAY_6 (PAY_0 is renamed to PAY_1 for consistency in variable names): Repayment status in September - April, 2005 (-1=pay in advance for one month, 0=pay on time, 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_AMT6: Amount of bill statement in September - April, 2005 (NT\$)
- PAY_AMT1 - PAY_AMT6: Amount of previous payments in September - April, 2005 (NT\$)

Data Label

default.payment.next.month (renamed to def_pay): Whether the client will default next month

Data Type

- LIMIT_BAL, BILL_AMT1 - BILL_AMT6, PAY_AMT1 - PAY_AMT6: float64
- SEX, EDUCATION, MARRIAGE, AGE, PAY_0 - PAY_6, default.payment.next.month: int64

Descriptive Data Table for Attributes

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000
mean	167484.322667	1.603733	1.853133	1.551867	35.485500	-0.016700	-0.133767	-0.166200	-0.220667	-0.266200
std	129747.661567	0.489129	0.790349	0.521970	9.217904	1.123802	1.197186	1.196868	1.169139	1.133187
min	10000.000000	1.000000	0.000000	0.000000	21.000000	-2.000000	-2.000000	-2.000000	-2.000000	-2.000000
25%	50000.000000	1.000000	1.000000	1.000000	28.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
50%	140000.000000	2.000000	2.000000	2.000000	34.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	240000.000000	2.000000	2.000000	2.000000	41.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	1000000.000000	2.000000	6.000000	3.000000	79.000000	8.000000	8.000000	8.000000	8.000000	8.000000

BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6
30000.000000	30000.000000	30000.000000	30000.000000	3.000000e+04	30000.000000	30000.000000	30000.000000	30000.000000
43262.948967	40311.400967	38871.760400	5663.580500	5.921163e+03	5225.68150	4826.076867	4799.387633	5215.502567
64332.856134	60797.155770	59554.107537	16563.280354	2.304087e+04	17606.96147	15666.159744	15278.305679	17777.465775
-170000.000000	-81334.000000	-339603.000000	0.000000	0.000000e+00	0.000000	0.000000	0.000000	0.000000
2326.750000	1763.000000	1256.000000	1000.000000	8.330000e+02	390.000000	296.000000	252.500000	117.750000
19052.000000	18104.500000	17071.000000	2100.000000	2.009000e+03	1800.000000	1500.000000	1500.000000	1500.000000
54506.000000	50190.500000	49198.250000	5006.000000	5.000000e+03	4505.000000	4013.250000	4031.500000	4000.000000
891586.000000	927171.000000	961664.000000	873552.000000	1.684259e+06	896040.000000	621000.000000	426529.000000	528666.000000

Missing Values

EDUCATION: value 5 and 6 are “unknown”, and 0 is undocumented. They can be considered as missing values.

MARRIAGE: value 0 is undocumented. They can be considered as missing values. Every missing value is filled with a random value generated according to probability of occurrence. In this way, the overall probability of each value for a certain feature will not be amended.

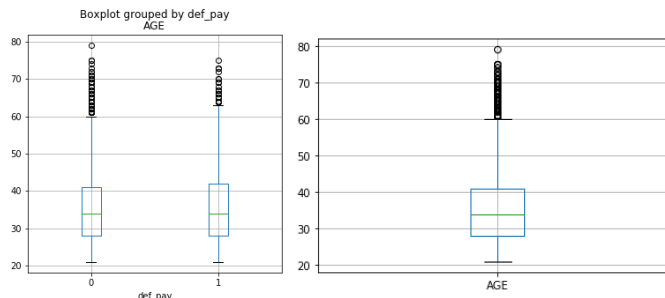
Skewness and Outliers

We assess skewness and outliers through data visualization.

