Credit Card Default Prediction: A Comparative Study Of Machine Learning Models Based On Accuracy, Sensitivity, And Specificity

Szu-Hsien Lin[1] , Trang Nguyen[2] ,
Huei-Hwa Lai[*3] , Mei Hua Huang[4]

[1]Associate Professor, Department of Accounting and Information Systems, Asia University, Taiwan
aleclin.tw@gmail.com

[2]Department of Finance, Asia University, Taiwan
baotranghsgs@gmail.com

[3]Assistant Professor, Department of Business and Administration, ChaoYang University of Technology, Taiwan
edithlai2005@gm.cyut.edu.tw

[4]Assistant Professor, Department of Accounting and Information Systems, Asia University, Taiwan
meihuang@asia.edu.tw

Abstract

Objective: This analysis aimed to compare three machine learning models—Logistic Regression, Naive Bayes, and Linear Discriminant Analysis (LDA)—for their ability to predict credit card default.

Methods: Each model's performance was evaluated based on accuracy, sensitivity, and specificity metrics using a dataset of credit card holders.

Results: All three models demonstrated similar accuracy levels, between 0.969 and 0.9715, indicating a good ability to correctly classify cases overall. In terms of sensitivity, or the ability to correctly identify non-default accounts, all models achieved high scores (0.9953 to 0.9979). However, there were differences in the specificity, or the ability to correctly identify default accounts. The Logistic Regression model showed a higher specificity (0.2754) compared to the Naive Bayes and LDA models (both 0.2319), suggesting a better performance in identifying default accounts.

Implications: While all three models showed high accuracy and sensitivity, the Logistic Regression model outperformed in terms of specificity, making it the preferred model for this task. However, all models exhibited relatively weak performance in identifying default accounts, indicating a potential need for further optimization, consideration of other metrics, or different modeling approaches, especially given the high cost associated with misclassifying defaulting accounts. However, all models exhibited relatively weak performance in identifying default accounts. This is likely due to the imbalanced nature of the data. Therefore, different modeling approaches or techniques to handle imbalanced data might be necessary to improve the identification of defaults.

Keywords: Credit Default Prediction; Machine Learning; Logistic Regression; Naive Bayes; Linear Discriminant Analysis

Introduction

Predicting credit defaults is an essential task for financial institutions worldwide. The rise in the availability of financial products and services and the growing number of credit users necessitates that banks and credit card companies have precise models to accurately forecast credit defaults. Doing so helps them mitigate risk and maintain financial stability. For decades, traditional models such as Logistic Regression and other statistical techniques have served as the cornerstone for predicting credit defaults (Hand & Henley, 1997). These models, while robust, often grapple with high-dimensional, non-linear data, a common scenario in the current era of big data.

The advent of machine learning (ML) techniques has opened a new chapter in credit risk modeling. Machine learning models, such as Naive Bayes and Linear Discriminant Analysis, have demonstrated promising results in various financial prediction tasks (Huang et al., 2004). These models, characterized by their flexibility, can adeptly handle complex data patterns, thereby providing more accurate and reliable predictions than traditional statistical methods. However, the application of these advanced models is not without challenges. One of the primary issues they confront is handling imbalanced datasets, a prevalent problem in credit default prediction (Chawla et al., 2002).

In this study, we delve into the application of different machine learning techniques - Logistic Regression, Naive Bayes, and Linear Discriminant Analysis - to predict credit defaults using virtual data. We follow the methodology proposed by Lessmann et al. (2015) and evaluate the performance of these models in terms of accuracy, sensitivity, and specificity.

Using virtual data for our study allows us to assess the effectiveness of these machine learning techniques under a variety of conditions, without compromising privacy and confidentiality - an important consideration when dealing with sensitive financial information. Although our study employs virtual data, the methods and techniques used can be easily applied to proprietary business data, making the insights gained from our research directly applicable to real-world business scenarios.

By offering a comprehensive understanding of the performance of different machine learning techniques in credit default prediction and providing valuable insights into how to effectively manage the issue of imbalanced datasets, we aim to make a significant contribution to the existing body of literature. Our findings will be of practical use to businesses seeking to enhance their credit risk modeling techniques, thus bridging the gap between academia and industry.

Data

5.1 Data source

“Default” dataset is a simulated dataset commonly used in research and educational settings to explore credit card default behavior. This dataset is a valuable resource for studying the factors that contribute to credit card defaults and developing predictive models to aid in risk assessment and decision-making in the financial domain. In this section, a brief description of the Default dataset will be provided, including the sources of data, and variables explanation.

The dataset is secondary data originated from R program, which is typically sourced from the "ISLR" package (Introduction to Statistical Learning with Applications in R). This package provides datasets and functions that accompany the textbook "An Introduction to Statistical Learning" by James, Witten, Hastie, and Tibshirani.

The dataset's simulated nature allows researchers to explore the relationship between the predictor variables and the likelihood of default in a controlled environment. In our research, we utilized the Default dataset to investigate the factors contributing to credit card default. By analyzing this dataset, we aimed to uncover patterns and relationships that can inform risk assessment strategies in the financial industry. The dataset's characteristics, including both binary and numeric variables, make it suitable for building predictive models using techniques such as logistic regression.

To ensure the validity of our results, we conducted appropriate data preprocessing steps, such as checking for missing values, performing exploratory data analysis, and addressing potential outliers by R program. We also split the dataset into training and testing parts to evaluate the performance of our predictive models.

