Analyzing the Credit Card crisis in Taiwan

Abstract

According to a Bloomberg report, more than 60 million of Americans used Credit cards to meet spending needs within the previous week. Almost 47% of the adults has credit card debts. Furthermore, the COVID-19 pandemic has resulted in many job losses and people strapping for money. This will increase the number of credit card defaults around the world. In this paper, we analyse the "credit card crisis" that happened in Taiwan, back in 2005.

1 Introduction

In the 1990s, the Taiwanese government loosen the bank licence regulations and allowed the formation of new banks. In order to increase their market share, many banks begin to simplify card application process and lowered the requirements for unqualified applicants. They invested big amount of money on marketing and targeted youngsters, particularly college students, to apply for their credit cards. Many of the card holders became 'credit card slaves', a term often referring to people only able to pay the minimum sum of their debt every month.

In the early 2000s, the number of credit card defaulters' amount to almost 400,000 people and the debts from credit cards increased to almost NT$ 70billions[1]. This resulted in many social issues as the debtors will explore illegal means to repay their loans. Eventually, the government issued new laws to modify the requirements of credit card applications in order to curb the issue.

In this paper, we will be using a dataset of Taiwanese credit card applicants from 2005. Using machine learning methods, we will study the attributes of the clients and identify the clients that got the highest potential to default their payments. By identifying the potential defaulters, the credit cards issuer will be able reject those applications to minimize their risk and lower their losses.

2 Data exploration

2.1 Data overview

Dataset: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

The dataset consisted of 30,000 clients' information. The target is whether the client will default their next payment. There are 24 attributes in the dataset, ranging from Customer ID, Credit Limit, Demographics factors and payment history in Taiwan from April 2005 to September 2005. The target for the dataset is whether the client will default the next payment.

2.2 Data exploration

2.2.1 Missing Data

There are no missing data in our dataset as shown in Appendix B.

[1] NT$70billion = USD$2.44 Billion

2.2.2 Class imbalance

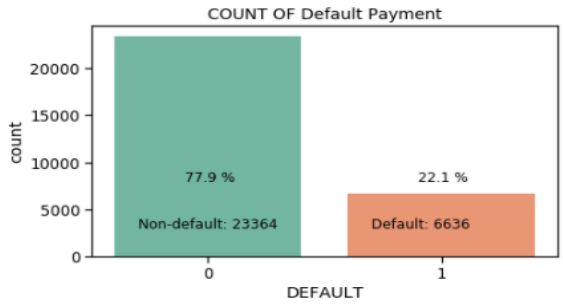


Figure 1: Count of default payment

The most significant finding is the imbalance of the Target. There are 77.9% of non-default counts compared to only 22.1% of default counts. In order to mitigate this problem, we have to balance the class by applying resampling technique on the data. (Figure 1)

2.2.3 Credit information

Credit Limit: Higher defaults for Credit Limit below NT$100,000. (Figure 2) This could be resulted by the lower income group struggling to cope with the increasing standard of living.

Payment Repayment Status: More discriminatory power in the month of September and August. (Figure 3) Taiwan was affected by Typhoon Talim, a Cat 4 super typhoon that resulted in almost NT 1.6billion losses, in August 2005. That could have caused the sharp increase in number of defaults for those months.

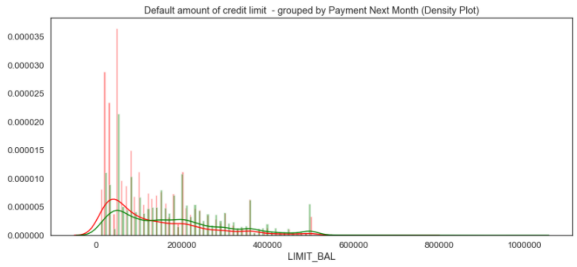


Figure 2: Default per credit limit

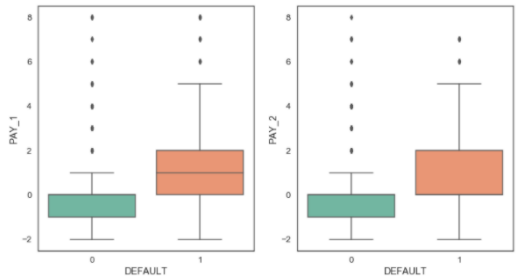


Figure 3: Default for Aug(PAY_2) and Sept(PAY_1) 2005

2.2.4 Feature correlation

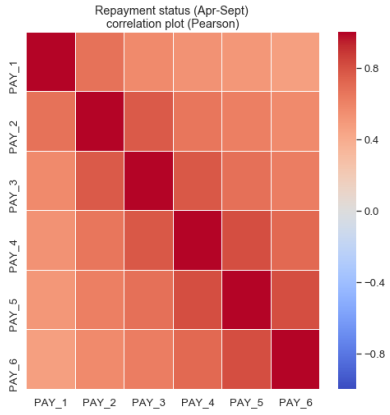


Figure 4: Correlation Plot for Repayment Status

Correlation between the repayment status is decreasing with distance between months and the lowest correlations are between Sept and April 2005.