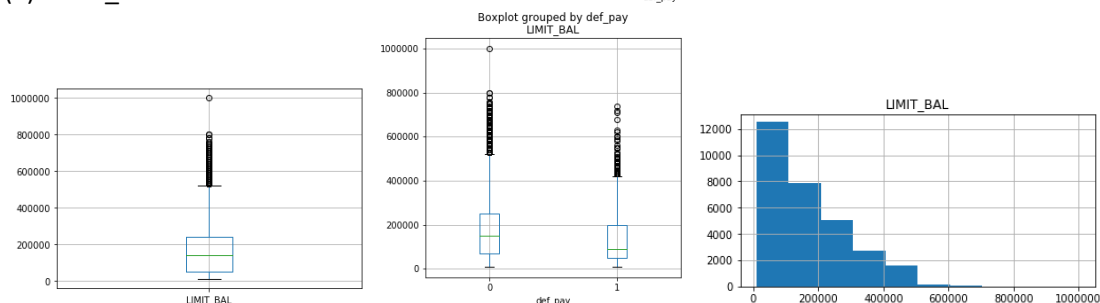
1. Numerical features

(1) AGE

There are a few outliers which will not be removed because they are meaningful for this feature. Age distribution is different for default clients and non-default clients, which implies that age is a significant feature.



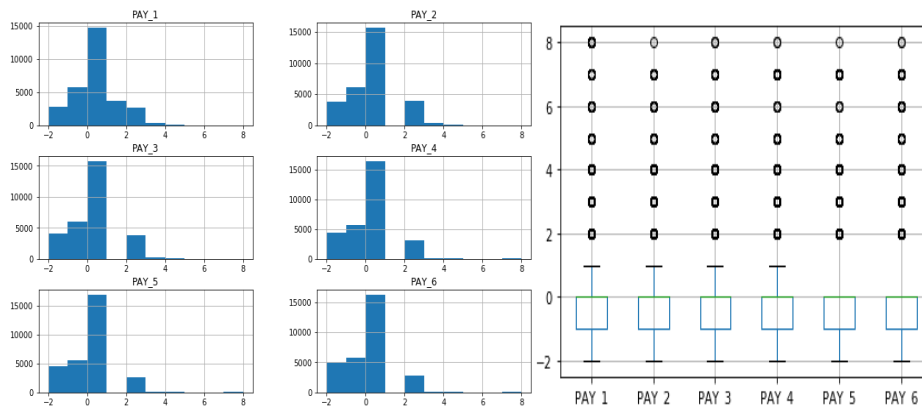
(2) LIMIT_BAL



There are a few outliers which will not be removed, because the balance limit of each client is determined by the credit card issuer which indicates that the data is relatively accurate. The balance limit distribution is different for default clients and non-default clients, which implies that it is a significant feature. We spot skewness in the distribution, so we will use normalization such as log normalization in certain models.

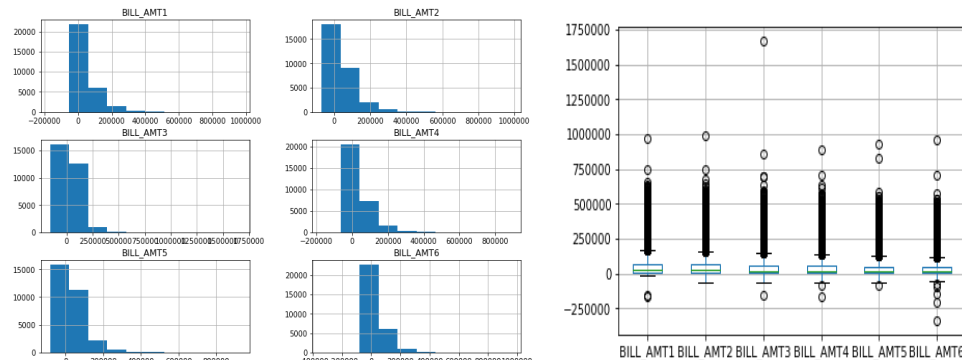
(3) PAY_1 - PAY_6

There are a few outliers which will not be removed, because they come from the defaulted clients. Even though they are the minority, they provide very meaningful data. Skewness is insignificant here. We will still apply normalization in certain models if necessary.



