Business Analysis

Predicting default payments with logistic regression

Authors: Juan Castro, Rolando Diaz, and David Heller

ABSTRACT

This paper investigates the predictive capabilities of logistic regression in identifying credit card clients at the highest risk of defaulting on payments in the subsequent month. Utilizing the Default of Credit Card Clients Dataset, which encompasses a comprehensive array of demographic factors, credit data, and payment history of clients in Taiwan from April 2005 to September 2005, our analysis aims to discern the pivotal features influencing default behavior.

The research questions guiding our inquiry delve into the effectiveness of machine learning in credit risk assessment, the socio-economic factors correlated with default likelihood, and the discernible spending habits predictive of future default occurrences.

Logistic regression, chosen for its aptitude in binary classification tasks, serves as the predictive model. Through a systematic evaluation of model performance and feature importance, we elucidate the determinants driving credit card default and shed light on actionable insights for financial institutions.

The findings highlight the key predictors such as repayment status, credit limit, and payment history exerting substantial influence. Additionally, demographic and socio-economic factors such as age, education, and marital status exhibit notable associations with default propensity.
Furthermore, our analysis unveils discernible patterns in customer spending habits that portend future default occurrences, providing valuable insights for risk mitigation strategies.

This study underscores the application of data science in business analytics, showcasing how the fusion of statistical analyses and machine learning techniques can yield actionable insights for risk management in the financial sector.

The implications of our findings extend beyond predictive accuracy, offering avenues for informed decision-making and proactive intervention to safeguard against credit card defaults.

MOTIVATION

As Business Analytics track students, we recognize the pivotal role of data science in informing strategic decision-making processes within organizations. Our motivation to undertake this study stems not only from academic inquiry but also from a desire to shed light on the practical applications of data analytics in business operations.

One of our key motivations is to demonstrate how the data science process can be translated into actionable insights for managers and non-data personnel.

In essence, this research is motivated by the practical implications for business analytics and risk management strategies. By elucidating the factors driving credit card default and providing actionable insights for identifying high-risk clients, our study contributes to the development of proactive risk mitigation measures and the optimization of resource allocation within financial institutions.

Ultimately, the motivation behind this study lies in its potential to empower financial stakeholders with the knowledge and tools necessary to navigate the intricacies of credit risk assessment in an increasingly dynamic and data-driven landscape. By bridging the gap between data science and business analytics, we endeavor to foster informed decision-making and cultivate resilience in the face of evolving financial challenges.

DATA AND LIBRARIES

For this study, we utilized the Default of Credit Card Clients Dataset, sourced from the UCI Machine Learning Repository. The dataset comprises 25 variables, including demographic information, credit data, payment history, and default payment status of credit card clients in Taiwan from April 2005 to September 2005.

There are 25 variables:

ID: ID of each client
LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit
SEX: Gender (1=male, 2=female)
EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
MARRIAGE: Marital status (1=married, 2=single, 3=others)
AGE: Age in years
PAY_0: Repayment status in September, 2005 (-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)
PAY_2: Repayment status in August, 2005 (scale same as above)
PAY_3: Repayment status in July, 2005 (scale same as above)
PAY_4: Repayment status in June, 2005 (scale same as above)
PAY_5: Repayment status in May, 2005 (scale same as above)
PAY_6: Repayment status in April, 2005 (scale same as above)
BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)
BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)
BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)
BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)

BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)
PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)
PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)
PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)
PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)
PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)
default.payment.next.month: Default payment (1=yes, 0=no)

To conduct the logistic regression in R, we utilized several libraries for data manipulation, model fitting, and interpretation. The primary library is stats, which is included by default in R and provides functions for basic statistical analyses. Additionally, glm (Generalized Linear Models) library is indispensable for fitting logistic regression models using the glm() function. For data manipulation and preparation, dplyr and tidyr came in handy, offering a variety of functions for data wrangling and tidying. Visualization of model results and diagnostic plots were generated with ggplot2. Lastly, for model evaluation and inference, car (Companion to Applied Regression) and broom were really useful, providing functions for model diagnostics, hypothesis testing, and result extraction. Using all of these libraries in conjunction, we were able to achieve the results and graphs needed for our analysis.

