IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 17, Issue 2 Ser. IV (Mar. – Apr. 2021), PP 45-51
www.iosrjournals.org

Statistical Learning for Analysis of Credit Risk Data

Li-Pang Chen
Fire Investigation Division, National Fire Agency, Ministry of the Interior, New Taipei City, 235, Taiwan (ROC)

Abstract:
In the financial sector, credit risk and financial modeling have been widely explored in practice, establishing particular scale characterization through pre-existing models and now the introduction of machine learning approaches. Our investigation is to generate a prediction model on a "Give Me Some Credit" dataset from Kaggle to help understand credit scoring and potential patterns of delinquency. Using various analytical models based on machine learning methods, risk levels of future credit loans are identified by accurately predicting the probability of an individual experiencing future financial distress. The results of data analysis in terms of the accuracy and the quality of the classifier are inspected through the ROC curve fitting. The ability to curate a precise model that can validate an individual's credit behaviour is further investigated in the report along with the insight of significant variables. Modelling an individual's credit score is imperative as the categorization is the initial and indicative impression of their financial responsibility.

Key Word*: Credit scoring; data analysis; financial distress; machine learning*

Date of Submission: 10-04-2021 Date of Acceptance: 26-04-2021

I. Introduction

In the past few weeks, world stock markets experienced a historical decrease. According to Yahoo Finance, S&P 500, DOW, and a series of stock indexes have dropped over 30%. It may be influenced by the wild spreading COVID-19, but it is also an inevitable trend that the ten-year bull market will turn into a recession. For the United States, the government has announced that they plan to use 2 trillion dollars as an emergency response for coronavirus. The policy looks like a strong stimulation for their market economy, but the consequence is predicted negatively, and it should be tested by time. Also, the restriction for banks is more severe due to the new Basel III reform. The 2007 to 2008 subprime crisis was mainly caused by the incorrect probability of default prediction. Under several factors, a bank needs to construct a business model to predict the probability of default with high accuracy.

In contemporary data analysis or era of data science, artificial intelligence and machine learning have been poplar and has attracted people's attention in recent years. They have also been widely applied to many fields, such as COVID-19 data [1], stock price data [2], drug development and healthcare as reviewed by [3].

Our goal aims to build such a model with the dataset "Give Me Some Credit". The source of our dataset is found on Kaggle. The dataset contains 11 variables. It includes information about each client. The variable "Serious Dlqin 2yrs" contains binary class "0" and "1" is considered as the response variable with "1" reflecting that customers are creditworthy and "0" otherwise. It is our interest and what we would like to predict for the test group. In addition, the remaining 10 features are taken as predictors, including "Revolving Utilization Of Unsecured Lines", "age", "Number Of Time 30-59 Days Past Due Not Worse", "Debt Ratio", "Monthly Income", "Number Of Open Credit Lines And Loans", "Number Of Times 90 Days Late", "Number Real Estate Loans Or Lines", "Number Of Time 60-89 Days Past Due Not Worse", and "Number Of Dependents". The dataset contains 150 thousand samples. Mathematically, let I denote the number of class, and $I=2$ is the case in the dataset. In addition, with sample size $n=150$, for each subject $i=1,..,n$, let Y_i denote the binary response and let X_i denote the predictors.

Our interest in this article belongs to classification problem, one of important topics in supervising learning. Many strategies to deal with classification have been developed in recent years, including discriminant analysis, support vector machine, and so on, as reviewed by [4] and [5]. In the past literature, several projects have analyzed the same dataset in Kaggle, such as "Modeling: Give Me Some Credit", and "Comp Stats Group Project – Final". Those two projects gave a good analysis of the dataset by using linear regression, random forest classification, cross-validation, etc. However, we find that they did not carefully consider the impacts of correlation for predictors, and they did not use some other methods to fit the model for further analysis. As a result, the purpose of this article is to predict the probability of default (PD) for the dataset with different methods and then compare these methods with further discussion. Therefore, we will split the dataset into a

training group and test group. The training group is used to construct a model and the test group is used to check the accuracy.