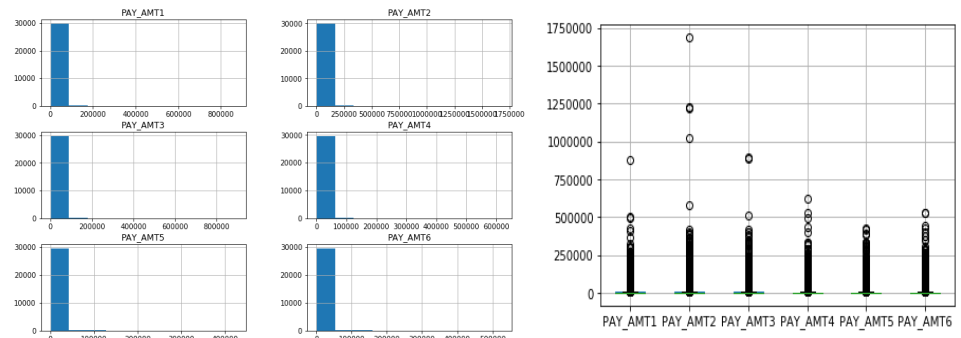
(4) BILL_AMT1 - BILL_AMT6

There are a few outliers which will not be removed, because the clients can barely lie about their bill amounts under the verification of credit card issuers, which means the data can be trusted. Skewness can be noticed here, so we will use normalization such as log normalization in certain models.



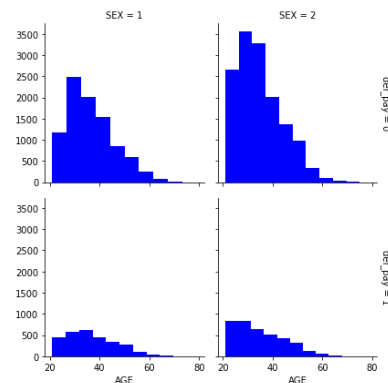
(5) PAY_AMT1- PAY_AMT6

There are a few outliers which will not be removed, because the payment amounts were recorded by the credit card issuer which means the data can be trusted. Skewness can be noticed here, so we will use normalization such as log normalization in certain models.

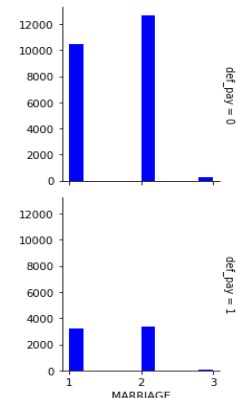


2. Categorical features

(1) SEX
Sex is a significant feature, and we can infer that male are more likely to default than female.



(2) EDUCATION
Marriage is a significant feature, and we can infer that married people are more likely to default than single people.



Class Imbalance

No significant class imbalance is shown. Thus, none of the attributes is deemed useless.

Label Imbalance

The overall default probability is 0.2212, indicating that the dataset has a mild degree of imbalance, which is not considered as a significant problem.

Model Building

Model 1: Decision Tree

Decision tree classifier is a suitable model for this classification problem. We start with the simplest model without feature engineering, then try to normalize some of the numerical features and see if the result improves.

We decide not to do binning for numerical features because decision tree will automatically do this for us. Note that there is not much to do with the categorical features since they are directly fed into the model.

Method 1: Without feature engineering

1. Data preparation: No feature engineering is carried out.
2. Modeling: We use GridSearchCV to determine the best parameters -- max_depth = 5 and max_leaf_nodes = 2000.
3. Evaluation: 10-fold cross validation gives an accuracy of 0.8196 and AUC of 0.7631.

Method 2: Normalize numerical features

1. Data preparation: No feature engineering for categorical features. Among the numerical features, the balance limit, bill amounts, and payments are skewed, so they need to be normalized. In terms of the

normalization method, we decide to choose log-scaling since the data are long-tailed. Min-max scaling is sensitive to outliers, and z-score is not suitable here since the data does not follow normal distribution.