3 Data pre-processing

We dropped the 'ID' feature. We tried the data with and without ID but the performance of our models did not

deviate much. Hence, we decided that ID is just a unique index. Moreover, in our dataset, the order of 'ID' is from 1 to 30000. This implies that ID has no relation and has not much effect on the target variable (default payment).

During data exploration, we saw that there were 7 levels of 'education'. The level 0, 5 and 6 were unknown and had very less data points, so we clubbed it into level 0. We saw the same thing in the feature 'Marriage' but when we clubbed the levels, it effected the accuracy negatively. So, we opted out of modifying any other feature or do other pre-processing.

We categorised the categorical data to type 'Category'.

After the above steps, we addressed the imbalance problem by doing oversampling and scaled the data as our dataset have a huge variation among the variables as shown below (figure 5).

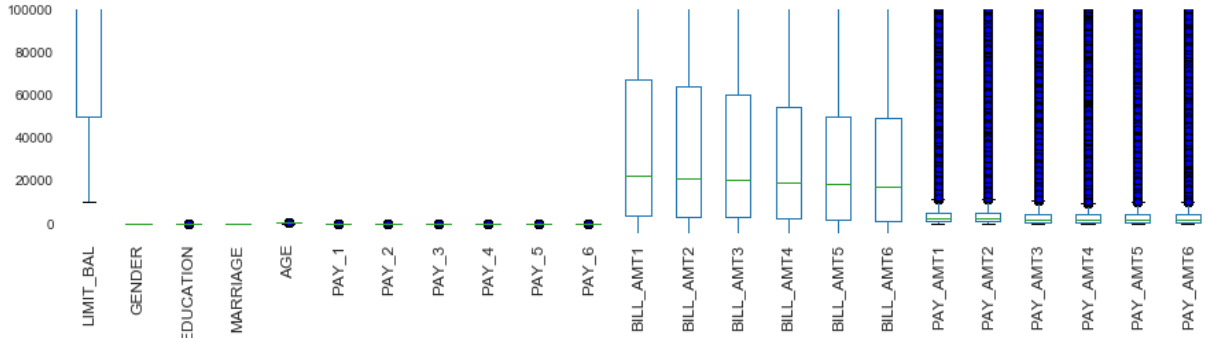


Figure 5: Variation of features

Typically, this is done by removing the mean and scaling to unit variance. However, since our data has considerable number of outliers, it can influence the sample mean and variance in a negative way. In such cases, the median and inter-quartile range often gives better results. Thus, we used RobustScalers. This was done on the training data and was then applied to the test data using the median and interquartile range of the training data. For handling the unbalanced data, we oversampled the minority class that is 'Default' by using SMOTE (Synthetic minority over-sampling technique). SMOTE synthesises new minority instances between existing (real) minority instances. In general, one might say that SMOTE loops through the existing, real minority instance. At each loop iteration, one of the K closest minority class neighbours is chosen and a new minority instance is synthesised somewhere between the minority instance and that neighbour.

We used scaling and oversampling for all the machine learning models but we didn't use scaling for decision trees as such algorithm can handle data of various scales internally within itself. For methods like SVM, Naïve Bayes, standardisation is important. For logistic regression, though we are only concerned about the prediction, different scales can affect the interpretation of the co-efficients in terms of magnitude. So, we decided to scale the training data before modelling.

For each machine learning algorithms, we split the data into training and test data in the ratio 70:30. The training and test data was further split it into features(x) and target(y) variables.

4 Data modelling

4.1 Logistic Regression

4.1.1 Description

Logistic Regression is one of the simplest algorithms which estimates the relationship between one dependent binary variable and independent variables, computing the probability of occurrence of an event. The sigmoid function is a function that resembles an "S" shaped curve when plotted on a graph. It takes values between 0 and 1 and "squishes" them towards the margins at the top and bottom, labelling them as 0 or 1. Since our target

variable default payment is a binary variable, logistic regression would be a good approach for our dataset.

4.1.2 Result

According to the accuracy of 68.7%, we are doing moderately good in terms of accuracy rate. Although we have a high recall of 64.4%, we are having a low precision of 37.99%. Since our dataset are of high dimension, the <70% accuracy may be due to the overfitting of the dataset. Furthermore, logistic regression assumes linearity among the dependent and independent variables where in our dataset, it might not be the case. Hence, generally since logistic model still yields a moderately good accuracy result, and it is one of the simplest models for classification problem, it is generally still a good model to use.

4.2 Decision trees

4.2.1 Description

Decision Tree is another very popular algorithm for classification problems because it is easy to interpret and understand. An internal node represents a feature, the branch represents a decision rule, and each leaf node represents the outcome. Some advantages of decision trees are that they require less data pre-processing, i.e., no need to normalize features. However, noisy data can be easily overfitted and results in biased results when the data set is imbalanced. Hence, oversampled are required in training dataset.

