Credit card default prediction

feifei yan

Contents

Introduction

Problem statement

In Taiwan, credit card issuers faced a serious debt crisis in 2006. In order to increase market share, credit card issuers issued credit cards to unqualified applicants who overused their cards but had no repayment ability. More than half a million people were not able to repay their loans and the debt reached $268 billion USD.

This is a big challenge for credit card companies, cardholders, and the whole financial system. From the perspective of risk management, the goal of this project is to predict default status of customers given their information on demographic factors, credit data, history of payment, and bill statement. Credit card companies are the primary target audience of this project. A good prediction model will not only help companies identify existing customers who are likely to default, offer them financial advice, and issue regulations, but will also help companies decide whether to offer credit cards to new customers.

Dataset

The dataset is downloaded from Machine Learning Repository website. It has 30,000 rows and 24 attributes and contains information from April 2005 to September 2005. The target variable is default (1=default, 0= non-default). There are 23 predictor variables. I have listed them below:

1. LIMIT_BAL: Amount of given credit (in NT dollars, include both individual consumer credit and his/her family (supplementary) credit)
2. SEX (1=male, 2=female)
3. EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
4. MARRIAGE: Marital status (0= unknown, 1=married, 2=single, 3=others)
5. AGE: from 21 to 79
6. PAY_0: Payment record in September 2005 (-2,0= unknown, -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)
7. PAY_2: Payment record in August 2005 (-2 ~ 9)
8. PAY_3: Payment record in July 2005 (-2 ~ 9)
9. PAY_4: Payment record in June 2005 (-2 ~ 9)
10. PAY_5: Payment record in May 2005 (-2 ~ 9)
11. PAY_6: Payment record in April 2005 (-2 ~ 9)
12. BILL_AMT1: Amount of bill statement in September 2005 (in NT dollar)
13. BILL_AMT2: Amount of bill statement in August 2005 (in NT dollar)
14. BILL_AMT3: Amount of bill statement in July 2005 (in NT dollar)
15. BILL_AMT4: Amount of bill statement in June 2005 (in NT dollar)
16. BILL_AMT5: Amount of bill statement in May 2005 (in NT dollar)
17. BILL_AMT6: Amount of bill statement in April 2005 (in NT dollar)
18. PAY_AMT1: Amount of previous payment in September 2005 (in NT dollar)
19. PAY_AMT2: Amount of previous payment in August 2005 (in NT dollar)
20. PAY_AMT3: Amount of previous payment in July 2005 (in NT dollar)
21. PAY_AMT4: Amount of previous payment in June 2005 (in NT dollar)
22. PAY_AMT5: Amount of previous payment in May 2005 (in NT dollar)
23. PAY_AMT6: Amount of previous payment in April 2005 (in NT dollar)

Data cleaning

After initial review, the dataset is relatively clean. I perform few more steps to make it consistent and tidy. First, both marriage and education have some unknown values. I group them into the "others" categories. Second, listing separate variables for each month increases redundancy. In order to make the information more useful, I create four new variables. Using PAY_0 to PAY_6, I create missing_payment indicating how many times a customer missed payment in the last six months. Using BILL_AMT1 to BILL_AMT6, I create a variable called avg_bill to get the average bill amount for the last six months. Similarly, avg_payment is created to show average repayment amount for the last months. The last variable that I create is utilization ratio, which is equal to average bill amount divided by the amount of given credit. (LIMIT_BAL). Original variables are dropped to avoid data redundancy. Third, I use binning technique for the variable AGE to facilitate the analysis.

EDA

In this part, I will analyze and interpret the data using visualization techniques to obtain some insights.

This dataset has no missing values. From boxplot (figure1), it does have some outliers for continuous variables. However, the outliers all fall in the normal range. For example, some consumers have really high credit limits, and it is common in the real world. Outliers can capture valuable information too, so I choose to keep them.

From Figure 2, the dataset is imbalanced with 78% non-default (0) and 22% default (1).

From figure 3, defaulters are more likely to have lower credit limits, spend similar amount of money on average as non-defaulters but pay back less, and have higher utilization ratio.