By utilizing the Default dataset, we contribute to the existing knowledge base surrounding credit card default behavior and provide insights that can assist financial institutions in making informed decisions regarding risk management and customer creditworthiness assessment.

5.2 Data description

Table 1: Variable explanation

Order	**Variable Name**	**Type**	**Label Description**
1	Default	Binary	A factor with levels No and Yes indicating whether the customer defaulted on their debt
2	Student	Binary	A factor with levels No and Yes indicating whether the customer is a student
3	Balance	Numerical	The average balance that the customer has remaining on their credit card after making their monthly payment
4	Income	Numerical	Income of customer

The Default dataset consists of a collection of 10,000 observations on credit card holders and their default payment status. It includes several variables that provide insights into the characteristics and financial behavior of the credit card holders. The main variable of interest is "Default," which represents the binary response variable indicating whether a customer defaulted on their credit card payment. It takes on two values: "No" indicating no default, and "Yes" indicating a default. In addition to the Default variable, the dataset includes several predictor variables that can potentially influence the likelihood of default. These variables provide additional information about the customers, such as whether they are students ("Student"), the outstanding balance on their credit card ("Balance"), and their annual income ("Income").

Research Methodology

3.1 Model construction

3.1.1 Logistic Regression

Logistic regression is a statistical modeling technique employed to investigate the relation between a dependent variable and one or more independent variables. It is primarily employed when the dependent variable represents a categorical outcome with two possible outcomes, such as "yes" or "no".

The goal of logistic regression is to estimate the probability of the occurrence of a particular outcome based on the values of the independent variables. It utilizes a logistic function, also known as the sigmoid function, to transform the linear combination of the independent variables into a probability value between 0 and 1. In our case, the predictors of the model such as student, balance, and income are used to predict the probability of default. If predicted value is larger than 0.5, meaning that it is nearer to 1, it will be defined as “yes”, otherwise the predicted default result is anticipated as “no”.

Logistic regression makes use of maximum likelihood estimation to determine the optimal coefficients that define the relationship between the independent variables and the probability of the outcome. These coefficients are typically interpreted as the log-odds or odds ratios, which indicate the impact of each independent variable on the likelihood of the outcome occurring.

We have the theorical framework:

In logistic regression, the dependent variable is modeled as a binary variable, typically denoted as "y," where y = 1 represents the occurrence of the event of interest (e.g., success) and y = 0 represents the absence of the event. The logistic function is defined as follows:

$$p = \frac{1}{1+e^{-z}}$$

where: p represents the probability of the event of interest (y = 1); e is the base of the natural logarithm (approximately 2.71828), and z is the linear function of the predictors.

The linear function of the predictors is calculated as:

$$z = \log\frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$$

where: β_0, β_1, β_2, ..., β_p are the coefficients (also known as weights) associated with each independent variable; x_1, x_2, ..., x_p are the values of the independent variables; and p represents the number of independent variables (predictors) in the model.

The coefficients (β_0, β_1, β_2, ..., β_p) in logistic regression are estimated using maximum likelihood estimation. The estimation process aims to find the values of the coefficients that maximize the likelihood of observing the given set of outcomes. Once the coefficients are estimated, they can be used to predict the probability of the event of interest for new observations by plugging in the corresponding values of the independent variables into the logistic function.

Based on the framework, we apply the model into our case, and have the model:

$$\log\frac{\text{Default}}{1-\text{Default}} = \beta_0 + \beta_1\text{*Student} + \beta_2\text{*Balance} + \beta_3\text{*Income}$$

3.1.2 Linear Discriminant Analysis (LDA)

Another model for classification tasks is Linear Discriminant Analysis (LDA), which both help simplify the process and improve computational efficiency. With the technique of dimensionality reduction, LDA aims to reduce the dimensionality of the input feature space while preserving the discriminatory information between different classes, maximizing the separation among them.

The theoretical framework of LDA is based on Bayes' theorem and assumes that the data follows a Gaussian distribution. The key idea is to model the distribution of each class and then use this information to determine the optimal linear discriminant function.

In LDA, the separation between classes is quantified using the Separation formula, defined as the ratio of the between-group variance to the within-group variance. Symbolically, this can be represented as **S = W^{-1} * B**, where W denotes the within-group variance and B represents the between-group variance. By calculating the Separation Matrix, LDA identifies the eigenvectors and eigenvalues, which indicate the directions of maximum separation and the importance of these directions, respectively. Based on the eigenvectors, a model is constructed to compute the discriminant scores. These scores are then normalized to obtain classification values that range from 0 to 1. A higher score indicates a higher likelihood of default. This normalization process ensures that the scores are interpretable and can be compared across different observations.

Through its theoretical framework, LDA facilitates dimensionality reduction and optimal class separation, resulting in a simplified and efficient classification process. By identifying the most discriminative directions, LDA provides valuable insights into the factors driving classification outcomes.

3.1.3 Naïve Bayes

Naïve Bayes is a classification algorithm that utilizes Bayes' theorem to estimate the probability of class labels based on observed features. It assumes that the features are independent of each other. The theoretical framework of Naive Bayes revolves around calculating the conditional probability of a class label given a specific combination of feature values.

The theorical framework of Naïve Bayes model:

$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)}$$

Where: P(y|X) represents the posterior likelihood of class y given the features X; P(X|y) stands for the likelihood probability of features X given class y; P(y) denotes the prior likelihood of class y; and P(X) signifies the probability of features X.

In our scenario, the classification problem involves two distinct classes: default and non-default. We consider three features, namely student, balance, and income. The key assumption made is that these features are independent of each other. By calculating the probabilities of default and non-default for each observation, we can determine the class label. If the probability of default for a given observation exceeds the probability of non-default, the observation is classified as default, and vice versa.