We tried different variations of decision trees to improve the results. (Figure 6)

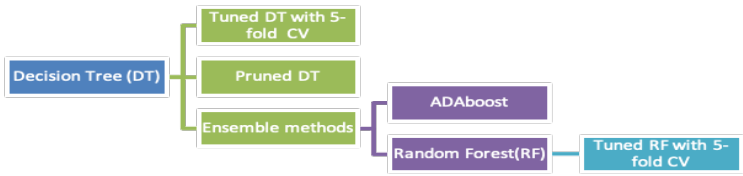


Figure 6: Decision tree models used for predicting default

AdaBoost is a boosting ensemble model and works especially well with the decision tree. Boosting model's key is learning from the previous mistakes, e.g. misclassification data points. AdaBoost learns from the mistakes by increasing the weight of misclassified data points. Random forest classifier is comprised of multiple decision trees. It creates different random subset of decision trees from the training set as its predictors and selects the best solution by means of voting. As a result, the Random Forest model avoids overfitting problems. Pruning reduces the size of decision trees by removing parts of the tree that do not provide power to classify instances.

4.2.2 Tuning Parameters

For the cross validation of decision tree and random forest with five-fold cross validation, we used RandomizedSearchCV function to select the best parameters. For decision tree , it gives parameters of {'min_samples_split': 100, 'min_samples_leaf': 30, 'max_features': 10, 'max_depth': 8, 'criterion': 'entropy'} while for random forest, the parameters are {'n_estimators': 150, 'max_features': 3, 'max_depth': 15, 'criterion': 'entropy'}.

4.2.3 Feature Importance

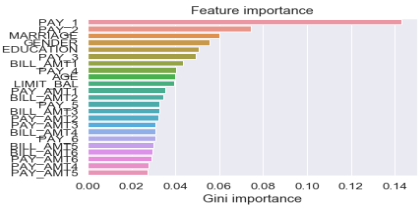
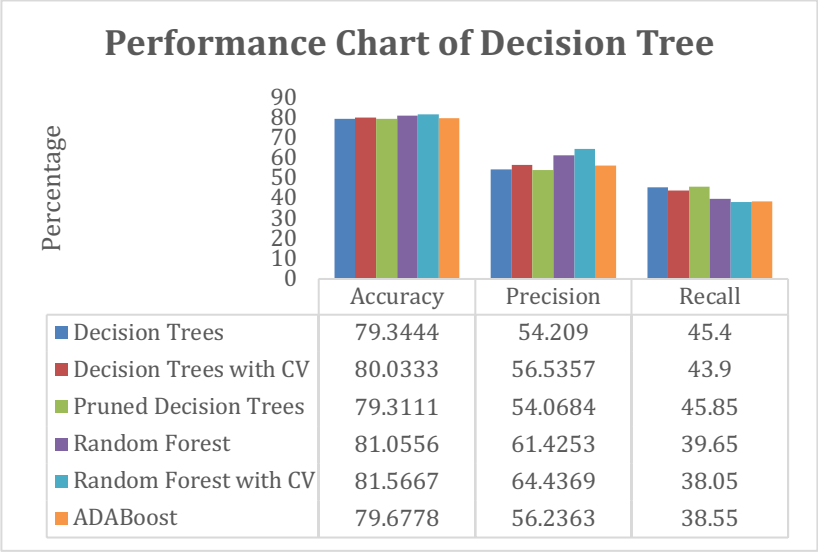


Figure 7: Feature Importance in Random Forest

In Random Forest, feature importance is plotted. PAY_1 which is the repayment status in September was observed to have the largest influence on the predicting the default payment. This is followed by PAY_2 (repayment status in August) and Marriage.

4.2.4 Result



	Accuracy	Precision	Recall
Decision Trees	79.3444	54.209	45.4
Decision Trees with CV	80.0333	56.5357	43.9
Pruned Decision Trees	79.3111	54.0684	45.85
Random Forest	81.0556	61.4253	39.65
Random Forest with CV	81.5667	64.4369	38.05
ADABoost	79.6778	56.2363	38.55

Figure 8: Decision tree results

After tuning for both decision tree and random forest using the five-fold CV, the accuracy and precision have increases slightly while recall had decreased slightly too.

Most of the decision tree models and its variants have quite a balance between recall and precision with the differences less than 10% except for random forest, random forest with CV and AdaBoost.

From the performance chart, there is an observation that there is a trade-off between precision and recall.

If we consider accuracy, all the models have accuracy greater than 79%. Decision trees and its variant models predict well but random forest with cross validation performs the best. As per precision, random forest with cross validation performs the best. As per recall, pruned tree performs the best. Furthermore, Decision Trees are easy to understand and interpret. Overall, decision trees and its variant models are still a pretty good model to predict default payments.