I also have some interesting findings from the bar plots. From figure 4, male is more likely to default than females. From figure 5, married people are more likely to default than single people. From figure 6, high school degree holders are more likely to default than other degree holders. From figure 7, middle age people are more likely to default than young people. From figure 8, as the number of missing payments increases, the likelihood of default increases too. Missing_payment variable has the strongest correlation with the target variable in the correlation matrix (figure 9).

I also study the relationship between given credit and other variables. From figure 10 and 11, we can see that senior men usually have higher credit limits than other groups. Graduate school holders tend to have higher given credit especially if you are married.

Data preparation

After getting some insights, I do more to transform the data. For categorial variables such as education and age which have hierarchy in them, I use label encoding to relabel them according to the order. For example, high school is 1, university is 2, and graduate school is 3. For categorial variables that do not have a natural rank ordering such as sex and marriage, I use one hot encoding to avoid the limitations of label encoding. However, one hot encoding creates dummy variable trap. For continuous variables that are in different ranges, I apply feature scaling for better model performance.

In the correlation matrix, some predictors are highly correlated. For example, SEXmale and SEXfemale are negatively correlated (-1) due to the dummy variable trap. Utilization ratio is highly correlated with avg_bill because the former is calculated from the latter. To deal with this, I use principal component analysis (PCA) to reduces multicollinearity. After PCA, there are 10 predictor variables and 1 target variable.

Model selection and development

Selection

I split the dataset into 80% for training and 20% for testing. Default is a categorial variable. This problem is a classification problem. Among classification algorithms, I select 5 models for training, namely Random Forest, Naïve Bayes, XGBoost, Logistic Regression and KNN. XGBoost is decision-tree-based ensemble algorithm that uses a gradient boosting framework. Different from bagging which decreases variance, boosting decreases bias. In Bagging, each model is built independently whereas in Boosting new models are influenced by performance of previously built models. It is very popular in recent year for its prediction speed and performance. So, I choose to use it here.

Development

I use random search to tune the parameters for some algorithms. For random forest, I tune both mtry and nodesize. For XGBoost, I tune the parameters including max_depth, eta, lambda, min_child_weight, colsample_bytree, subsample and nrounds. For KNN, I tune k and l. Because we have imbalanced data, I use stratified 5-fold cross-validation for sampling. To boost the training speed, I use parallel processing. For each algorithm, I train the models with three datasets. I use the original training data and the other two datasets using oversampling and under-sampling. Oversampling randomly duplicates examples in the minority class (1). Under-samplings randomly delete examples in the majority class (0). Both are used to solve imbalanced dataset issue in this project for better performance.

Model evaluation

Metrics

For this project, accuracy is not a good metric since data are imbalanced. A high accuracy doesn't mean that we have a good model because it doesn't measure false positive and false negative. In credit card default case, false positive leads to unsatisfied customers or loss of new customers because they are wrongly classified as defaulters. False negative directly leads to financial loss because it fails to identify defaulters. Precision is a good metric to use when the costs of false positive is high. Recall is a good metric to use when the cost associated with false negative is high. F-score is a weighted average of precision and recall. All three metrics are useful. Credit card companies don't want to lose market share or suffer from financial loss.

Performance

In table 1, I list the best model performance for each algorithm among original data, oversampling data and under-sampling data. I notice that logistic regression using under sampling works slightly better than other algorithms. It has the best Recall and F-score. KNN has higher precision than logistic regression but lower Recall and F-score. However, the results between different algorithms are very close. For example, XGBoost has the same recall as logistic regression and very close F score and precision. Because we are dealing with a real business world case, I also include prediction speed and interpretability as evaluation criteria. Logistic regression has the quickest prediction speed and easy to interpret.

Conclusion

After training the data with five different algorithms, the conclusion is that logistic regression using under-sampling performs the best in prediction in terms of its model performance, prediction speed, and interpretability. Credit card companies can even use this algorithm to assist credit card approval. The limitation of this project is that the dataset is limited as it only includes information for six months. Credit card companies should collect data from more customers for longer period to improve prediction accuracy.

Future work

There are a few more things that can be done in the future to potentially boost performance. First, more advanced models such as Support Vector Machine, Neural Network, and Gradient Boosting can be used for prediction. Second, to handle the imbalanced dataset, I use oversampling and under-sampling to solve the issue. Another sampling technique called as SMOTE can also be used. Third, more features can be created to help with prediction.