The structure of the project is as follows. In Section 2, we clean the missing values, useless and high correlated variables, and outliers. In Section 3, the uncorrelated predictors are analyzed using Weight of Evidence, and suggested predictors would be selected by their information value. In addition, different methods are introduced and then used to analyze the dataset, including logistic regressions, linear discriminant analysis, quadratic discriminant analysis, random forest, and support vector machine. In Section 4, data analysis is presented. Section 5 contains further discussion and conclusion.

II. Data Processing

Data preprocessing is a crucial step and is always the first thing needed to be done for data analysis. Filtering for the relative features for our response variable, "SeriousDiqin2yrs", by their definitions, we decided to keep all ten features. Since two predictors "NumberOfDpendents" and "MonthlyIncome" contain missing values in the original dataset, which can influence our future modeling results, we filled their missing values with a median. It is important to ensure the predictive variables used in the fitting model are uncorrelated to each other. We build a correlation matrix as shown in Figure1 and find the predictors "Number Of Time 60-89 Days Past Due Not Worse", "Number Of Times 90 Days Late" and "Number Of Time 30-59 Days Past Due Not Worse" have correlation 0.99 to each other, thus, we decide to remove the first two from the dataset and do not consider them in our models. Outliers can cause a model to be inefficient, so we check the distribution and cut the outliers or heavy tails for each predictor to make them distributed more normally.

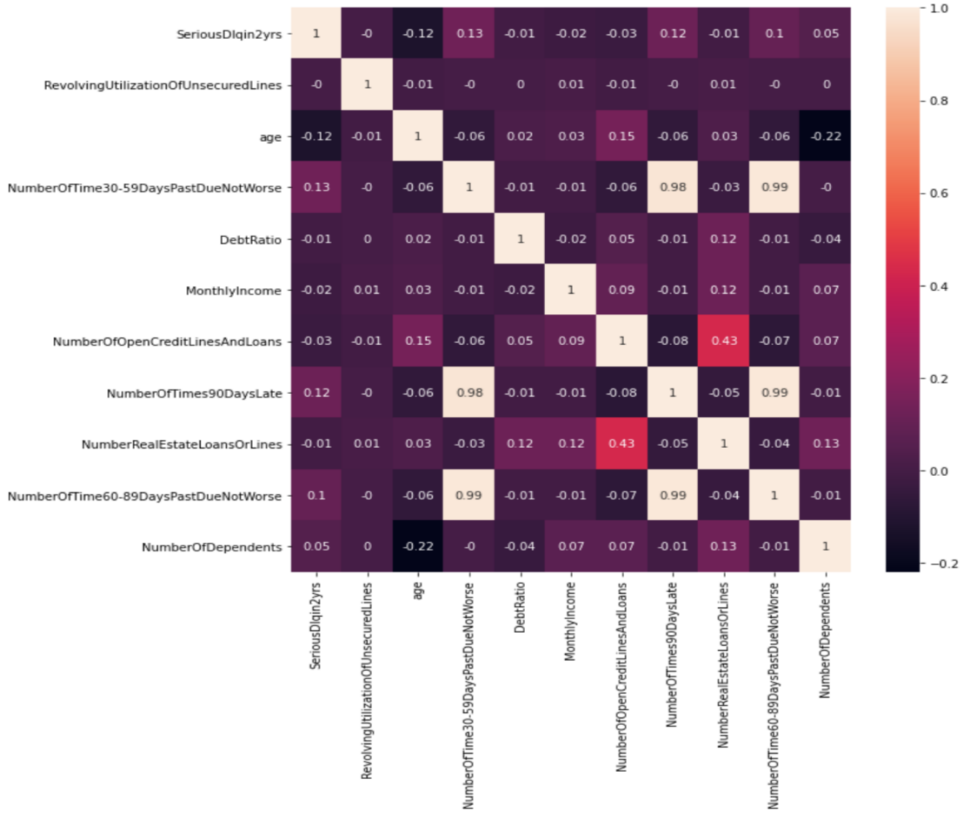


Figure1: Correlation matrix between different variables

III. Methodology

3.1 Weight of Evidence and Information Value

After determining eight predictors in Section 2, we next examine impacts and importance of those predictors, and then decide whether or not to exclude them. Our strategy is to apply information value (IV), which is a measurement of the prediction ability of the corresponding predictor. IV can be calculated by the weighted sum of Weight of Evidence (WOE), which can be interpreted as the predictive power of independent predictors in relation to the dependent predictor, and it is a form of encoding for the original independent variable.