4.3 Naïve bayes classifier

4.3.1 Description

A Naive Bayes classifier is a probabilistic machine learning model that is used for classification task. The crux of the classifier is based on the Bayes theorem. It is easy and fast to predict class of test data set. It also performs well in multi class prediction. When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and we need less training data. It performs well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

4.3.2 Results

According to the results, very high recall score of 85.55%. 85.55% of recall is good because according to our aim to minimize the risk and loss of credit card issuers. Hence, it means that out of true default, 85.55% of them are correctly identified as default which helps in minimizing the risk and loss of the credit card issuers. However, this model suffers from a very low accuracy and precision score of 44.25% and 26.57% respectively. This means that out of all the predicted default, only 26.57% of them are correctly identified as default and out of the test data only 44.25% of them are detected correctly.

Hence, despite the high recall, it is still a bad model for our dataset. Naïve bayes assume the independency of predictor variables which is impossible that a data set comes completely uncorrelated to each other for the predicted variables and numerical inputs are known to be normal distribution by Naïve Bayes Classifier. These might be the reason it results in the low accuracy and precision score. Hence, that is why Naïve bayes is not a good model for our dataset. Often, Naïve bayes is called the ideal classifier.

4.4 Support vector machine (SVM)

4.4.1 Description

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. SVM is a good model when we have no idea of the data, works well with any data structure, generally have lesser risk of overfitting then other models and compared to deep learning models like ANN, SVM gives a better result most of the time. On the other hand, SVM requires a long training time for a large dataset, difficult to understand and interpret and not easy to fine-tune the parameters to visualize their impact. Hence, we used five-fold cross validation (CV) to select the best parameters. The parameters that we used for SVM after CV are cost of 1 and gamma of 0.01. We used GridSearchCV to select the tuning parameters. In SVM, we used the default parameters which are SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)

4.4.2 Results

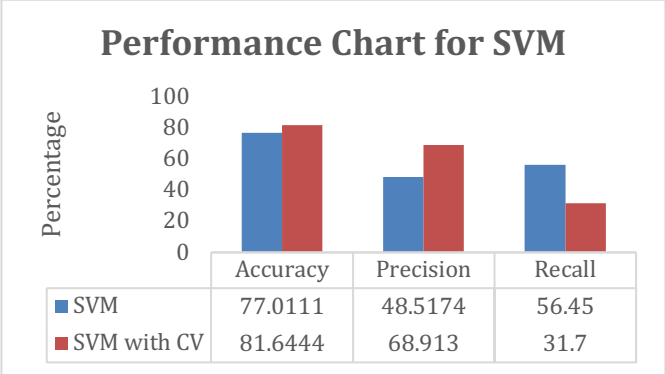
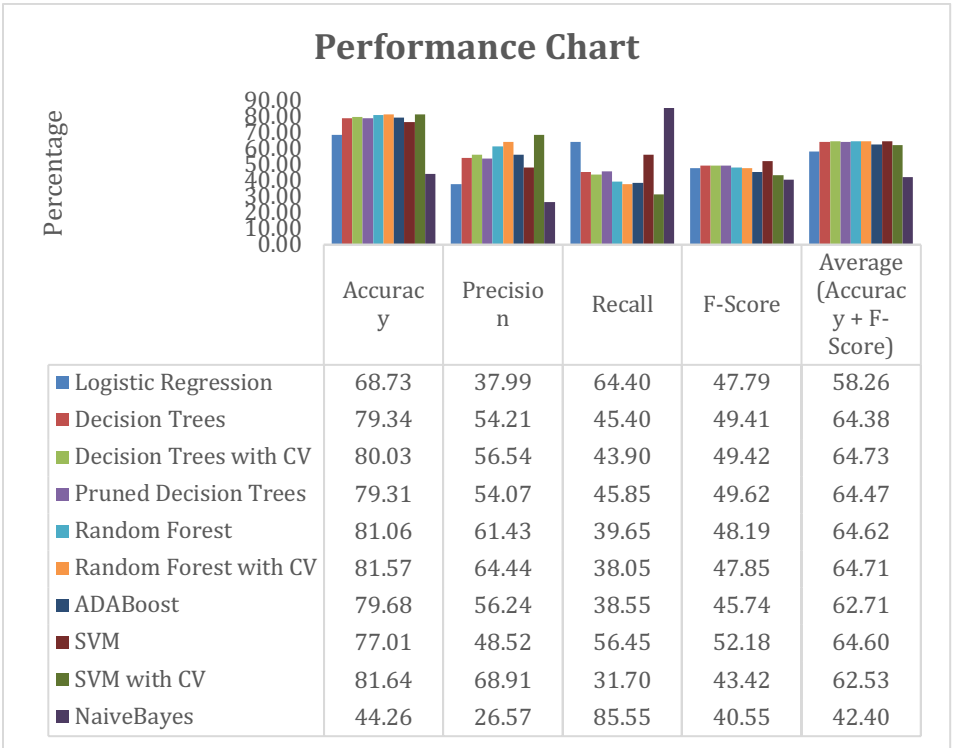


Figure 9: Performance Chart of SVM

After tuning, the accuracy and precision increases from 77 % to 81.6% and 48% to 68% respectively. However, a much lower recall from 56% to 31%. There seems to be a trade-off between the precision and recall. Overall, SVM models are generally still a good model to use. However, SVM was recommended instead of SVM with CV. The reason are the cost of using SVM with CV took too much time to train the data as in our case, it took six to seven hours to train the data and it yields a larger difference between the precision and recall. Hence, SVM was recommended over SVM with CV.