References

Archive.ics.uci.edu. 2020. *UCI Machine Learning Repository: Default Of Credit Card Clients Data Set*. [online] Available at: <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>

Brownlee, J., 2020. *A Gentle Introduction To Xgboost For Applied Machine Learning*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/>

Brownlee, J., 2020. *Hyperparameter Optimization With Random Search And Grid Search*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/>

Brownlee, J., 2020. *Ordinal And One-Hot Encodings For Categorical Data*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/>

Brownlee, J., 2020. *Random Oversampling And Undersampling For Imbalanced Classification*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/>

dummies. 2020. *Machine Learning: Creating Your Own Features In Data - Dummies*. [online] Available at: <https://www.dummies.com/programming/big-data/data-science/machine-learning-creating-features-data/>

En.wikipedia.org. 2020. *Data Binning*. [online] Available at: <https://en.wikipedia.org/wiki/Data_binning>

En.wikipedia.org. 2020. *Multicollinearity*. [online] Available at: <https://en.wikipedia.org/wiki/Multicollinearity#:~:text=Multicollinearity%20refers%20to%20a%20situation,equal%20to%201%20or%20%E2%88%921.>

Frost, J., 2020. *Guidelines For Removing And Handling Outliers In Data - Statistics By Jim*. [online] Statistics By Jim. Available at: <https://statisticsbyjim.com/basics/remove-outliers/>

Jeremy Jordan. 2020. *Learning From Imbalanced Data.*. [online] Available at: <https://www.jeremyjordan.me/imbalanced-data/>

Kaggle.com. 2020. *Bagging Vs Boosting*. [online] Available at: <https://www.kaggle.com/prashant111/bagging-vs-boosting>

Kumar, N., Kumar, N. and profile, V., 2020. *What Is Binning? What Is The Difference Between Fixed Width Binning And Adaptive Binning?*. [online] Theprofessionalspoint.blogspot.com. Available at: <http://theprofessionalspoint.blogspot.com/2019/03/what-is-binning-what-is-difference.html>

Medium. 2020. *Why, How And When To Scale Your Features*. [online] Available at: <https://medium.com/greyatom/why-how-and-when-to-scale-your-features-4b30ab09db5e>

Mlr.mlr-org.com. 2020. *Hyperparameter Tuning. — Tuneparams*. [online] Available at: <https://mlr.mlr-org.com/reference/tuneParams.html>

Sevenpillarsinstitute.org. 2020. [online] Available at: <https://sevenpillarsinstitute.org/case-studies/taiwans-credit-card-crisis/>