The first step of calculating WOE is to properly bin the sample into M groups. Usually, one runs a tree and manually adjust those cases that do not follow a logical pattern. After binning the variables, for each group, WOE can be calculated by

$$WOE_m = ln\left(\frac{py_m}{pn_m}\right) = ln(\frac{\#\, y_i/\#n_i}{\#\, y_T/\#n_T}),$$

where for $m = 1, \dots M$, py_m is the ratio of the number of class 1 in bin m to all success trial in the samples, pn_m is the ratio of class0 in bin m to all failure in the samples, $\#y_m$ and $\#y_T$ are the number of class "1" in bin m and all samples, respectively, and $\#n_m$ and $\#n_T$ are number of class 0 in bin i and al sample, respectively.

From this formula, one can realize that WOE is the difference between the ratio of bad customers to all bad customers in each group and the ratio of good customers to all good customers in each group. The difference is expressed in terms of the ratio of these two ratios and then take the logarithm. The greater the WOE, the greater the difference is, which indicates the greater the probability of sample response in the group is. The smaller the WOE is, the smaller the difference is, and the smaller the possibility of sample response in this group is.

Based on WOE, IV is calculated by the weighted sum of WOE. Specifically, for each bin m,

$$IV_m = (py_m - pn_m) * WOE_m.$$

Then, the IV of the predictor is given by

$$IV = \sum_m^M IV_m.$$

By convention, the values of the IV statistic and the corresponding interpretation are summarized in Table 1. The ideal value of IV is within 0.1 to 1.

Information Value	**Variable Predictive**
IV < 0.02	No predictive ability, remove
0.02≤ *IV* < 0. 1	Small predictive ability, suggest to remove
0.1≤*IV* < 0.3	Medium predictive ability, leave
0.3≤*IV* < 1	Good predictive ability, leave
1≤IV	Strong predictive ability, Suspicious variable

Table 1: Rules related to Information Value

In addition, we implement the definition of IV to calculate eight predictors in our data, and the corresponding results are displayed in Figure 2. We observe that the remaining eight predictors have explainable trends so there are no need to do the bin adjustment for them. Moreover, all IV values of predictors are in 0.1 to 1, so we consider them all as significant predictors and then retain them in our analysis.