3.2 Model Evaluation

When evaluating a model's performance in binary classification tasks, accuracy, specificity, and sensitivity are commonly used measures. Accuracy represents the overall correctness of the model's predictions and is calculated as the ratio of correct predictions to the total number of predictions. The formula for accuracy is:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Here, TP (True Positives) represents the number of correctly predicted positive instances, TN (True Negatives) represents the number of correctly predicted negative instances, FP (False Positives) represents the number of incorrectly predicted positive instances, and FN (False Negatives) represents the number of incorrectly predicted negative instances.

Specificity, also known as the true negative rate, evaluates the model's ability to correctly identify negative instances. It is calculated as the ratio of true negatives to the sum of true negatives and false positives:

$$\text{Specificity} = TN / (TN + FP)$$

Sensitivity, also known as the true positive rate or recall, measures the model's ability to correctly identify positive instances. It is calculated as the ratio of true positives to the sum of true positives and false negatives:

$$\text{Sensitivity} = TP / (TP + FN)$$

Accuracy provides an overall measure of the model's correctness, while specificity and sensitivity focus on the model's performance for negative and positive instances, respectively. Specificity helps in scenarios where correctly identifying negative instances is crucial, while sensitivity is important when correctly capturing positive instances is of utmost importance.

To evaluate a model using these metrics, the dataset is split into two parts as training and testing. The model is trained on the training data, and its predictions are compared to the true outcomes in the testing data. The accuracy, specificity, and sensitivity values can then be calculated based on the predicted and observed outcomes using the formulas.

By considering accuracy, specificity, and sensitivity, analysts gain a comprehensive understanding of a model's performance in binary classification tasks. These metrics aid in making informed decisions and adjusting the model, if necessary, ultimately improving its effectiveness and applicability in real-world scenarios.

By considering accuracy, specificity, and sensitivity, analysts gain a comprehensive understanding of a model's performance in binary classification tasks. These metrics aid in making informed decisions and adjusting the model, if necessary, ultimately improving its effectiveness and applicability in real-world scenarios.

Empirical Result

4.1 Descriptive Statistics

4.2.1 Numerical Variables

Table 2: Descriptive Statistics of Balance and Income

Vars	N	Mean	Sd	Median	Min	Max	Range	Skew	Kurtosis	SE
Balance	10000	835.37	483.71	823.64	0	2654.32	2654.32	0.25	-0.36	4.84
Income	10000	33516.98	13336.64	34552.65	771.97	73554.23	72782.27	0.07	-0.9	133.37

The provided descriptive statistics table offers valuable insights into the characteristics of the data, focusing on two numerical variables: Balance and Income.

Regarding Balance variable, we observe that the mean balance across 10,000 accounts is approximately $835.37, while the median balance stands at $823.64. The range of balances spans from 0 to $2,654.32, indicating a considerable variation in account balances. The standard deviation of 483.72 quantifies the extent of spread or dispersion in the data. Moreover, the standard error of balance, which is 4.84, suggests that the population mean could deviate by approximately 4.84 units from the sample mean. The skewness of 0.25, close to 0, indicates that the distribution of balance values is nearly symmetrical. Furthermore, a kurtosis value of -0.36 suggests that the tail of the balance distribution is lighter than that of a normal distribution (where kurtosis is 3).

In terms of Income variable, we find that customer incomes range from $772 to $73,554, with an average income of approximately $33,517. The standard deviation of 13,337 reflects the significant variability in income levels among customers. Similar to the Balance variable, the Income variable also exhibits a near-symmetrical distribution with a skewness close to 0. Additionally, the kurtosis value of the income distribution (-0.36) indicates that its tail is lighter than that of a normal distribution.

The boxplot visualization clearly illustrates a substantial and significant difference in balance between default and non-default accounts. When comparing the mean balance, non-default customers exhibit a range of approximately 700-800, whereas default customers have a notably higher mean balance of nearly 1800. Additionally, the interquartile range, which represents the spread of the middle 50% of the data, further emphasizes the contrast. For non-default accounts, the interquartile range spans approximately 500 to around 1200, indicating a relatively moderate dispersion of balances. In contrast, default accounts display a wider interquartile range of 1500-2000, signifying a more substantial spread of balances. Therefore, it can be confidently concluded that default customers possess larger balances compared to their non-default counterparts, as evident from the distinct patterns showcased in the boxplot. This insightful analysis of the balance discrepancy between default and non-default customers provides valuable information for understanding the relationship between balance and default status. These findings highlight the potential significance of balance as a distinguishing factor in predicting and assessing the default risk associated with different accounts.