2. Modeling: We use GridSearchCV to determine the best parameters -- max_depth = 5 and max_leaf_nodes = 1500.

3. Evaluation: 10-fold cross validation gives an accuracy of 0.8196 and AUC of 0.7631.

Conclusion: The accuracy and AUC between the 2 methods are different but the difference is minor. Thus, doing normalization for decision tree does not improve the result. The simplest model has an acceptable accuracy yet not much can be done to improve it.

Model 2: Logistic regression

Logistic regression model is another model suitable for this classification problem. For categorical features, they always require one-hot encoding. For numerical features, they have relatively high values which leads to convergence warning in logistic regression, so in both methods below we need to normalize numerical features first.

Besides normalization, we also try binning for numerical features, and see if the result improves.

Method 1: One-hot encode categorical features + Normalize numerical features

1. Data preparation: One-hot encode categorical features. Among the numerical features, the balance limit, bill amounts, and payments are skewed, so we use log-scaling on them. It turns out convergence warning still exists, so we need to consider a further normalization on all numerical features to reduce the high value.

Min-max scaling works here because it scales data to (-1,1), which works for logistic regression. We choose (-1,1) rather than (0,1) because there are negative values in our dataset.

2. Modeling: The parameter max_iter is not high enough at first, and there are errors, so we raise it to 1000.

3. Evaluation: 10-fold cross validation gives an accuracy of 0.8069 and AUC of 0.7463.

Method 2: One-hot encode categorical features + Normalize numerical features + Binning numerical features with equal intervals

1. Data preparation: The log-scaling normalization part is similar to method 1. Yet we decide not to do min-max scaling because it is sensitive to outliers which may affect the binning result. We choose the number of bins based on the data visualization result, and label encode the binned features.

2. Modeling: max_iter is set to 1000 as mentioned above.

3. Evaluation: 10-fold cross validation gives an accuracy of 0.8051 and AUC of 0.7593.

Conclusion: The accuracy drops but AUC increases after binning numerical features. We deem that AUC is a better measure, so we conclude that method 2 is a better model for logistic regression.

Model 3: K-Means Clustering

K-means clustering measures the similarity between examples by Euclidean distance. We build the clustering model using features only and evaluate with labels later. For categorical features, one-hot encoding is required. For numerical features, we need to apply normalization so that the distance will not be dominated by features that have a larger scale.

Method: One-hot encode categorical features + Normalize numerical features

1. Data preparation: One-hot encode categorical features. Among the numerical features, the balance limit, bill amounts, and payments are skewed, so we use log-scaling on them. K-means clustering requires a universal scale among numerical features, so we then use min-max scaling to ensure the scale of (-1,1). We choose (-1,1) rather than (0,1) because there are negative values in our dataset.

2. Modeling: n_clusters is set to 2, because we would like to find 2 clusters (default and non-default). After building the model and making predictions, we count the value for different clusters regarding each type of def_pay, in preparation for performance evaluation.

3. Evaluation: In this case, the cardinality is fixed due to that the number of clusters is fixed, so the way of checking whether cardinality correlates with magnitude cannot be used. Thus, the only way is to interpret the prediction results, which kind of serve as a "confusion matrix". Accuracy turns out to be around 0.5, which is similar to random guess.

Conclusion: K-means clustering gives an accuracy similar to random guess. It may not be suitable for this problem.

Model 4: Naïve Bayes

We have learnt that the multinomial Naive Bayes classifier is suitable for classification with discrete features and Gaussian Naive Bayes is suitable for dealing with continuous data. However, here we have mixed categorical and continuous numeric features. To apply the Naive Bayes model, we try 3 approaches:

1. Transform all features into categorical representation by data discretization, then apply the multinomial Naïve Bayes model on the transformed categorical features.
2. View all the categorical encoded features as numerical features and apply GaussianNB.
3. Independently fit a GaussianNB model on the numeric part of the data and a MultinomialNB model on the categorical part. Then multiply the class assignment probabilities to produce final predicted probabilities.