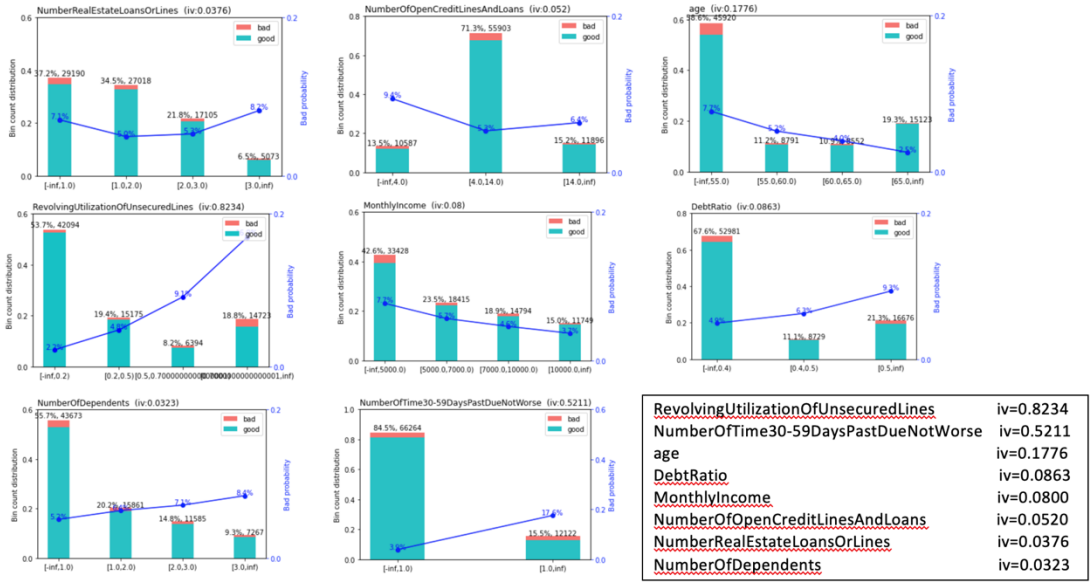


Figure 2: Information values for predictors

3.2 Logistic Regression

Logistic regression is one of the most frequently used approaches in Credit Scoring, which aims to predict the creditworthiness of a customer and determine whether they will be able to meet a given financial obligation or default on it. The basic assumption is that independent variables are required to linearly related to the log odds and dependent variable is required to be binary and ordinal. The model formulation is given by

$$\pi_i = \frac{\exp(\beta_0+\beta_1 x_1+\cdots+\beta_n x_n)}{1+\exp(\beta_0+\beta_1 x_1+\cdots+\beta_n x_n)}$$

where π_i is the probability of default for a subject i, x_k is the realization value of the explanatory factor k, and β_k is the regression coefficient associated with the predictor x_k. We use the logistic regression function in python with 1e-4 as the tolerance, 100 as the penalty to fit and predict the data.

3.3 Linear Discriminant Analysis (LDA)

Linear discriminant analysis is a commonly used technique in dimensionality reduction, which aims to find a connection between a categorical dependent variable and the linear combinations of multiple independent variables. It is assumed that independent variables are required to follow the multivariate normal distribution. Under this assumption and application of the Bayesian's Theorem, we have the estimated linear discriminant function:

$$\widehat{\delta_l}(x) = \log \widehat{\pi_l} - \frac{1}{2}\widehat{\mu_l}^T \widehat{\Sigma}^{-1}\widehat{\mu_l} + \widehat{\mu_l}^T \widehat{\Sigma}^{-1} x, \qquad i = 1, \dots, I$$

where $\widehat{\delta_l}(x)$ is the estimated discriminant score that the observation will fall in the kth class within the response variable based on the value of the predictor variable [6], $\widehat{\pi_l}$ denotes the estimated prior probability of class l with $\sum_{l=1}^{I} \widehat{\pi_l} = 1$. It can be estimated by $\widehat{\pi_l} = \frac{n_l}{n}$, where n_{il} is the number of observations in class l, $\widehat{\mu_l}$ is the empirical estimate of mean based on observations in class l, and $\hat{\Sigma}$ is the weighted average of the sample variance for class i, and $\hat{\Sigma} = \frac{1}{n-I}\sum_{l=1}^{I}\sum_{j=1}^{n_l}(x_j - \widehat{\mu_l})(x_j - \widehat{\mu_l})^T$.

When predicting a new subject with covariate X^*, we calculate $\widehat{\delta_l}(X^*)$ for $l = 1, \dots, I$, and then find l^* such that $l^* = \text{argmax } \widehat{\delta_l}(X^*)$, or equivalently, $\widehat{\delta_{l^*}}(X^*) = \max \widehat{\delta_l}(X^*)$. We fit the LDA model using the linear discriminant function in python with 1e-4 as the tolerance and use the LDA model to fit and predict the data.

3.4 Quadratic Discriminant Analysis (QDA)

Quadratic discriminant analysis is another classification method in discriminant analysis, which can be used when each class has its covariance matrix such that $\Sigma_i \neq \Sigma_j$ for $i \neq j$. The quadratic discriminant function is defined by

$$\widehat{\varphi_l}(x) = -\frac{1}{2}\log\left|\widehat{\Sigma_l}\right| - \frac{1}{2}(x-\widehat{\mu_l})^T \widehat{\Sigma_l}^{-1}(x-\widehat{\mu_l}) + \log \widehat{\pi_l}, \qquad i = 1, \dots, I$$

where $\widehat{\Sigma_l}$ is the weighted average of the sample variance for each class l, and $\widehat{\Sigma_l} = \frac{1}{n_l-1}\sum_{j=1}^{n_l}(x_j - \mu i\, xj - \mu iT$. When predicting a new observation with covariate $X*$, we calculate $\varphi iX*$ for $i = 1, \dots, I$, and then find i^* such that $l^* = \text{argmax } \widehat{\varphi_l}(X^*)$, or equivalently, $\widehat{\varphi_{l^*}}(X^*) = \max \widehat{\varphi_l}(X^*)$. We fitted the QDA model using the quadratic discriminant function in python and used the QDA model to fit and predict the data.