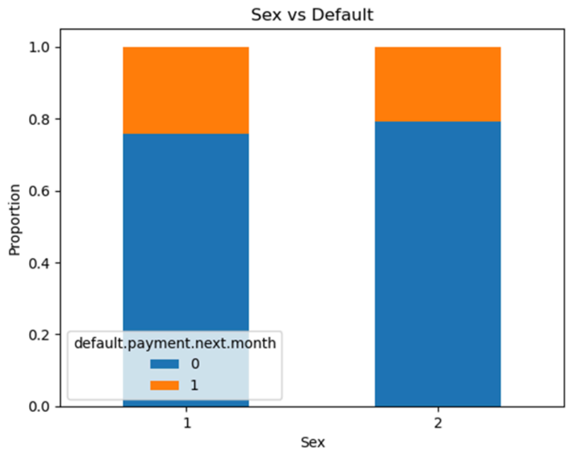
DATA PREPARATION

Ensuring data integrity is essential for meaningful analysis. In our study, we meticulously prepared the Default of Credit Card Clients Dataset to ensure its suitability for accurate modeling and prediction of credit card default behavior.

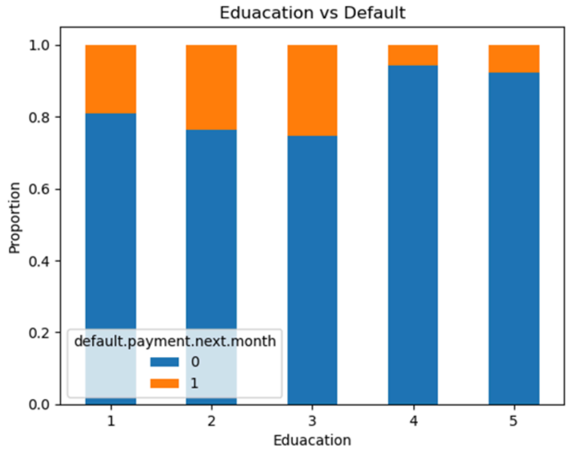
Addressing missing values was our first priority. We identified that only the 'education' and 'marriage' columns contained missing values. To maintain the dataset's integrity, we employed careful imputation strategies.

EXPLORATORY DATA ANALYSIS

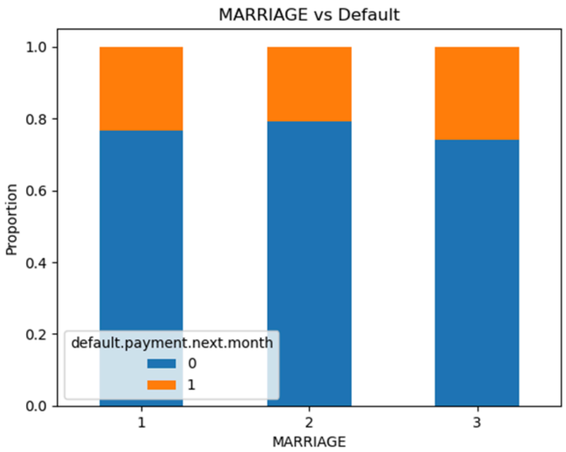
Analysis of the relationship between gender and default status revealed that, overall, there was a small margin between males and females in terms of default likelihood. Interestingly, females exhibited a slightly lower proportion of defaults compared to males, although the difference was not substantial. This suggests that gender alone may not be a significant predictor of default behavior, as both genders displayed similar patterns of default propensity.