5 Result comparison & Conclusion of model



	Accuracy	Precision	Recall	F-Score	Average (Accuracy + F-Score)
Logistic Regression	68.73	37.99	64.40	47.79	58.26
Decision Trees	79.34	54.21	45.40	49.41	64.38
Decision Trees with CV	80.03	56.54	43.90	49.42	64.73
Pruned Decision Trees	79.31	54.07	45.85	49.62	64.47
Random Forest	81.06	61.43	39.65	48.19	64.62
Random Forest with CV	81.57	64.44	38.05	47.85	64.71
ADABoost	79.68	56.24	38.55	45.74	62.71
SVM	77.01	48.52	56.45	52.18	64.60
SVM with CV	81.64	68.91	31.70	43.42	62.53
NaiveBayes	44.26	26.57	85.55	40.55	42.40

Figure 10: Performance Chart

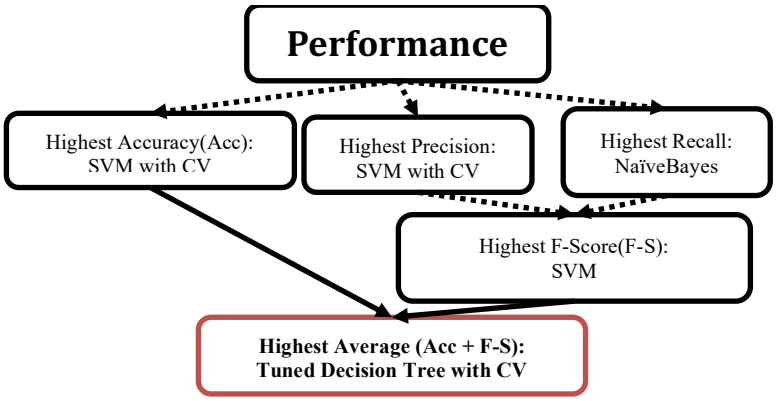


Figure 11: Performance Tree

- **Accuracy**: Out of all the test data, the % of them predicted correctly
  - High Accuracy → Low Misclassification Error
- **Precision**: Out of predicted default, the % of them predicted as default correctly
- **Recall**: Out of the true default, the % of them predicted as default correctly
  - High Recall → Low Type II error
- **F-Score**: Combination of Precision and Recall

The highest accuracy and precision are SVM with CV which are 81.64% and 68.91% respectively while the highest recall is NaïveBayes with 85.55%. We cannot conclude solely based on these three scores because from our result, there is often a trade-off between the precision and recall. Furthermore, as previously stated, NaiveBayes is not a good model. Hence, we decide to combine precision and recall giving F-score with the best performance of it being SVM with 64.6%. Finally, high accuracy is a very important too as it would mean a lower misclassification error and that model works well. Other than being able to minimize the risk and loss of credit card issuers, credit card issuers would want to earn from the non-defaulters too. Thus, we combine accuracy and F-Score by averaging them with equal weights to give the best model of tuned decision tree with CV of 64.73%. Furthermore, we could select the best tuning parameters using CV and Decision Tree is faster and easier to interpret and understand the results.

To conclude, we recommend tuned decision tree with CV to predict the default payment.

6 Reflection and Future Direction

In this course, we have learnt several methods such as supervised learning, unsupervised learning, and deep learning. We managed to apply the pre-processing techniques that we have learnt during the course onto the dataset. We also managed to tune the parameters to improve the performances of the model. All these enabled us to have a better understanding of how we can apply machine learning onto real life applications and problems.

In the future direction, we could do more different pre-processing methods and compare the results such as transformation of variables in the dataset or dropping of variables. We could use more different evaluation methods such as AUC and ROC to further evaluate the performance of our model. Other than oversampling, we could try under-sampling too and compare the results between doing oversampling and under-sampling. Furthermore, we could do more diverse technique like deep learning to compare and further evaluate the result.