3.5 Support Vector Machine (SVM)

SVM is an algorithm that is capable of performing classification on a dataset. The basic idea of SVM is to find the optimal separation hyperplane that is defined as $w \cdot x + b = 0$, where w and b are unknown parameters and x is the predictor. It can map the given data into their labeled classes correctly and make sure they have the largest geometric interval.

Consider a binary classification problem with the dataset

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\},$$

where $y_i \in \{0,1\}$ for $i = 1, \dots \dots, n$. The algorithm of SVM with the non-linear kernel is as following:

- Each point (x_i, y_i) are mapped into a high dimension feature space by the kernel function $K(x,z)$ with penalty parameter C > 0 and solve

$$\min_{\alpha} \frac{1}{2}\sum_{i}^{n}\sum_{j}^{n}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i}^{n}\alpha_i$$

subject to $\sum_i^N \alpha_i y_i = 0$, where $0 \leq \alpha_i \leq C, i = 1,2, \dots, n$. Then we get the optimal value of $\alpha = (\alpha_1, \dots, \alpha_n)$, denoted as α^*.

- Given α_j^* such that $0 \leq \alpha_j \leq C$, we calculate the model bias by the formula

$$b^* = y_j - \sum_{i}^{N}\alpha_i^* y_i K(x_i, x_j)$$

- The decision function in the feature space is defined as follows

$$f(x) = sign\left(\sum_{i}^{N} \alpha_i^* y_i K(x_i, x_j) + b^*\right)$$

Note that the core idea of SVM is to set the complex nonlinear classification into another linear classification with the gradient in high dimensional space by using the kernel function, thus the choice of kernel function is important. There are some kernel functions are commonly used:

- Linear Kernel: $K(x, y) = X^T y + c$
- Polynomial Kernel: $K(x, y) = (\alpha X^T y + c)^d$
- Gaussian Kernel: $K(x, x') = \exp\{-\frac{\|x-x'\|^2}{2\sigma^2}\}$
- Radial basis function: $K(x, x') = -\frac{\|x-x'\|^2}{2\sigma^2}$

In this paper, we use Radial basis function as our kernel function.

3.6 Random Forest

Random forest is an ensemble learning method for classification that creates a certain number of uncorrelated decision trees and produces the mode of classes. It helps reduce the variance of an estimator and solves the overfitting issues.

Starting with a dataset with N cases and V variables, and initially, we need to choose some parameters:

- A sub-sample size m, commonly we could use $\frac{2}{3} \times N$ to be the sub-sample size.
- Let p be the number of variables in each tree, typically use $\frac{1}{3} \times V$.
- Let B denote the number of trees to train, rule: $10^{floor(\log(N)-2)}$.
- A minimum leaf size n_{min}, which should be between 0.1% - 1% of the dataset.

The algorithm of the Random forest is given below:

- Take a random sub-sample from the original dataset.
- Construct a decision tree (T_b), and stop when reaching the minimum leaf size n_{min}
- Repeat changing the random sub-samples until we obtain the number of B trees that we want to train
- Predict the result for each decision tree, and average the prediction of each decision tree
- Define the average prediction as $f(x) = \frac{1}{B}\sum_b T_b(x)$.

Under random forest classification, each decision tree can be treated as a random variable, with correlation ρ. Then the final variance can be denoted by:

$$\sigma_{RF}^2 = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

When the number of trees B increases, the fraction factor $\frac{1-\rho}{B}$ decreases, and the correlation between decision trees decreases. Using the random forest method achieves a better trade-off between variance reduction and predictive performance.

For our data, we use Random Forest Classifier in python to fit the model with resulting values summarized in Table 2:

Parameters	Value
Minimum samples in a leaf	0.001
Minimum impurity decrease	0.0001
Number of trees to train	1000
Minimum samples to create a split	2
Maximum number of nodes	None
Maximum depth of the trees	None

Table 2: Parameters of Random Forest Classifier

IV. Main Results and Data Analysis

After implementing methods in Section 3 to fit the data and then making predicted value, we first follow the definition of the confusion matrix in Table 3 to compute our results, which are summarized in Figure 3.