Figure 1: Describe balance variable

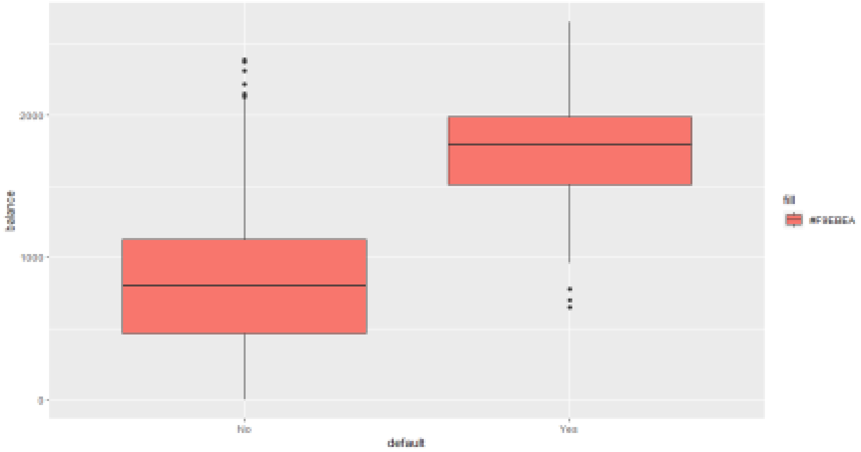
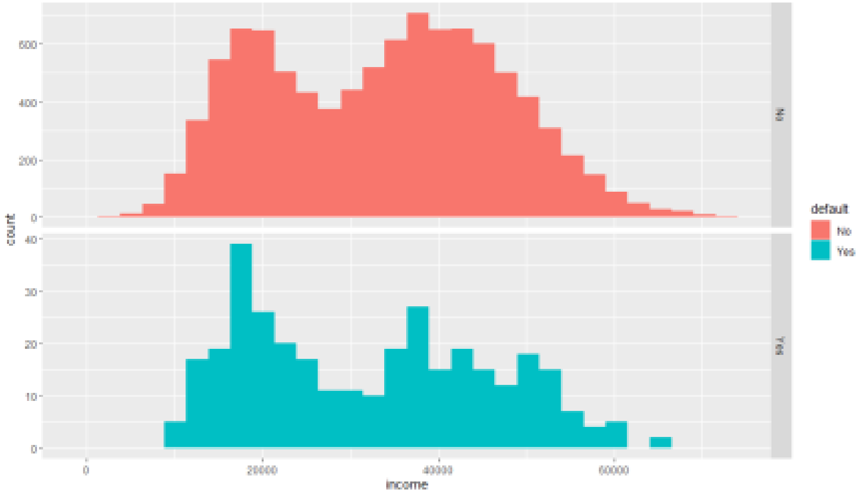


Figure 2: The histogram of income between default and non-default accounts

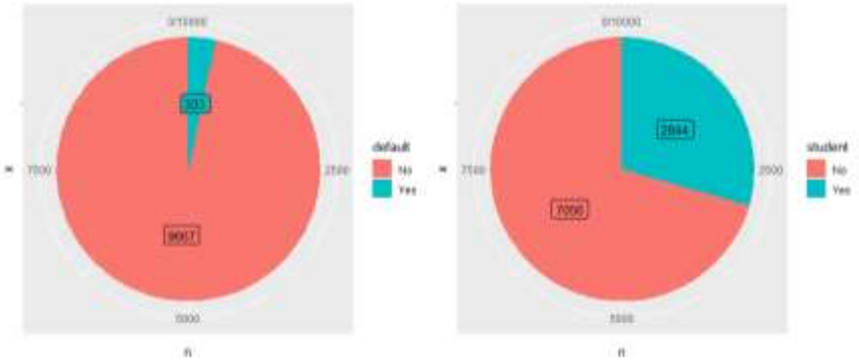


The histograms provide visual representations of the income distribution among customers. Both charts exhibit a similar pattern, characterized by two peaks occurring at around 20000 and 40000. In the histogram representing default accounts, the income range around 30000 constitutes only one-fourth of the area compared to that around 20000.

Similarly, in the histogram for non-default accounts, the corresponding income range also accounts for only two-thirds of the area in comparison. As a result, it can be concluded that there is no distinct pattern observed in the histograms that clearly differentiates default and non-default accounts. Therefore, further testing and analysis are necessary in the model phase to gain deeper insights and determine whether income plays a significant role in predicting default status. These histograms serve as a starting point for exploration, prompting the need for more advanced techniques to uncover potential relationships and dependencies between income and default status within the dataset.

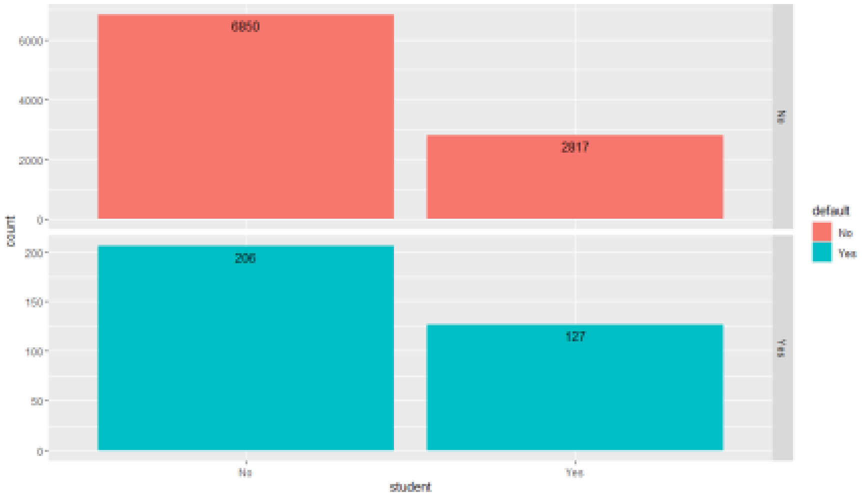
4.2.2 Binary Variables

Figure 3: The proportions of Default and Student variables



As can be seen in the graph, the number of default accounts is 333, which accounts for 3% of the total accounts. Besides, among all customers, there are 2944 students and 7056 of customers are not students.