Method 1: Discretizing Continuous Features and Use Multinomial NB

1st trial: Data Binning with Equal Intervals

1. Data preparation
 - We apply data binning to all the numeric features: age, balance limits, previous payments and bill statements and convert string labels in age to numbers.
 - Since MultinomialNB disallow negative input, we add a constant 9 to all repayment status value.
2. Modeling and evaluation

We create a Multinomial NB model and feed all binned numeric data and categorical data as the features, and `def_pay` as the label to the model to perform a 10-fold cross validation and get the average accuracy as 0.7788. We get a confusion matrix as shown on the right and AUC=0.68. This shows that the model predicts all the instances to have `def_pay=0`, which means all customers will not default the credit card.

Confusion Matrix:
[[23364 0]
[6636 0]]

3. Conclusion

From the 1st trial, we see that the model classifies all instances as 0 and has the same performance as the benchmark model, majority-class prediction model. We suspect that this situation may have occurred because some numeric data are skewed and are concentrated in one or two intervals when binned, thus have low impact on the predicted probability. To address this problem, we try 2 methods:

- (1) Normalize skewed data by log normalization before binning.
- (2) Use data binning with equal quantity instances in each interval.

2nd trial: Data Binning with Equal Intervals + Data Normalization (*Best performer for MultinomialNB*)

1. Additional data preparation
 - We perform log normalization on the highly skewed features, limit balance and bill amounts, before binning.
2. Evaluation on new model

Similarly, we use Multinomial NB model and perform a 10-fold cross validation and the average accuracy increased to 0.7826. The confusion matrix is shown on the right and AUC increases to 0.72. Performing data normalization, we improve the recall for 1 class from 0 to 0.03.

Confusion Matrix:
[[23269 95]
[6413 223]]

3rd trial: Data Binning with Equal Quantity of Instances in Each Interval

1. Data preparation

- In this trial, we used the quantile-based discretization function *pandas.qcut* instead of *pandas.cut*, which will discretize variable into equal-sized buckets based on rank.

2. Evaluation on new model

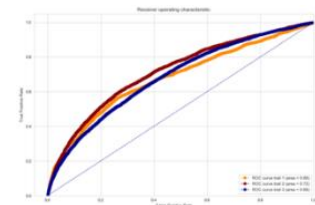
The average accuracy for 10-fold is 0.7825. The new confusion is shown on the right.

There appear more predictions on class 1 (default) and the recall for class 1 has increased from 0 to 0.19. The AUC has remained 0.68.

Confusion Matrix:
[[22218 1146]
[5378 1258]]

Comparison of 3 trials

We use the same evaluation measure on 3 trials and compare their accuracy and ROC curve. We can see from the graph that the ROC curve of trial 2 is always higher than that of trial 1 & 3. Also, trial 2 has the highest accuracy as shown below. **Trial 2 (Data binning with equal intervals + data normalization) is the best performer of MultinomialNB.**



Accuracy of trial 1 Model: 0.7787999999999999
Accuracy of trial 2 Model: 0.7830666666666667
Accuracy of trial 3 Model: 0.7825333333333333

Method 2: View All Categorical Features as Numeric and Apply GaussianNB (Best performer for GaussianNB)

The categorical features sex, education, marriage and payments are somehow ordered, so we can view them as categorical features in GaussianNB model.

1. Data preparation

- We perform log normalization on skewed features balance limits and payment amounts. Since Gaussian Naive Bayes assumes that data is normally distributed, so we have to normalize the skewed data before modeling.
- To reduce the impact of the data scale, we applied min-max scaling on all features.

2. Modeling

We create a GaussianNB model and perform a 10-fold cross validation and get the average accuracy as 0.7837. The model got the AUC=0.75.