References

[1] Taiwan's Credit Card Crisis, https://sevenpillarsinstitute.org/case-studies/taiwans-credit-card-crisis/

[2] Nearly 25 million active credit cards in Taiwan, https://www.taiwannews.com.tw/en/news/2840492

Appendix A: Dataset information

```
Credit_Data.info()
```

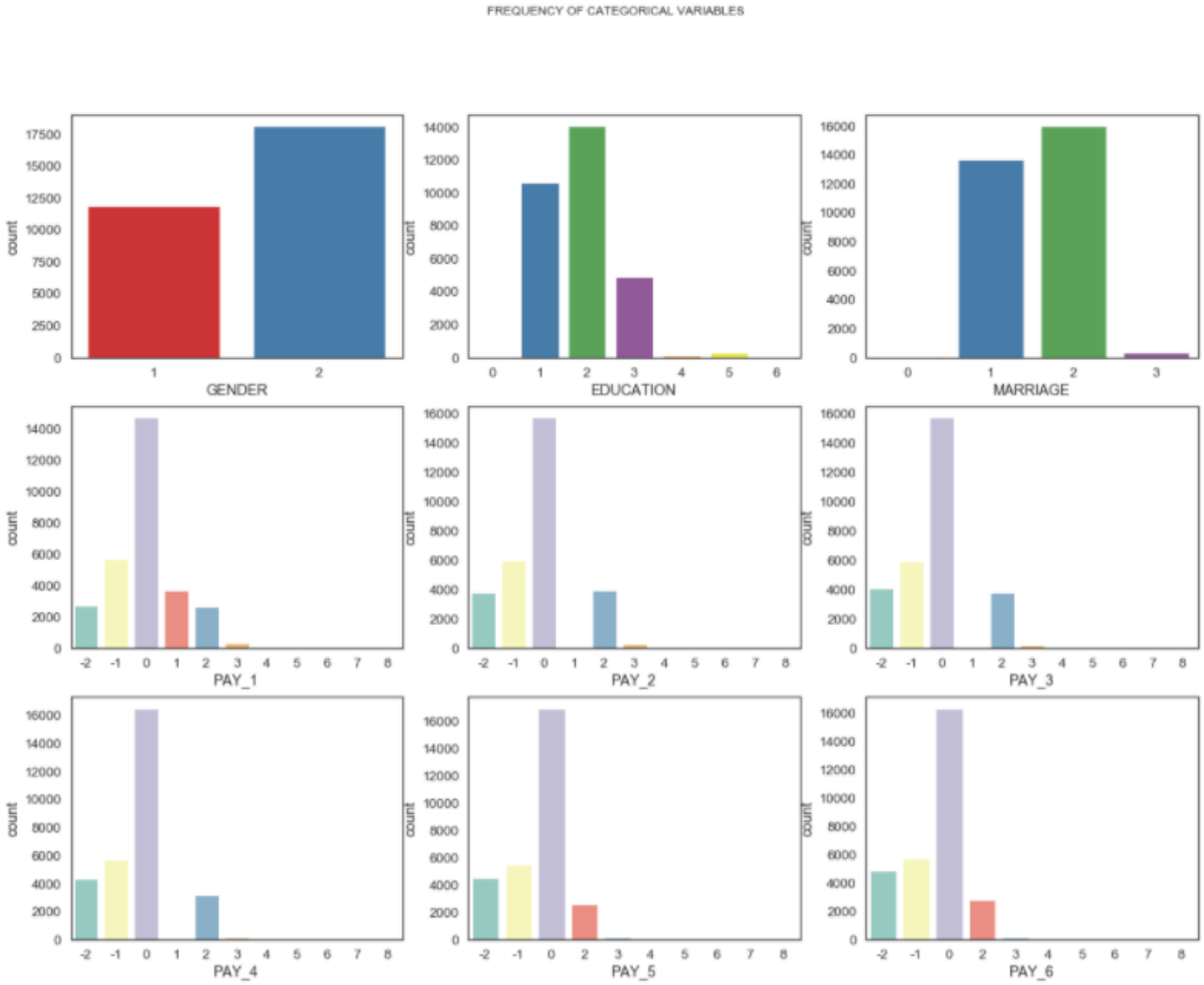
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 25 columns):
ID           30000 non-null int64
LIMIT_BAL    30000 non-null int64
GENDER       30000 non-null int64
EDUCATION    30000 non-null int64
MARRIAGE     30000 non-null int64
AGE          30000 non-null int64
PAY_1        30000 non-null int64
PAY_2        30000 non-null int64
PAY_3        30000 non-null int64
PAY_4        30000 non-null int64
PAY_5        30000 non-null int64
PAY_6        30000 non-null int64
BILL_AMT1    30000 non-null int64
BILL_AMT2    30000 non-null int64
BILL_AMT3    30000 non-null int64
BILL_AMT4    30000 non-null int64
BILL_AMT5    30000 non-null int64
BILL_AMT6    30000 non-null int64
PAY_AMT1     30000 non-null int64
PAY_AMT2     30000 non-null int64
PAY_AMT3     30000 non-null int64
PAY_AMT4     30000 non-null int64
PAY_AMT5     30000 non-null int64
PAY_AMT6     30000 non-null int64
DEFAULT      30000 non-null int64
dtypes: int64(25)
memory usage: 5.7 MB
```

Appendix B: No missing data