Figure 4: The histogram of income between default and non-default accounts

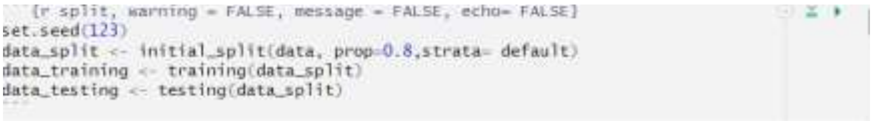


In terms of default customers, there are 127 students, making up approximately 38.14% of the total. Regarding non-default customers, out of 9667 customers, 2817 are students, accounting for approximately 29.14% of the total.

4.2 Models' Result

To build the model, we split the data into 2 parts in a proportion of 80/20, with 80% of data as training data and 20% are testing data. Splitting data is a fundamental step in machine learning and data analysis, allowing us to effectively evaluate and train models. The process involves dividing the available dataset into two or more subsets: a training set and a testing/validation set. The training set is used to train the model, enabling it to recognize patterns and relationships within the data. The testing/validation set, on the other hand, is utilized to assess the model's performance and generalization capabilities on unseen data. By splitting the data, we can simulate real-world scenarios and evaluate how well the model performs on new, unseen examples. It helps in identifying potential issues such as overfitting or underfitting.

Figure 5: How the data is split in R-program



~~~~
```{r split, warning = FALSE, message = FALSE, echo= FALSE}
set.seed(123)
data_split <- initial_split(data, prop=0.8,strata= default)
data_training <- training(data_split)
data_testing <- testing(data_split)
```
~~~~

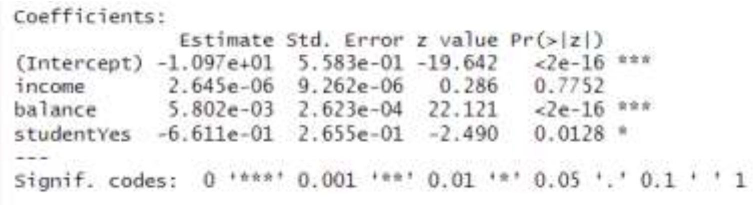
Step 1: the function set.seed() is used to set the starting point for generating random numbers. When a seed value is set using set.seed(), it ensures that the sequence of random numbers generated remains the same each time the code is run with the same seed value. This can be useful for replicating results or creating reproducible analyses. By setting a seed value, controlling the randomness in functions is controlled that involve random number generation, such as sampling, permutation, or simulation. This allows the model to obtain consistent results when working with random processes.

Step 2: The data is splitting into a proportion of 80/20 with a stratified sampling by default data. Stratified sampling is a method that ensures the representation of different classes or categories in the data is maintained proportionally across both subsets. By employing stratified sampling, the resulting training and testing sets reflect the overall distribution and characteristics of the original data, enhancing the generalizability and accuracy of the subsequent analyses or modeling processes.

4.2.1 Logistic Regression

We apply the logistic model above into the model by glm function in R:

Figure 6: Coefficient of logistic regression model



```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -1.097e+01  5.583e-01 -19.642   <2e-16 ***
income       2.645e-06  9.262e-06   0.286   0.7752    
balance      5.802e-03  2.623e-04  22.121   <2e-16 ***
studentYes  -6.611e-01  2.655e-01  -2.490   0.0128 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the result, we have the model:

Log $(\frac{\text{p(Default)}}{1-\text{p(Default)}})$ = -10.97+ 0.00000026*Income +0.0058 * Balance - 0.0661 * Student

It is important to note that the variable "Student" is a categorical variable represented by a dummy variable, where "Yes" is coded as 1 and "No" is coded as 0.

Moving on to the Income variable, the coefficient is determined to be 0.00000026, indicating a positive relationship between Income and log odds of Default's probability. Specifically, for every 1 unit increase in Income, the log odds for probability of default also increases by approximately 0.000000026 units. This suggests that higher income levels are associated with a slightly higher likelihood of default.

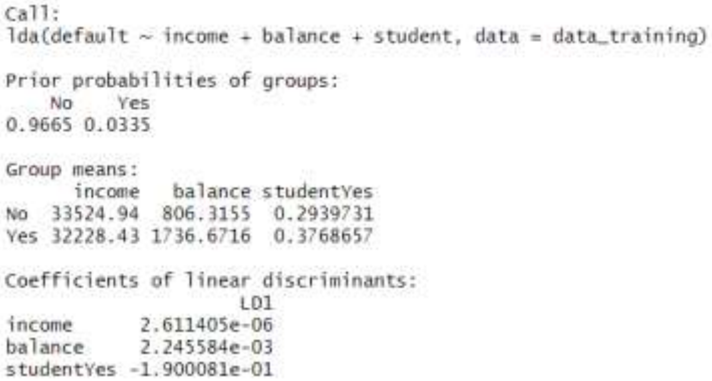
In the case of the Balance variable, the coefficient is 0.0058. This implies that for every 1 unit increase in the customers' Balance, the log odds for probability of default also increase by 0.0058 units. This coefficient indicates a stronger impact of Balance on the probability of default compared to the Income variable.

Regarding the Student variable, the coefficient is -0.0661. This negative coefficient suggests that customers who are students have a log odd for probability of default that is approximately 0.0661 units lower than customers who are not students. In other words, being a student is associated with a reduced likelihood of default, highlighting the potential influence of student status as a protective factor against default.

These coefficients provide insights into the direction and magnitude of the relationships between the variables and the probability of default. By understanding the impact of each variable, we can gain a deeper understanding of the factors contributing to default risk and make more informed decisions or interventions to manage and mitigate such risks.

4.2.2 Linear Discriminant Analysis (LDA)

Figure 7: Result of LDA model