Method 3: Use GaussianNB for Continuous Features and MultinomialNB for Categorical Features Respectively

We perform the data preparation techniques in method 1&2 separately on numeric and categorical data. We then multiply the predicted probabilities for the 2 labels from MultinomialNB and GaussianNB model to get the probabilities based on all features. The model turns out to predict all the instances as default, which is the same with majority-class prediction.

Model 5: K Nearest Neighbor

1. Data preparation

- KNN performs better when all features are numerical and continuous, here, we map all categorical features to numerical value.
- To reduce the impact of the data scale, we use *StandardScaler* to standardize scale of all features

2. Modeling

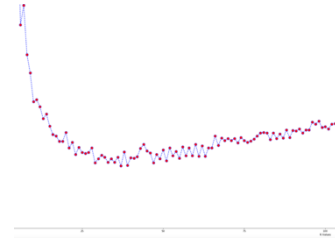
We split the dataset into 2 for training and testing purpose. Then we define the K value to be $\sqrt{30000} = 173$ and create the KNN model. The accuracy turns out to be 0. 0.8065.

3. Finetune the parameter

We try the K value from 1 to 200 and plot the error rate as shown on the right. The k-value of 23 seems to give a decent error rate without too much noise, as we see with k-values of 28 and larger.

4. Modeling with finetuned parameter

With K=23, the new model has higher accuracy of 0.8107.



Model 6: Ensemble Methods

1. Data preparation

We use power transformation to normalize the skewed numerical features including LIMIT_BAL, BILL_AMT{1-6}, and PAY_AMT{1-6}. Other features are unchanged.

2. Modeling and evaluation

We use 1 bagging method: Random Forest.

We also use 4 boosting methods: AdaBoost, and three gradient boosting methods (GradientBoosting by scikit-learn, CatBoost, LightGBM, and XGBoost).

For each of the 5 methods mentioned above, we train the model and calculate the average accuracy from a 10-fold cross validation. Then we predict the probabilities, plot the ROC curve, and calculate AUC.

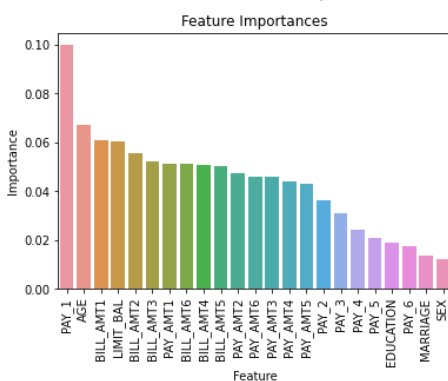
The details are as follows:

Model	Accuracy	AUC
Random Forest	0.8158	0.7645
AdaBoost	0.8163	0.7714
CatBoost	0.8193	0.7806
GradientBoosting by scikit-learn	0.8208	0.7801
LightGBM	0.8212	0.7800
XGBoost	0.8214	0.7805

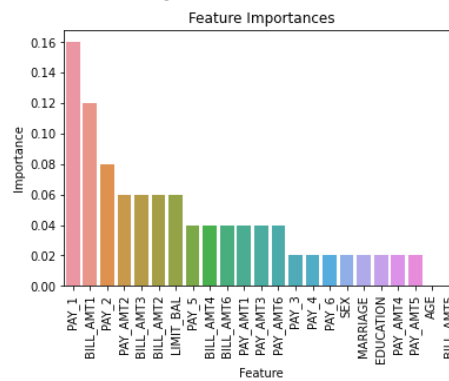
The gradient boosting methods are slightly superior to Random Forest and AdaBoost.

The differences between the four gradient boosting methods are almost negligible. XGBoost and GradientBoosting by scikit-learn run significantly faster, and the former also has the best performance.