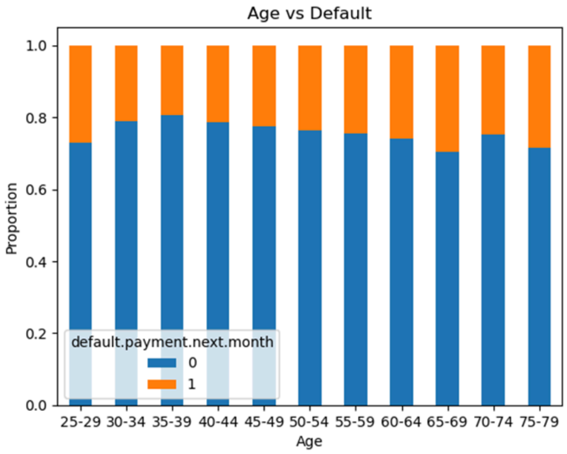
Examining the impact of education level on default likelihood revealed notable trends across different education categories. Graduates (coded as 1) and individuals with bachelor's degrees (coded as 2) displayed lower proportions of default compared to those with high school education (coded as 3). This trend suggests that lower levels of education are associated with a decreased default propensity. Additionally, individuals categorized as 'others' or 'unknown' (coded as 4 or 5) exhibited lower default rates, indicating that they may not be typical students. This observation underscores the potential influence of educational debt on default behavior, with higher education levels potentially correlating with greater financial power.



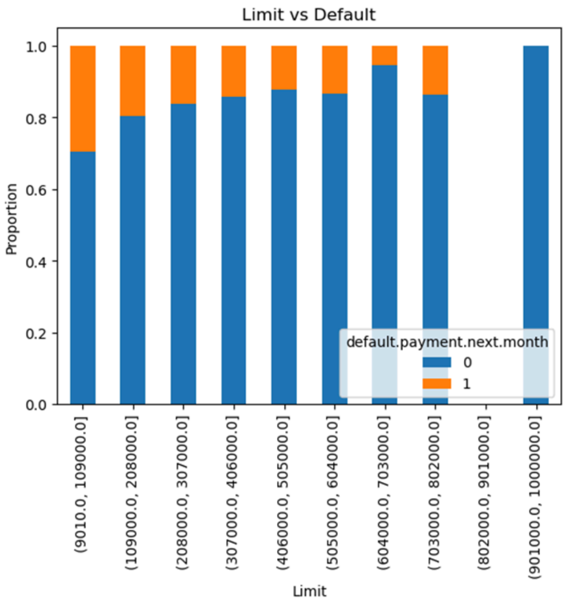
Analysis of marital status in relation to default behavior indicated that single individuals exhibited a slightly lower proportion of defaults compared to married individuals and those categorized as 'others'. However, the differences between marital status groups were minimal overall, suggesting that marital status alone may not be a strong predictor of default likelihood. These findings imply that other factors beyond marital status may play a more significant role in determining credit card default behavior.



Exploring the relationship between age and default likelihood revealed intriguing patterns. The age group of 25-29 exhibited the highest likelihood of default. The age groups of 30-34 and 35-39, which showed a significant improvement in default rates. Subsequently, a periodic increase in default likelihood was observed as age increased, reaching a peak around age 79. These findings suggest that age may be a crucial determinant of default behavior, with younger individuals and older adults exhibiting higher default propensity.



Analysis of credit limit in relation to default behavior demonstrated a clear trend: lower credit limits were associated with higher likelihoods of default, while higher credit limits corresponded to lower default rates. This observation suggests that individuals with higher credit limits may be more financially stable and capable of managing their debts effectively. Notably, this trend highlights the importance of creditworthiness and financial capacity in mitigating default risk, as individuals with greater financial resources exhibit lower default propensity.



In summary, exploratory data analysis revealed intriguing insights into the relationships between demographic factors and credit card default behavior. While certain variables such as education level and age displayed discernible trends in default likelihood, others such as gender and marital status exhibited minimal differences. These findings underscore the complexity of default prediction and highlight the multifaceted nature of factors influencing credit risk.

METHODS AND EVALUATION

Logistic Regression

Logistic regression predicts the probability that an observation belongs to one of two classes (binary outcome) by fitting data to a logistic curve. It calculates the odds of the probability (p) of being in the default class (e.g., event happening) as a function of the independent variables. The logistic function ensures that the probability estimate is bounded between 0 and 1. Mathematically, logistic regression estimates a multiple linear regression function defined as:

$$logit(p) = \log\left(\frac{P(y = 1\ given\ X)}{p(y = 0\ given\ X)}\right) = \log\,(\frac{p}{1-p}) = \beta_0 + \ \beta_1 x_1 + \beta_2\, x_2 + \cdots \beta_k x_k$$