```
Call:
lda(default ~ income + balance + student, data = data_training)

Prior probabilities of groups:
    No    Yes 
0.9665 0.0335 

Group means:
      income   balance studentYes
No  33524.94  806.3155  0.2939731
Yes 32228.43 1736.6716  0.3768657

Coefficients of linear discriminants:
                     LD1
income      2.611405e-06
balance     2.245584e-03
studentYes -1.900081e-01
```

First of all, the given result illustrates prior probabilities of the two groups, "No" (non-default) and "Yes" (default). While the proportion of non-default accounts is reported as 0.9665, the default accounts are a minority, accounting for only 3.35%.

The groups' means provide insights into the average values of the predictor variables for each group. In the non-default group, the mean income is approximately 33524.94, the mean balance is around 806.32, and the proportion of students is approximately 0.294. In terms of default accounts, the mean income is approximately 32228.43, the mean balance is around 1736.67, and the proportion of students is approximately 0.377.

From the result, we have the model:

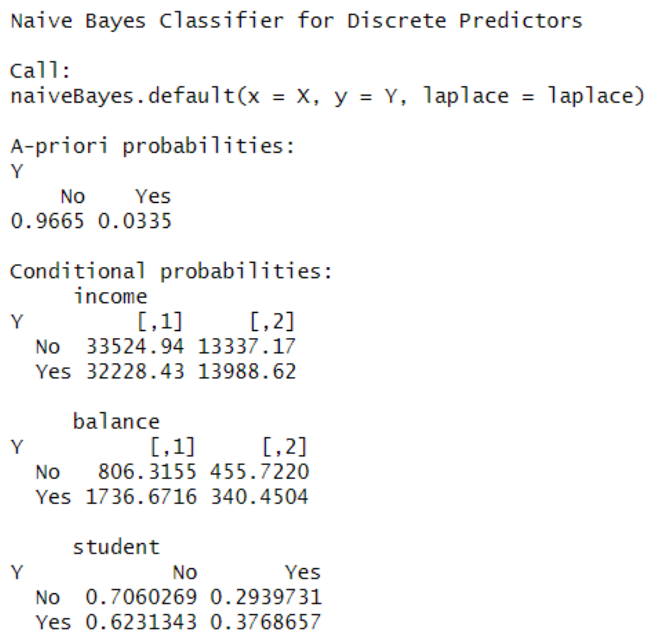
Default = 0.0000026*Income +0.00225 * Balance – 0.19 * Student

The coefficients of linear discriminants play a crucial role in the discriminant function by assigning weights to each predictor variable. In this model, the coefficient for "income" is approximately 0.0000026, indicating a modest positive contribution to the discriminant score. On the other hand, the coefficient for "balance" is approximately 0.00225, implying a more substantial positive impact on the discriminant score. This suggests that "balance" carries more weight in distinguishing default and non-default cases. Additionally, the coefficient for the "student" variable is approximately -0.19, demonstrating a negative contribution to the discriminant score. Since "student" is represented as 1 for "yes" and 0 for "no", the negative sign suggests that the probability of default is lower for the student group compared to the non-student group.

These coefficients provide valuable insights into the relationship between predictor variables and the likelihood of default. The positive coefficient for "income" indicates that higher income levels will lead to an increased probability of default, while the larger positive coefficient for "balance" underscores its greater influence in predicting default. Conversely, the negative coefficient for the "student" variable highlights that being a student is associated with a reduced probability of default when compared to non-students. These findings offer valuable guidance for understanding the factors that contribute to credit card default and inform decision-making processes aimed at managing default risks.

4.2.3 Naïve Bayes

Figure 8: Result of Naïve Bayes model



```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
    No    Yes 
0.9665 0.0335 

Conditional probabilities:
     income
Y         [,1]     [,2]
  No  33524.94 13337.17
  Yes 32228.43 13988.62

     balance
Y          [,1]     [,2]
  No   806.3155 455.7220
  Yes 1736.6716 340.4504