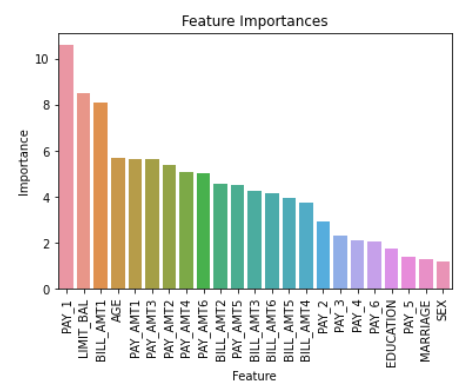
The feature importance discovered in training are visualized as follows:



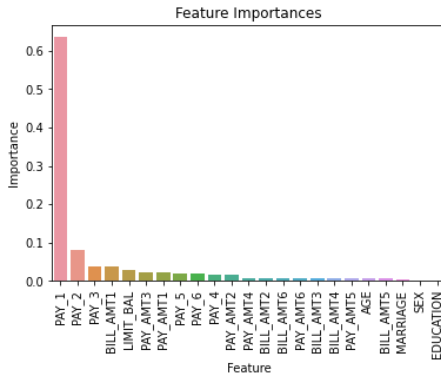
Random Forest



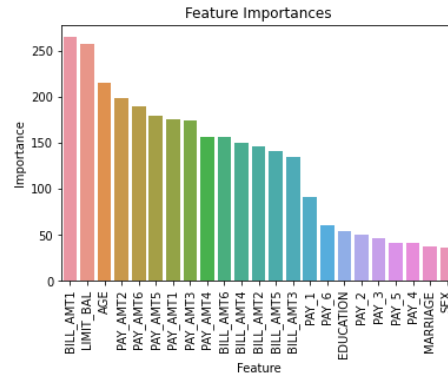
AdaBoost



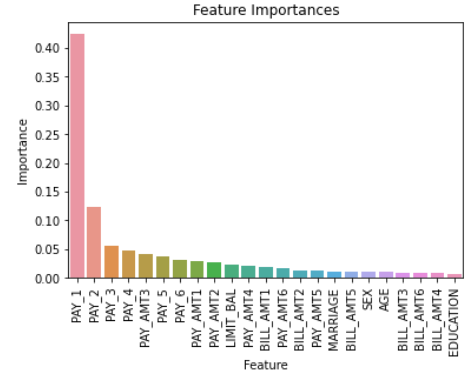
CatBoost



GradientBoosting by scikit-learn



LightGBM

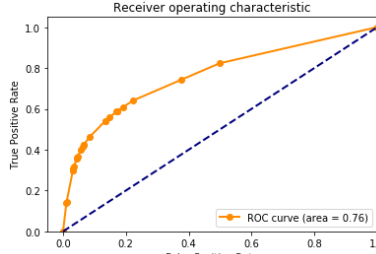
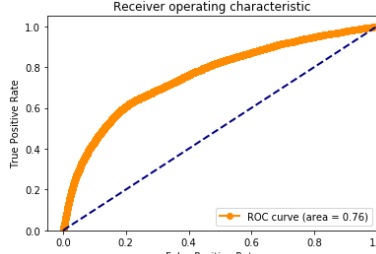


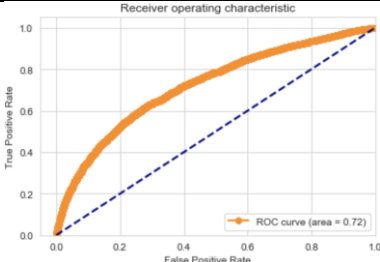
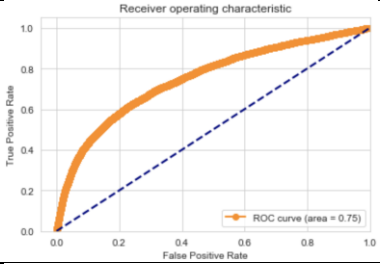
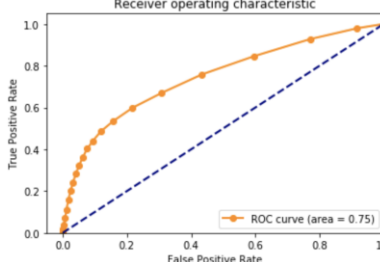
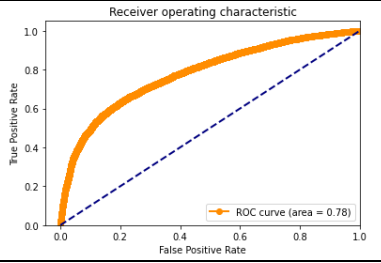
XGBoost