To further assess the performance of those methods and make fair comparisons with precise result and interpretation, we adopt some commonly used measures, such as accuracy, precision, recall, F-measure, and receiver operator characteristic (ROC) curves. First, accuracy is calculated as

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}.$$

Higher accuracy indicates better performance. It is obvious that the LDA (93%) gave the most accurate prediction, followed by the QDA (92%), while the SVM (47%) had the worst performance. However, for our test data, accuracy is not a convictive measure because of our asymmetric dataset. Since the false positive and false negative are very different, the calculation of accuracy is not accurate anymore.

Next, precision is calculated as

$$Precision = \frac{TP}{TP + FP},$$

which represents the ratio of the number of borrowers who are truly in the "Yes" class and predicted as "Yes" to the total number of borrowers predicted as "Yes". A lower precision, equivalent to a high 1-precision, means that there are more borrowers having good credit scores classified as "high-risk" customers, whose financial plan may not get good support from banks. If a bank uses such a credit model that has low precision measure for a long time, it will lose many trustable customers. Therefore, we want the precision to be high so that banks do not lose many "low-risk" borrowers. Among our five models, we notice that, in terms of precision, the LDA (34%) had the best performance, followed by the QDA (29%), while the SVM (8%) worked the worst.

Comparing with losing "low-risk" borrowers, banks usually take more care about giving finance to "high-risk" borrowers. Thus, we next look at the recall measure, calculated as

$$Recall = \frac{TP}{TP + FN},$$

which is a ratio of the number of borrowers who are truly in the "Yes" class and predicted as "Yes" to the total number of borrowers truly in the "Yes" class. A lower recall, equivalent to a high 1-recall, means that there are more borrowers having bad credit scores classified as "good" customers, who can, relatively, easily get finance from a bank. This is very dangerous for a bank to use such a model with low recall measure because it takes risks in terms of high probabilities of default among the borrowers. Therefore, we want the recall to be relatively high so that banks do not accept too many borrowers with low credit scores. Surprisingly, the LDA (10%), followed by QDA (22%), has the lowest recall measure, while the SVM (78%), logistic model (76%), and Random Forest(75%) have good performances.

Because the classes in the response are unevenly distributed, the F-measure usually gives a better overall evaluation for the models than the accuracy. The F-measure is defined as

$$F = 2 * \frac{Recall * Precision}{Rcall + Precision}.$$

Relatively, the QDA (25%), Random Forest (23%), and the logistic model (22%) overall perform better than the other two models.

ROC curve, commonly used in the machine learning field, can also evaluate the performances of binary classifiers. It is a graphical plot of the recall and the false positive rate, where

$$Flase\ positive\ rate = \frac{FP}{TN + FP},$$

A model has better performance if its ROC curve is closer to the top-left corner. Based on this rule, we find that all models have very similar good ROC curves except the SVM whose curve behaves like a baseline.

In order to compare the evaluations by the ROC curve more easily, we calculate the area under the curve (AUC). As a measure of predictive accuracy, it output a single measure instead of a curve. As we find from the results, those five models that have similar ROC curves also have very similar AUCs, however, the Random Forest (0.797) have the highest one, followed by the LDA (0.796).