Appendix

Figure 1

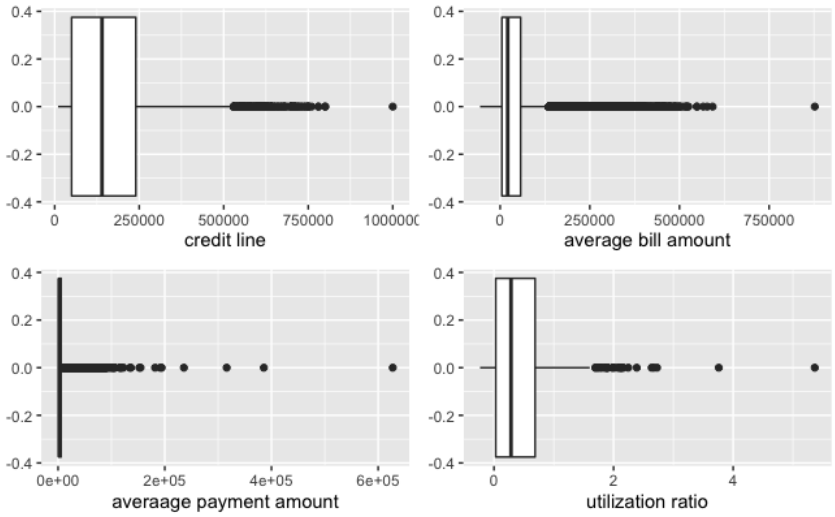

Figure 2

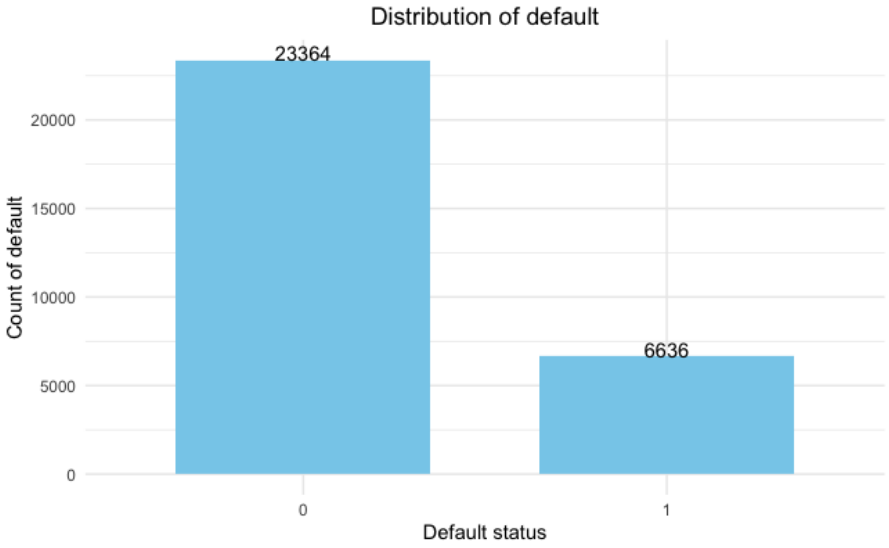

Figure 3

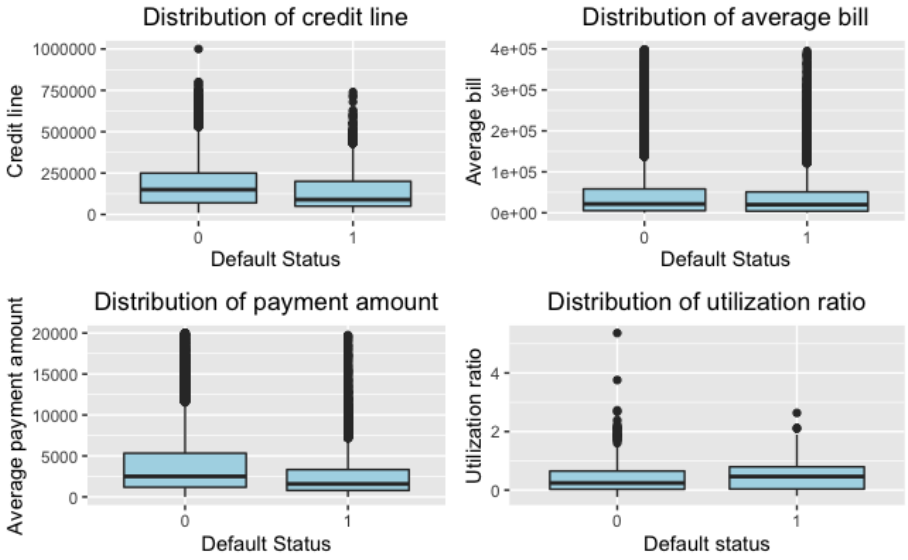

Figure 4

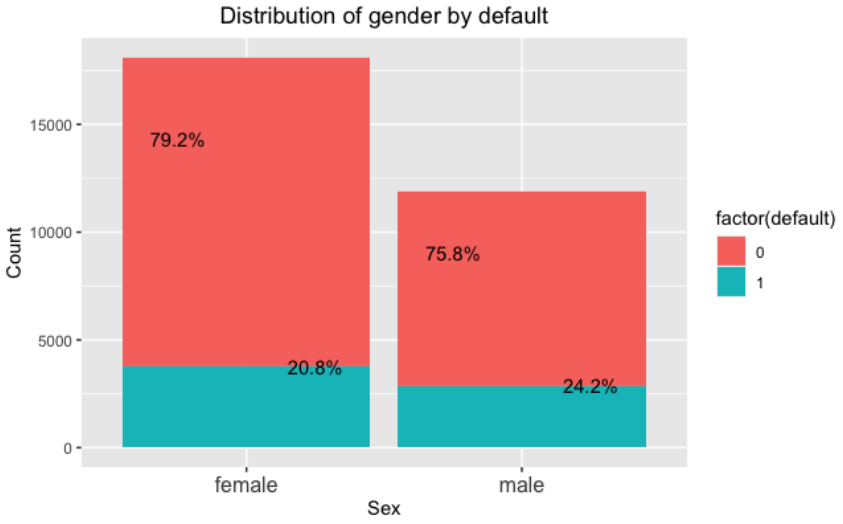

Figure 5

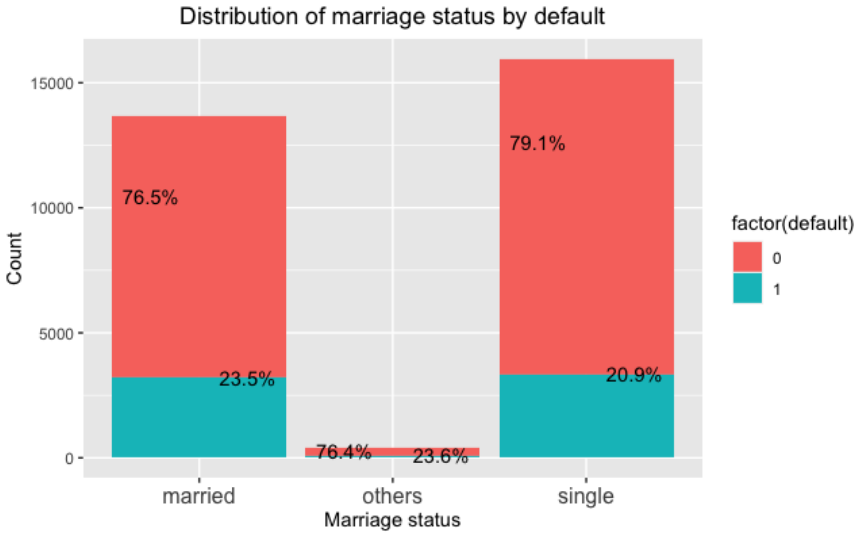

Figure 6