```
# Check for missing values
Credit_Data.isnull().sum()
```

```
ID           0
LIMIT_BAL    0
GENDER       0
EDUCATION    0
MARRIAGE     0
AGE          0
PAY_1        0
PAY_2        0
PAY_3        0
PAY_4        0
PAY_5        0
PAY_6        0
BILL_AMT1    0
BILL_AMT2    0
BILL_AMT3    0
BILL_AMT4    0
BILL_AMT5    0
BILL_AMT6    0
PAY_AMT1     0
PAY_AMT2     0
PAY_AMT3     0
PAY_AMT4     0
PAY_AMT5     0
PAY_AMT6     0
DEFAULT      0
dtype: int64
```

Appendix C: Overview Count of categorical variables



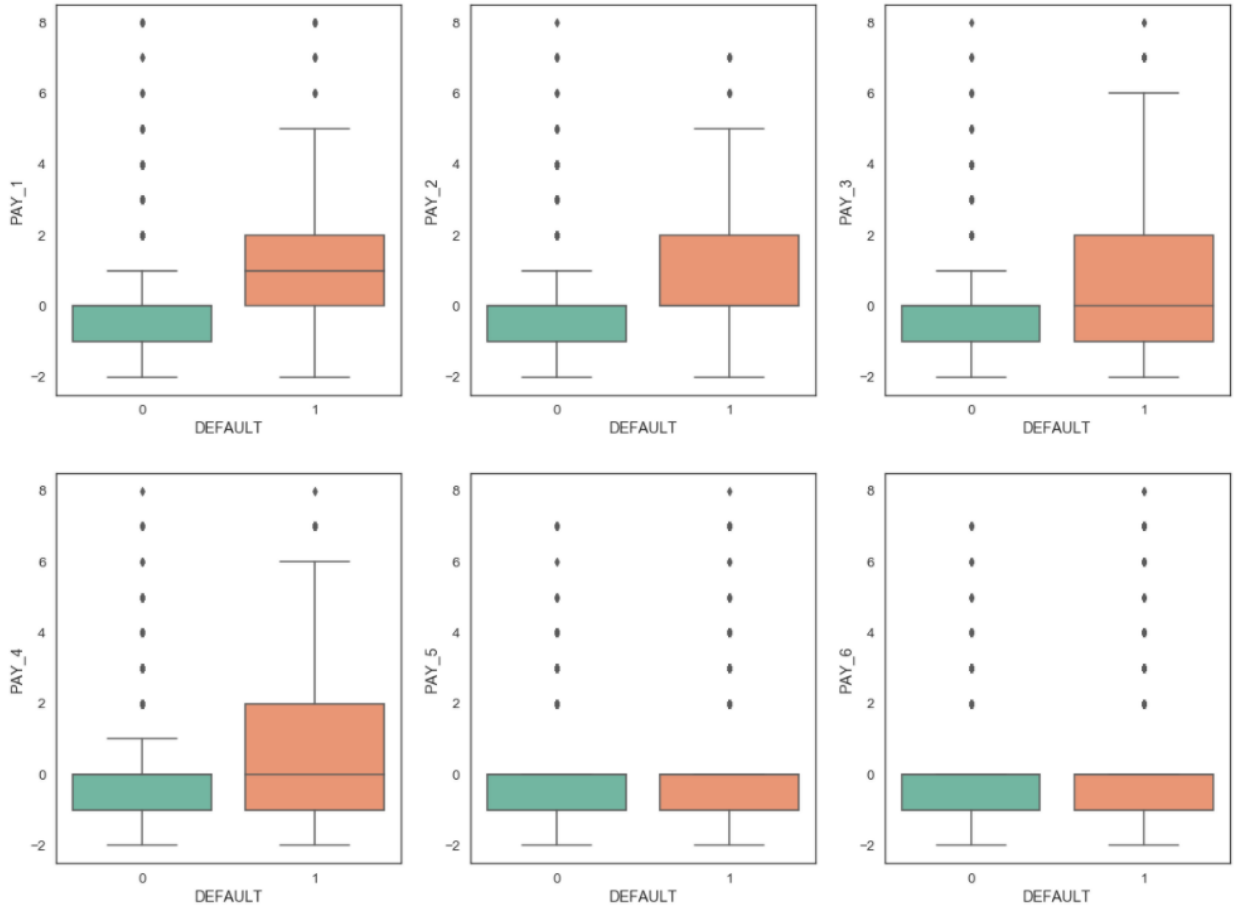
- GENDER: We do not have any unknown labels. 1=Male and 2=Female.

- EDUCATION: We have unknown labels of 0,5,6 in education. Since the number of data for labels 0,5,6 is not a lot, we can clutter these labels together as unknown. Known labels are 1= Graduate, 2= University student, 3 = high school and 4 = others.

- MARRIAGE: We have unknown labels of 0.

- Repayment Status: We have unknown labels of 0 and -2 in our data, we can see that the number of repayment status of label 0 in each month is the highest and label 0 is the majority class, hence, if we remove the unknown observations, it would have a great loss.