We can see that **the most significant feature for most models** (except LightGBM) is **PAY_1**, which is the **repayment status of the previous month**. This makes sense, because the recent repayment status can reflect one's financial difficulties if any. Besides, the features with a smaller serial number (for example, **PAY_1** compared to **PAY_6**) are **more important** because they are **more recent**.

We can also observe that those algorithms that run faster, like XGBoost and GradientBoosting by scikit-learn, tend to attach a much higher importance to the most important attribute. This may save some execution time.

Performance Evaluation

Model	Evaluation Measure	Accuracy	AUC	ROC Curve	Confusion Matrix	Precision and Recall
Decision tree	10-fold cross validation	0.8196	0.7631		Confusion Matrix: [[22343 1021] [4256 2380]] TP - True Negative 22343 FP - False Positive 1021 FN - False Negative 4256 TN - True Negative 2380	<pre> precision recall 0 0.84 0.96 1 0.70 0.36 </pre>
Logistic regression	10-fold cross validation	0.8051	0.7593		Confusion Matrix: [[22327 1037] [4809 1827]] TP - True Negative 22327 FP - False Positive 1037 FN - False Negative 4809 TN - True Negative 1827	<pre> precision recall 0 0.82 0.96 1 0.64 0.28 </pre>
K-means clustering	Directly interpreting predictions	About 0.5	N/A	N/A	Similar to confusion matrix: [[10848 12516] [3340 3296]]	N/A

Multinomial Naïve Bayes	10-fold cross validation	0.7831	0.72		Confusion Matrix: [[23269 95] [6413 223]] TP - True Negative 23269 FP - False Positive 95 FN - False Negative 6413 TN - True Positive 223	<pre>precision recall 0 0.78 1.00 1 0.70 0.03</pre>
Gaussian Naïve Bayes	10-fold cross validation	0.7837	0.75		Confusion Matrix: [[20297 3067] [3422 3214]] TP - True Negative 20297 FP - False Positive 3067 FN - False Negative 3422 TN - True Positive 3214	<pre>precision recall 0 0.86 0.87 1 0.51 0.48</pre>
KNN	Separate test set	0.8107	0.75		Confusion Matrix: [[22189 1175] [4506 2130]] TP - True Negative 22189 FP - False Positive 1175 FN - False Negative 4506 TN - True Positive 2130	<pre>precision recall 0 0.83 0.95 1 0.64 0.32</pre>
XGBoost	10-fold cross validation	0.8214	0.7805		Confusion Matrix [[22220 1144] [4215 2421]] TP: 22220 FP: 1144 FN: 4506 TN: 2421	<pre>precision recall 0 0.84 0.95 1 0.68 0.36</pre>

Comparing the AUC and accuracy of all models, we select XGBoost as our best model.

Project Conclusion

The business problem is that the default risk of credit card users is hard to predict. To tackle this binary classification problem, we used a few machine learning models including the decision tree, logistic regression, k-means clustering, Naïve Bayes, k-nearest neighbors, and some ensemble methods. In the end, we pick XGBoost, one of the boosting algorithms as the best model to use, with the accuracy of 82.14% and AUC of 0.7805.

The analysis of the ensemble methods also shed some light on the different importance of different features. One consensus of most models is that the repayment status of the previous month is the most significant feature. Another insight is that the more recent the features are, the more important they are to the model.

In the future, we can further tune the parameters of the existing models or explore more advanced methods like neural networks. We may also try to adapt our solution to similar problems in other regions outside Taiwan.