Evaluation of Logistic Regression model

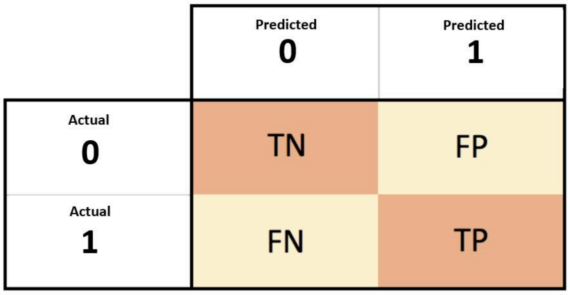
Null Deviance measures the deviance of a model with no predictor variables. It provides a baseline for comparison when assessing the improvement in model fit with the inclusion of predictors.
Residual Deviance measures the deviance after fitting a model with predictor variables. It represents the unexplained

variability in the response variable after accounting for the predictors. The comparison between null deviance and residual deviance is commonly used for model assessment, with a significant reduction from null to residual deviance indicating that the predictors contribute significantly to explaining the variability in the response variable.

Information Criteria: Akaike Information Criteria (AIC) and BIC (Bayesian Information Criterion)

Both AIC and BIC provide a method for assessing the quality of a model through comparison of related models. They are based on the Deviance but penalized for making the model more complicated. If there is more than one similar candidate model (where all the variables of the simpler model occur in the more complex models), then select the model that has the smallest AIC or BIC.

Confusion matrix is the most crucial metric commonly used to evaluate classification models. A confusion matrix is formed from the four outcomes produced from a binary classification. It provides a summary of the predictions made by a model compared to the actual outcomes. A binary classifier predicts all data instances of a test dataset as either positive or negative. This classification (or prediction) produces four outcomes – true positive, true negative, false positive and false negative. A confusion matrix of binary classification is a two-by-two table formed by counting the number of the four outcomes of a binary classifier. We usually denote them as TP, FP, TN, and FN.



	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

- Overall error rate is calculated as the number of all incorrect predictions divided by the total number of the dataset.

$$Error\ Rate = \frac{FN + FP}{N}$$

- Overall Accuracy rate is calculated as the number of all correct predictions divided by the total number of the dataset.

$$Accuracy\ Rate = \frac{TP + TN}{N}$$

- Sensitivity is calculated as the number of correct positive predictions divided by the total number of positives. It is also called true positive rate (TPR).

$$Sensitivity = \frac{TP}{TP + FN}$$

- Specificity is calculated as the number of correct negative predictions divided by the total number of negatives. It is also called true negative rate (TNR).

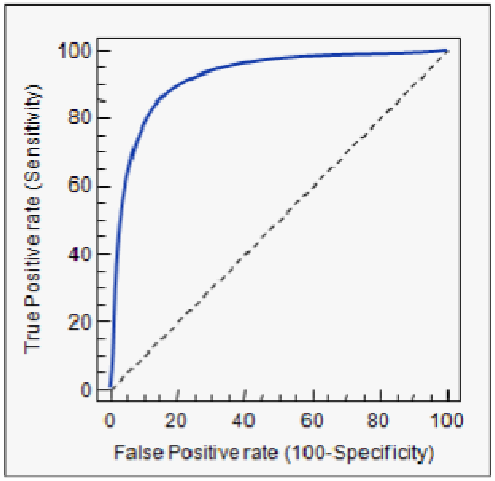
$$Specificity = \frac{TN}{FP + TN}$$

A Receiver Operating Characteristic (ROC) curve is a graphical representation used to assess the performance of a binary classification model, such as logistic regression.The ROC curve illustrates the trade-off between sensitivity (true positive rate) and specificity (true negative rate) at various decision thresholds. If the predicted probability is above the threshold, the observation is classified as the positive class; otherwise, it's classified as the negative class. The ROC curve visualizes the model's performance across different threshold values.

The area under the ROC curve (AUC) summarizes the overall performance of the model across all possible thresholds. AUC ranges from 0 to 1, where higher values indicate better performance. An AUC of 0.5 corresponds to a model that performs no better than random chance, while an AUC of 1 indicates perfect performance. A steeper ROC curve, closer to the top-left corner of the plot, suggests better overall performance.