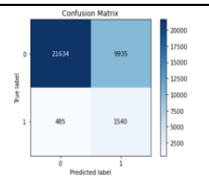
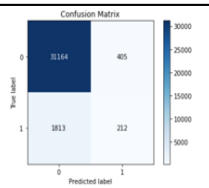
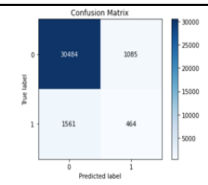
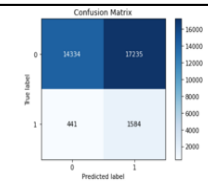
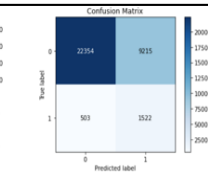
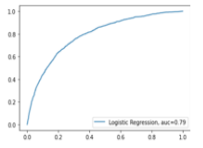
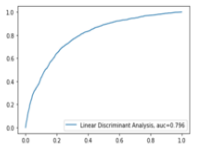
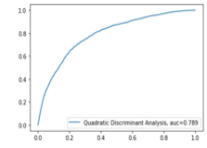
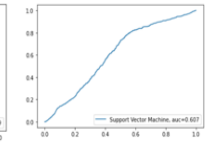
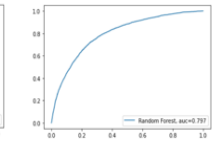
	Logistic Model	LDA	QDA	SVM	Random Forest
Confusion Matrix					
ROC Curve					
Accuracy	68%	93%	92%	47%	71%
Precision	13%	34%	29%	8%	14%
Recall	76%	10%	22%	78%	75%
F measure	22%	16%	25%	15%	23%

Figure 3: Performance of Classification Model

	Predicted Class		
True Class		SeriousDlqin2yrs =No	SeriousDlqin2yrs =Yes
	SeriousDlqin2yrs =No	TN	FP
	SeriousDlqin2yrs =Yes	FN	TP

TP = True Positive
FP = False Positive
FN = False Negative
TN = True Negative

Table 3: Definition of confusion matrix

V. Discussion

An individual's credit score indicates extensive detail on their ability to obtain goods or services before payment, primarily based on their financial history and the trust that the payment will be made in the future. In the scenario of requesting a loan, many predictors and significant characteristics are considered to help predict and anticipate the trust and performance of one's ability to repay the loan. Due to the extensive process of reviewing an individual's credit history and personal portfolio, it is important to impose credit score. Understanding the importance of credit score can now fuel the investigation of the model that optimizes the accuracy and significant predictors to evaluating one's personal portfolio when assessing a truthful credit score.

In this article, different methods are used to fit the credit risk data. The results show that every model we fit in this article is not perfect, and have both pros and cons. Analyzing from different perspectives, we suggest financial institutions focus more on saving their "low-risk" customers using the LDA, and those focus more on filtering "high-risk" customers using the SVM. However, Random Forest have the best overall performance among these models because its F-measure and AUC score were both great.

There are few things that can be improved in the future with respect to the accuracy of the model. For example, Logit regression is not good at presenting the non-linearity and interaction among features, which is due to its dependence on linearity and monotone relationship. Furthermore, logit models are vulnerable to overconfidence. Therefore, the overfitting of the Logit regression model mispresents the actual prediction. Thus, it would be a better choice to adopt tree-based algorithm that involves classification, which solves the problem of non-linearity and is also appropriate for analyzing categorical variables.

References

[1]. Chen, L.-P., Zhang, Q., Yi, G. Y., and He, W. (2021). Model-based forecasting for Canadian COVID-19 data. *PLOS ONE*, 16(1): e0244536. DOI: 10.1371/journal.pone.0244536.

[2]. Chen, L.-P. (2020). Using machine learning algorithms on prediction of stock price. *Journal of Modeling and Optimization*, 12, 84-99. DOI: 10.32732/jmo.2020.12.2.84

[3]. Chen, L.-P. (2020). Artificial Intelligence for Drug Development, Precision Medicine, and Healthcare by Mark Chang. *Biometrics*, 76, 1392-1394. DOI: 10.1111/biom.13390

[4]. Chen, L.-P. (2020), Model-based clustering and classification for data science: With applications in R Bouveyron, Charles Celeux, Gilles Murphy, T. Brendan Raftery, Adrian E. (2019). New York, NY: Cambridge University Press. 446 pages. CDN$91.95 (hardback). ISBN: 9781108494205. *Biometrical Journal*, 62: 1120-1121. DOI: 10.1002/bimj.201900390

[5]. Chen L.-P. (2019). Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar: Foundations of machine learning, second edition. *Statistical Papers*, 60, 1793–1795. DOI: 10.1007/s00362-019-01124-9

[6]. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An introduction to statistical learning: with applications in R*. New York: Springer.

Chen, L.-P. "Statistical Learning for Analysis of Credit Risk Data." *IOSR Journal of Mathematics (IOSR-JM)*, 17(2), (2021): pp. 45-51.