Appendix D: Repayment Status Boxplots



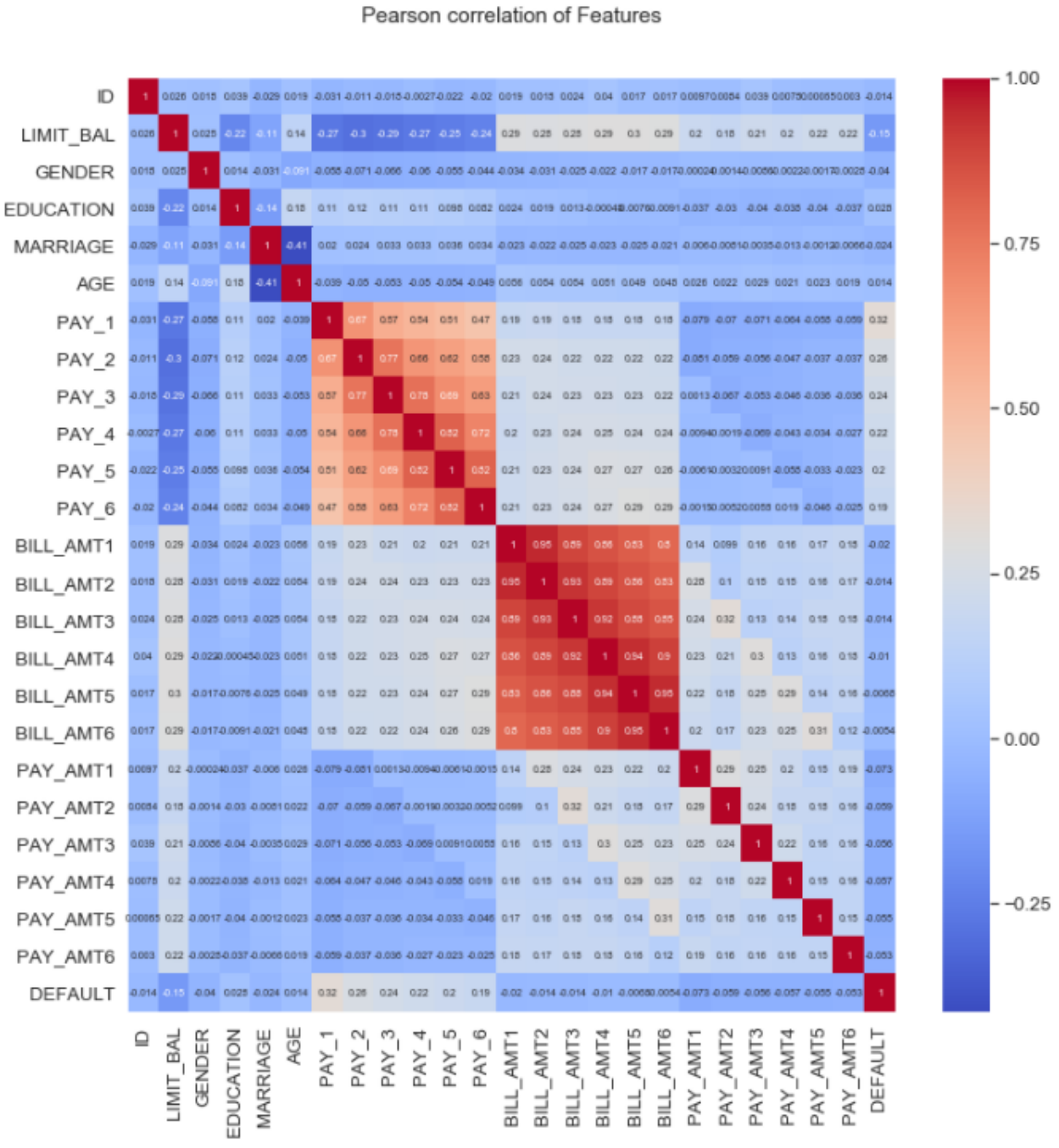
Observe:

- In September and August, there is more distinctive between the default and non-default. It seems that PAY_1 (Repayment status in September) and PAY_2 (Repayment status in August) has more discriminatory power than the repayment status in other months.

Appendix E: Age and Credit Limit with Target

Observe: More default is detected around the age from 30 to 5 at around 50K to 100K credit limit which are shown by more pink dots which are default payment cluttering around the area in the scatter plot.

Appendix F: Feature Correlation Matrix



Appendix F: Performance for Precision, Recall and Accuracy

	Logistic Regression	Decision Trees	Decision Trees with CV	Pruned Decision Trees	Random Forest	Random Forest with CV	ADABoost	SVM	SVM with CV	NaiveBayes
accuracy	68.7333	79.3444	80.0333	79.3111	81.0556	81.5667	79.6778	77.0111	81.6444	44.2556
precision	37.9941	54.209	56.5357	54.0684	61.4253	64.4369	56.2363	48.5174	68.913	26.5724
recall	64.4	45.4	43.9	45.85	39.65	38.05	38.55	56.45	31.7	85.55