The diagonal line (45-degree line) represents random guessing, and a model with good discrimination ability should have a curve above this line. An ideal ROC curve will hug the top left corner, so the larger the AUC the better the classifier. We expect a classifier that performs no better than chance to have an AUC of 0.5

ROC curves are useful for comparing different classifiers since they consider all possible thresholds.

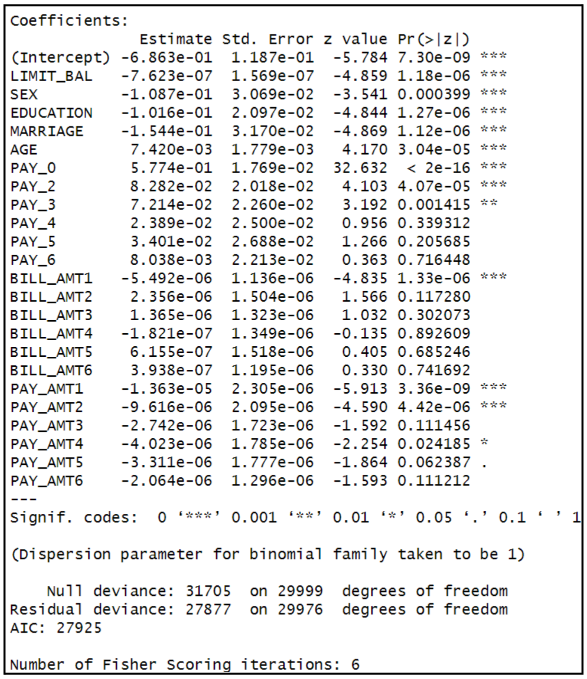


RESULTS

Full Model

log(p/1-p) = -6.863 -7.623LIMIT_BAL -1.087SEX -1.016EDUCATION -1.544MARRIAGE +7.420AGE +5.774PAY_0 +8.282PAY_2 + 7.214PAY_3 + 2.389PAY_4 + 3.401PAY_5 + 8.038PAY_6 -5.492BILL_AMT1 + 2.356BILL_AMT2 + 1.365BILL_AMT3 -1.821BILL_AMT4 + 6.155BILL_AMT5 + 3.938BILL_AMT6 -1.363PAY_AMT1 -9.616PAY_AMT2 -2.742PAY_AMT3 -4.023PAY_AMT4 -3.311PAY_AMT_5 -2.064PAY_AMT6
Where p = probability of defaulting the payment next month.

Summary:



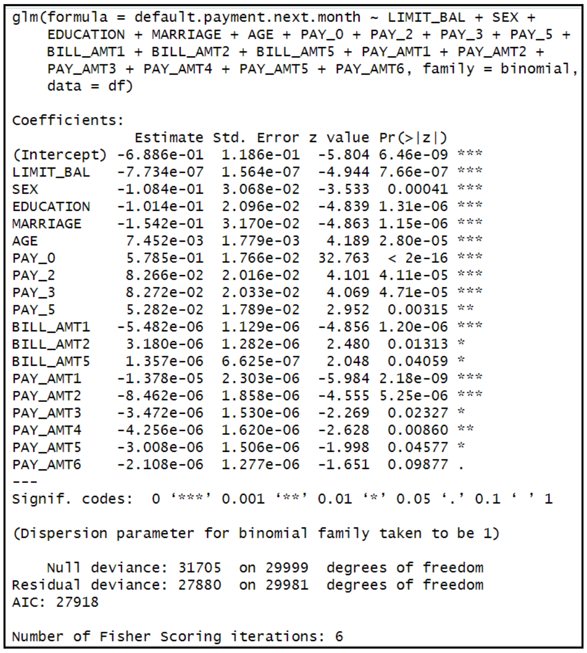
```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -6.863e-01  1.187e-01  -5.784 7.30e-09 ***
LIMIT_BAL   -7.623e-07  1.569e-07  -4.859 1.18e-06 ***
SEX         -1.087e-01  3.069e-02  -3.541 0.000399 ***
EDUCATION   -1.016e-01  2.097e-02  -4.844 1.27e-06 ***
MARRIAGE    -1.544e-01  3.170e-02  -4.869 1.12e-06 ***
AGE          7.420e-03  1.779e-03   4.170 3.04e-05 ***
PAY_0        5.774e-01  1.769e-02  32.632  < 2e-16 ***
PAY_2        8.282e-02  2.018e-02   4.103 4.07e-05 ***
PAY_3        7.214e-02  2.260e-02   3.192 0.001415 ** 
PAY_4        2.389e-02  2.500e-02   0.956 0.339312    
PAY_5        3.401e-02  2.688e-02   1.266 0.205685    
PAY_6        8.038e-03  2.213e-02   0.363 0.716448    
BILL_AMT1   -5.492e-06  1.136e-06  -4.835 1.33e-06 ***
BILL_AMT2    2.356e-06  1.504e-06   1.566 0.117280    
BILL_AMT3    1.365e-06  1.323e-06   1.032 0.302073    
BILL_AMT4   -1.821e-07  1.349e-06  -0.135 0.892609    
BILL_AMT5    6.155e-07  1.518e-06   0.405 0.685246    
BILL_AMT6    3.938e-07  1.195e-06   0.330 0.741692    
PAY_AMT1    -1.363e-05  2.305e-06  -5.913 3.36e-09 ***
PAY_AMT2    -9.616e-06  2.095e-06  -4.590 4.42e-06 ***
PAY_AMT3    -2.742e-06  1.723e-06  -1.592 0.111456    
PAY_AMT4    -4.023e-06  1.785e-06  -2.254 0.024185 *  
PAY_AMT5    -3.311e-06  1.777e-06  -1.864 0.062387 .  
PAY_AMT6    -2.064e-06  1.296e-06  -1.593 0.111212    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 31705  on 29999  degrees of freedom
Residual deviance: 27877  on 29976  degrees of freedom
AIC: 27925

Number of Fisher Scoring iterations: 6
```

Coefficients interpretation:
Each predictor's coefficient estimates its effect on the log-odds of the default happening, given all other predictors are held constant.
For example, a negative coefficient (-0.1016) for EDUCATION means that as the EDUCATION level increases by one unit, the log odds of defaulting on the next month's payment decreases by 0.1016. In terms of probability, this suggests that higher educational levels are associated with a lower probability of default.
In percentage, we can do e to the power of -0.1016 which is 0.9033. If we subtract 1 minus this, we get -0.0966. For each one-unit increase in the EDUCATION level, the odds of defaulting on the next month's payment decrease by approximately 9.66%.

Optimal Model

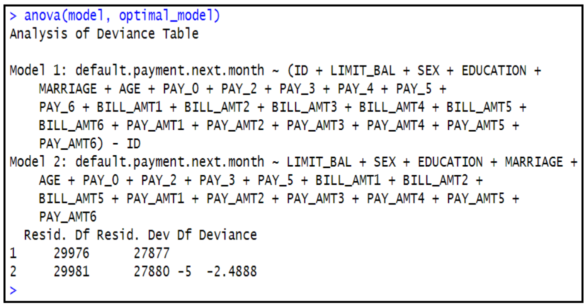
We use the step() function in R. Stepwise regression is used to select the best predictive variables for a model. It aims to simplify the model by including only those variables that significantly contribute to the prediction of the dependent variable.