     student
Y            No       Yes
  No  0.7060269 0.2939731
  Yes 0.6231343 0.3768657
```

The conditional probabilities provide valuable insights into the relationship between predictor variables and class labels. Specifically, when considering the "income" variable, we observe mean values of approximately 33524.94 (No) and 32228.43 (Yes). These values are accompanied by standard deviations of 13337.17 and 13988.62, respectively. This indicates the variation within each class for the "income" variable.

Similarly, for the "balance" variable, the mean values are approximately 806.3155 (No) and 1736.6716 (Yes). The standard deviations associated with these means are 455.7220 and 340.4504, respectively. These statistics shed light on the distribution and dispersion of the "balance" variable among the different class labels.

Moreover, the "student" variable provides insights into the probabilities of being a non-student or a student within each class. For the class label "No," the probability of being a non-student is approximately 0.706, while the probability of being a student is approximately 0.294. For the class label "Yes," the probability of being a non-student is approximately 0.623, while the probability of being a student is approximately 0.377.

4.3 Model Evaluation

Table 3: Metrics for comparison of the efficiency among three models

	Accuracy	Sensitivity	Specificity
Logistic Regression	0.971	0.9959	0.2754
LDA	0.9715	0.9979	0.2319
Naïve Bayes	0.969	0.9953	0.2319

When comparing the results of three machine learning models for credit card default prediction, namely Logistic Regression, Naive Bayes, and Linear Discriminant Analysis (LDA), several key performance metrics were evaluated: accuracy, sensitivity, and specificity.

Logistic Regression, referred to as Model 1, attained an accuracy of 0.971, denoting its capability to accurately classify 97.1% of the dataset. The sensitivity, a measure of the model's proficiency in correctly recognizing non-default accounts, was notably high at 0.9959, indicating its accurate prediction of 99.59% of the true non-default cases. However, the specificity score of 0.2754 suggests a relatively diminished capacity to precisely identify default accounts.

Model 2, LDA, achieved an accuracy of 0.9715, which is comparable to the other models. The sensitivity score of 0.9979 indicates that it accurately predicted 99.79% of the true non-default cases. However, similar to the Naive Bayes model, the specificity score of 0.2319 suggests a lower ability to correctly identify default accounts.

Model 3, Naive Bayes, demonstrated a slightly lower accuracy of 0.969, indicating a classification accuracy of 96.9%. The sensitivity score of 0.9953 suggests that it accurately predicted 99.53% of the true non-default cases. However, the specificity score of 0.2319 implies that it had a lower ability to correctly identify default accounts.

When comparing the results, all three models exhibit similar accuracy levels, ranging from 0.969 to 0.9715. This suggests that they are all reasonably effective in correctly classifying the majority of the cases. However, the sensitivity and specificity scores reveal some differences in their performance.

In terms of sensitivity, all three models achieved high scores, ranging from 0.9953 to 0.9979, indicating a strong ability to identify true non-default cases. However, when considering specificity, both the Naive Bayes and LDA models exhibit lower scores of 0.2319, indicating a relatively weaker ability to accurately identify default accounts. This means that these models are more likely to misclassify default accounts as non-default, leading to weak alarms for interventions. On the other hand, the Logistic Regression model demonstrates a higher specificity score of 0.2754, suggesting a comparatively better ability to correctly classify default accounts. This indicates a reduced likelihood of more accurate identification of customers who are likely to default on their credit card payments.

In summary, while all three models performed reasonably well in terms of accuracy and sensitivity, the Logistic Regression model appears to have a slight advantage in terms of specificity.

Robustness

Due to the imbalanced dataset, the performance of specificity compared to other metrics is dramatically lower. Worse performance of specificity means that the financial institutions fail to forecast the accounts which are likely to default. When the model fails to detect default customers, it can expose the lender to higher risks of financial losses and increased bad debt. Besides, this weakness of the model can also negatively impact the financial institution's portfolio management and overall risk assessment. Undetected default risks can lead to a skewed risk profile, potentially affecting the institution's overall financial stability.

Based on the financial instructions’ risk tolerance, we will have the desired objective. In our case, without any specific goals, we will justify the threshold and use Synthetic Minority Oversampling Technique (SMOTE) to tune the model.

5.1 Threshold Modification

Table 4: Models’ threshold justification

Threshold	Algorithms	Accuracy	Sensitivity	Specificity
Prob > 0.5 assigned as default	Logistic Regression	0.971	0.9959	0.2754
	LDA	0.9715	0.9979	0.2319
	Naïve Bayes	0.969	0.9953	0.2319
Prob > 0.4 assigned as default	Logistic Regression	0.9725	0.9933	0.3913
	LDA	0.9715	0.9959	0.2899
	Naïve Bayes	0.966	0.9891	0.3188
Prob > 0.3 assigned as default	Logistic Regression	0.969	0.9871	0.4638
	LDA	0.9715	0.9907	0.4348
	Naïve Bayes	0.9615	0.9793	0.4638

Table 4 presents the performance metrics of three different machine learning algorithms (Logistic Regression, Linear Discriminant Analysis - LDA, and Naïve Bayes) for credit card default prediction at different probability thresholds. Each row of the table corresponds to a specific threshold value, and the columns represent the algorithms' accuracy, sensitivity (true positive rate), and specificity (true negative rate) at that threshold.

As the threshold is lowered, moving from 0.5 to 0.4, the models' accuracy tends to increase slightly. This is because lowering the threshold allows the models to classify more instances as default, increasing true positives and reducing false negatives. Consequently, specificity also increases for all algorithms, indicating a better ability to correctly detect default accounts. However, this improvement in specificity comes at the cost of a decrease in sensitivity, as more instances are now classified as default, leading to a higher number of false negative.

In the context of logistic regression, we observed that reducing the probability threshold from 0.5 to 0.4 significantly improved the overall model performance. At the threshold of 0.4, the accuracy of the model increased substantially, reaching 97.25%. Although there was a slight decrease in sensitivity from 99.59% to 99.33%, the specificity showed a remarkable improvement, rising from 27.54% to 39.19%. This enhancement in specificity suggests that the model became more adept at correctly identifying non-default accounts, making it the most effective model overall. On the other hand, the performance of the Linear Discriminant Analysis (LDA) model remained consistent across different probability thresholds. Lowering the threshold from 0.5 to 0.4 led to a slight decrease in sensitivity, from 99.59% to 99.07%, while specificity increased from 23.19% to 28.99%. Despite this improvement, the LDA model still lags the logistic regression model in terms of overall performance. In comparison, the Naïve Bayes model consistently demonstrated weaker performance compared to the logistic regression and LDA models. At a threshold of 0.4, the Naïve Bayes model achieved an accuracy of 96.6%, a sensitivity of 98.91%, and a specificity of 31.88%. These metrics suggest that the Naïve Bayes model struggles to accurately classify both default and non-default accounts.