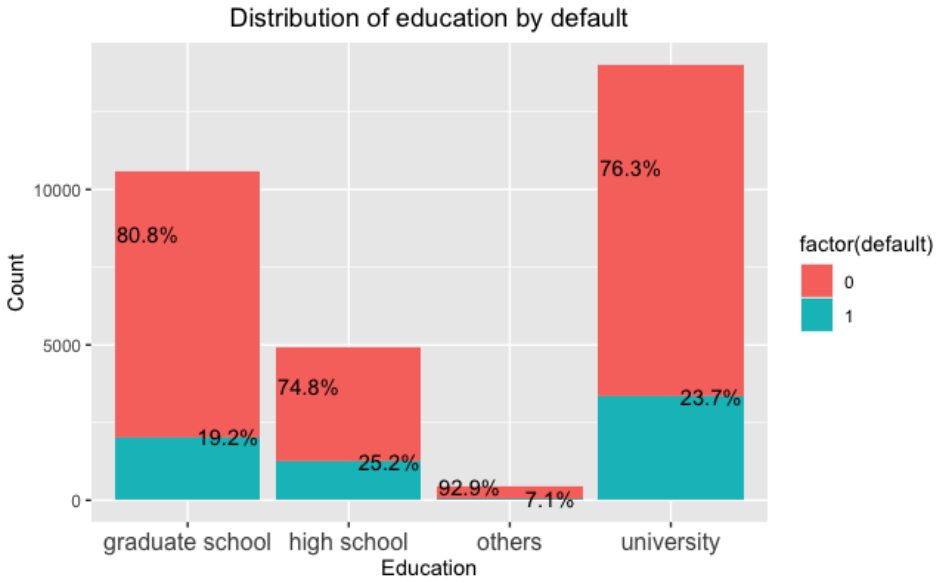

Figure 7

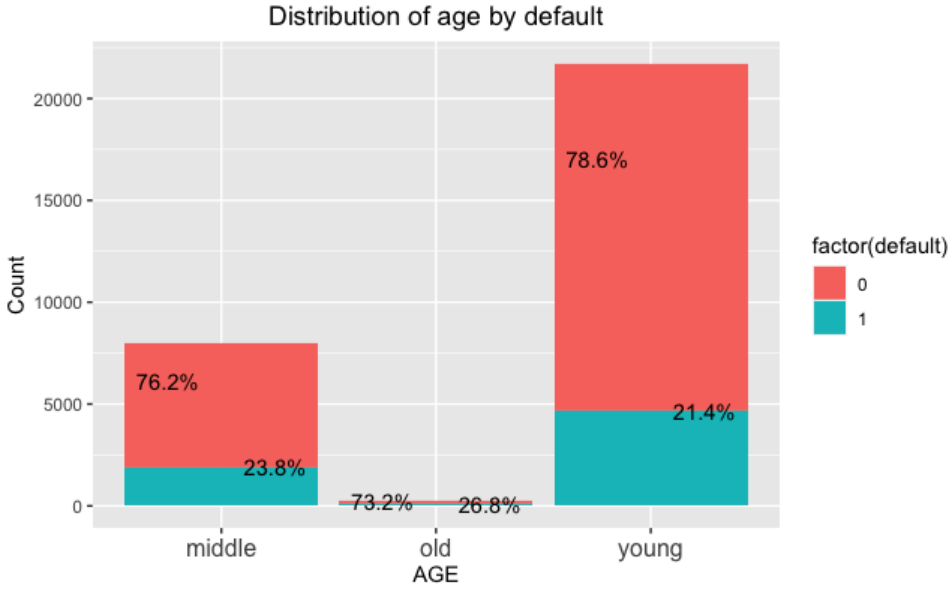

Figure 8

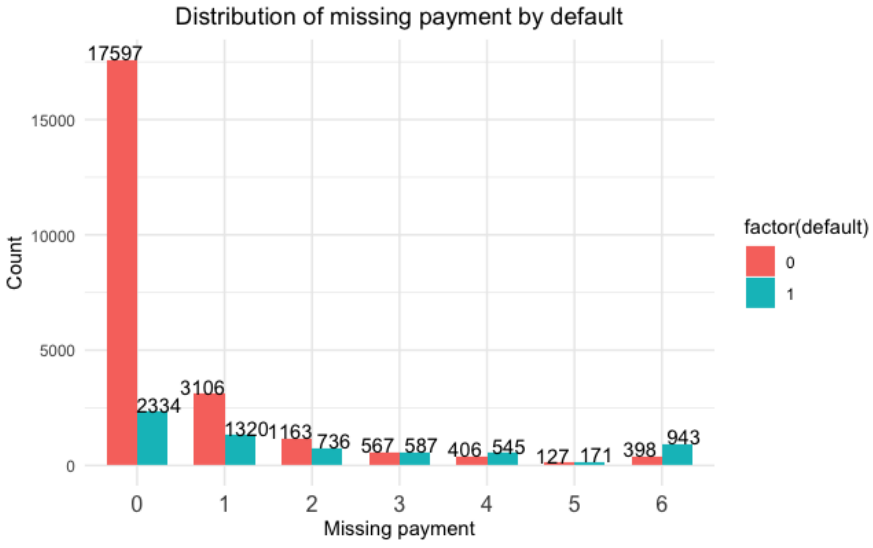

Figure 9

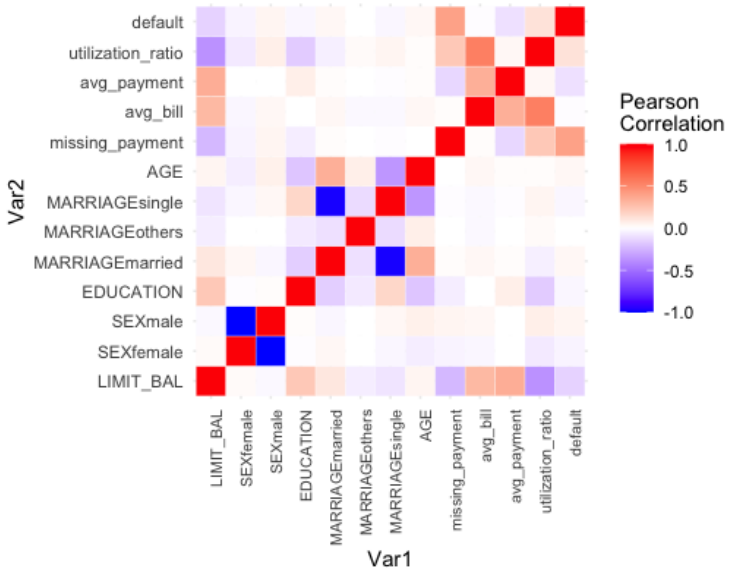

Figure 10

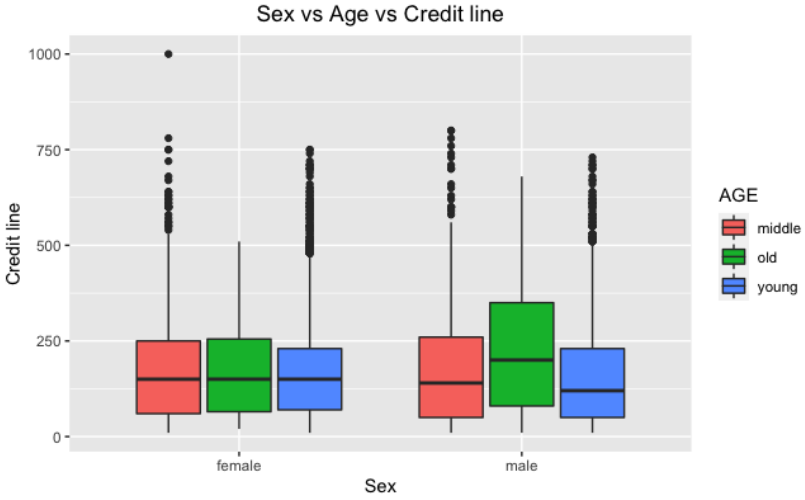

Figure 11