Further lowering the threshold to 0.3 results in a decrease in accuracy for Logistic Regression and Naïve Bayes, while LDA maintains a relatively stable accuracy value. At this threshold, the models classify even more instances as default, leading to higher specificity values, but sensitivity values continue to decrease. This trend is consistent across all three algorithms.

In comparison among three models, when threshold equal to 0.3, LDA is the best model when its accuracy is stable at 0.9715 compared to the decrease in accuracy in both Logistic Regression at 0.969 and Naïve Bayes at 0.9615. Specifically, the specificity in LDA is lowest at 0.4348, with the least trade off to keep high sensitivity at 0.9907.

5.2 Synthetic Minority Oversampling Technique (SMOTE)

Table 5: SMOTE method result

	Accuracy	Sensitivity	Specificity
Logistic Regression	0.859	0.8602	0.8261
LDA	0.832	0.8301	0.8841
Naïve Bayes	0.836	0.8353	0.8551

SMOTE, standing for Synthetic Minority Oversampling Technique, is a versatile and widely used technique in machine learning to address imbalanced datasets. It is based on the intuition that by introducing synthetic instances to the minority class (in our case the “Yes” – default group is considered to be minority), the decision boundary of the classifier can be better learned, leading to improved classification performance. SMOTE utilizes the concept of k-nearest neighbors to ensure that the synthetic instances are plausible and representative of the minority class. The synthetic instances are generated by linearly interpolating the feature values between the minority class instance and its randomly selected neighbor. This interpolation ensures that the synthetic instances fall within the same range as the existing instances in the minority class, preserving the underlying data distribution.

Table 5 demonstrates a balanced performance of three metrics when compared to the original models. The accuracies of the models range from 0.832 to 0.859, with Logistic Regression achieving the highest accuracy. In terms of sensitivity, Logistic Regression also outperforms the other models with a sensitivity index of 0.8602, while LDA and Naïve Bayes achieve sensitivities of 0.832 and 0.835, respectively. On the other hand, LDA surpasses the other two models in terms of specificity, achieving a value of 0.8841. This indicates that LDA is the most effective model in forecasting default accounts.

Conclusion

The prediction of an account's default status relies on three variables: income, balance, and student. The model indicates that both student status and balance exert a significant influence on default probability, while income does not provide sufficient evidence to reject the null hypothesis (suggesting it has no impact on default). Examining the relationships between these variables, it becomes evident that higher balances in accounts correspond to an increased likelihood of default risk. Interestingly, it is unexpected to find that customers with higher incomes exhibit a higher probability of default. Additionally, student customers appear to have a lower likelihood of default compared to non-student customers. These findings shed light on the intricate dynamics between income, balance, student status, and default probability, suggesting that the interplay of these factors plays a crucial role in understanding and predicting default behaviors in credit card accounts. Further research is necessary in exploring the underlying mechanisms behind these associations and refining the predictive models for more accurate default predictions. Such insights have the potential to inform credit risk management strategies, enabling financial institutions to make informed decisions and implement appropriate measures to mitigate default risks effectively.

In terms of model performance, the models exhibit a high level of accuracy, correctly classifying a significant portion of the observations. However, a deeper analysis of sensitivity and specificity metrics reveals additional considerations. The sensitivity metric demonstrates the model's effectiveness in predicting non-default accounts, with a remarkably high percentage of "No" values accurately classified. This strong performance is influenced by the imbalanced distribution of the dataset, where the number of non-default accounts outweighs the number of default accounts. While this boosts the overall accuracy, it is essential to acknowledge that the model's performance in detecting default accounts is relatively weaker. The specificity metric highlights the model's limitations in accurately identifying default accounts within the dataset. The low percentage of true "Yes" values correctly detected suggests room for improvement in effectively capturing default instances. It is crucial to address this issue to ensure that the model is reliable in detecting default accounts, which is of utmost importance in credit risk assessment.

Recommendation

Based on the results obtained, it is recommended to further refine and enhance the predictive model for credit card default prediction. While the model demonstrates significant predictive power, there are certain areas that can be improved to increase its effectiveness and reliability.

Firstly, considering the impact of income on default prediction, although it does not show a significant influence in the current model, it is advisable to explore alternative income-related variables or consider additional data sources that may provide more comprehensive insights into an individual's financial situation. This could potentially improve the model's ability to capture the nuances of income and its relationship with default risk.

Secondly, considering alternative machine learning algorithms that are known to perform well in handling imbalanced datasets, such as Random Forest or K-Nearest Neighbors (KNN). Additionally, fine-tune the model's hyperparameters through techniques like grid search or random search to optimize the model's performance specifically for detecting default accounts.

Thirdly, to further enhance the predictive power of the model, it is recommended to incorporate additional predictors or features such as macroeconomic indicators, payment patterns, demographic information, etc. The inclusion of suggested relevant variables can extend the understanding of factors contributing to credit card default in a more comprehensive way. By expanding the set of predictors, the model can capture a wider range of influential factors and potentially improve its accuracy and reliability in predicting default risks.

Moreover, it is important to continue monitoring and evaluating the model's performance over time as new data becomes available. This will help identify any changes or trends in default patterns and ensure the model remains up-to-date and effective in predicting default risks accurately.

Bibliography

1. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357.
2. Hand, D. J., & Henley, W. E. (1997). Statistical Classification Methods in Consumer Credit Scoring: a Review. Journal of the Royal Statistical Society: Series A (Statistics in Society), 160(3), 523-541.
3. Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. Expert Systems with Applications, 33(4), 847-856.
4. Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research, 247(1), 124-136.