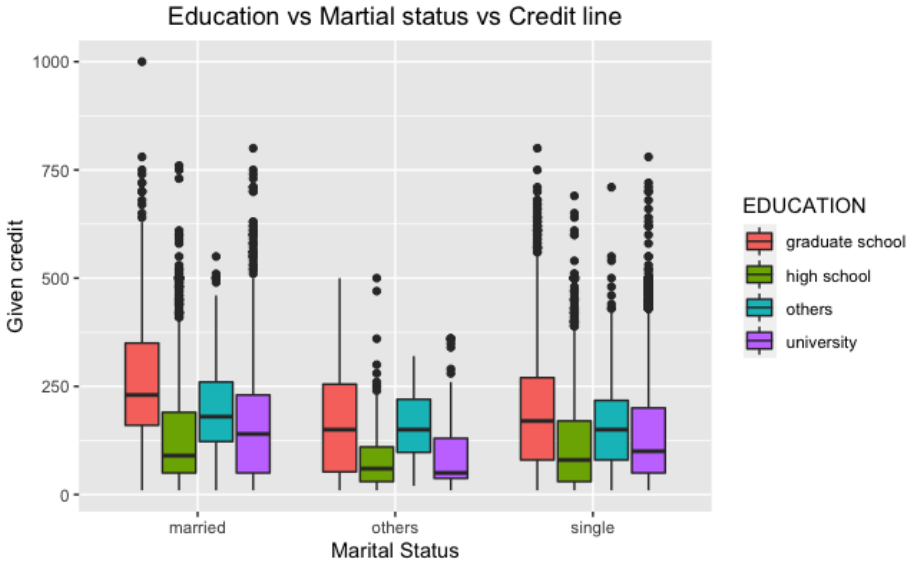

Table 1

Model Performance

Algorithms	Accuracy	Precision	Recall	F score	Prediction speed	Interpretability
Random Forest	0.71	0.64	0.69	0.66	0.124	Low
XGBoost	0.75	0.66	0.69	0.67	0.032	Low
Naïve Bayes	0.76	0.66	0.68	0.67	1.614	High
Logistic Regression (under)	0.78	0.68	0.69	0.68	0.012	High
KNN	0.80	0.70	0.62	0.66	1.777	High

Note:

1. Prediction speed will be slightly different every time it runs. But this doesn't change the conclusion.
2. The whole code will take around 5 minutes to run. Random forest tuning takes the longest.