Optimal Model Summary:



```
glm(formula = default.payment.next.month ~ LIMIT_BAL + SEX + 
    EDUCATION + MARRIAGE + AGE + PAY_0 + PAY_2 + PAY_3 + PAY_5 + 
    BILL_AMT1 + BILL_AMT2 + BILL_AMT5 + PAY_AMT1 + PAY_AMT2 + 
    PAY_AMT3 + PAY_AMT4 + PAY_AMT5 + PAY_AMT6, family = binomial, 
    data = df)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -6.886e-01  1.186e-01  -5.804 6.46e-09 ***
LIMIT_BAL   -7.734e-07  1.564e-07  -4.944 7.66e-07 ***
SEX         -1.084e-01  3.068e-02  -3.533  0.00041 ***
EDUCATION   -1.014e-01  2.096e-02  -4.839 1.31e-06 ***
MARRIAGE    -1.542e-01  3.170e-02  -4.863 1.15e-06 ***
AGE          7.452e-03  1.779e-03   4.189 2.80e-05 ***
PAY_0        5.785e-01  1.766e-02  32.763  < 2e-16 ***
PAY_2        8.266e-02  2.016e-02   4.101 4.11e-05 ***
PAY_3        8.272e-02  2.033e-02   4.069 4.71e-05 ***
PAY_5        5.282e-02  1.789e-02   2.952  0.00315 ** 
BILL_AMT1   -5.482e-06  1.129e-06  -4.856 1.20e-06 ***
BILL_AMT2    3.180e-06  1.282e-06   2.480  0.01313 *  
BILL_AMT5    1.357e-06  6.625e-07   2.048  0.04059 *  
PAY_AMT1    -1.378e-05  2.303e-06  -5.984 2.18e-09 ***
PAY_AMT2    -8.462e-06  1.858e-06  -4.555 5.25e-06 ***
PAY_AMT3    -3.472e-06  1.530e-06  -2.269  0.02327 *  
PAY_AMT4    -4.256e-06  1.620e-06  -2.628  0.00860 ** 
PAY_AMT5    -3.008e-06  1.506e-06  -1.998  0.04577 *  
PAY_AMT6    -2.108e-06  1.277e-06  -1.651  0.09877 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 31705  on 29999  degrees of freedom
Residual deviance: 27880  on 29981  degrees of freedom
AIC: 27918

Number of Fisher Scoring iterations: 6
```

Comparing the two models



```
> anova(model, optimal_model)
Analysis of Deviance Table

Model 1: default.payment.next.month ~ (ID + LIMIT_BAL + SEX + EDUCATION + 
    MARRIAGE + AGE + PAY_0 + PAY_2 + PAY_3 + PAY_4 + PAY_5 + 
    PAY_6 + BILL_AMT1 + BILL_AMT2 + BILL_AMT3 + BILL_AMT4 + BILL_AMT5 + 
    BILL_AMT6 + PAY_AMT1 + PAY_AMT2 + PAY_AMT3 + PAY_AMT4 + PAY_AMT5 + 
    PAY_AMT6) - ID
Model 2: default.payment.next.month ~ LIMIT_BAL + SEX + EDUCATION + MARRIAGE + 
    AGE + PAY_0 + PAY_2 + PAY_3 + PAY_5 + BILL_AMT1 + BILL_AMT2 + 
    BILL_AMT5 + PAY_AMT1 + PAY_AMT2 + PAY_AMT3 + PAY_AMT4 + PAY_AMT5 + 
    PAY_AMT6
  Resid. Df Resid. Dev Df Deviance
1     29976      27877            
2     29981      27880 -5  -2.4888
> 
```

-Degrees of Freedom (Resid. Df):

- Model 1 (Full Model) has 29,976 degrees of freedom.
- Model 2 (Optimal Model) has 29,981 degrees of freedom.

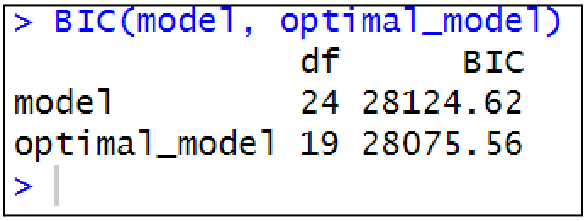
The increase in degrees of freedom from Model 1 to Model 2 indicates that Model 2 has fewer parameters (simpler model).
-Residual Deviance:

- Model 1 shows a residual deviance of 27,877.

- Model 2 shows a residual deviance of 27,880.

A smaller residual deviance generally indicates a better model fit, though in this case, the increase in deviance is very small as Model 2 sacrifices some of the fit to reduce model complexity.
-Change in Degrees of Freedom (Df):
The difference in degrees of freedom between the two models is -5, indicating that five parameters were removed in Model 2.
-Change in Deviance (Deviance):
The deviance change is -2.4888, reflecting a minimal increase in residual deviance despite the reduced complexity. This suggests that the omitted parameters in Model 2 did not significantly contribute to improving the model's ability to fit the data, making the Optimal Model a more efficient choice without losing significant predictive power.

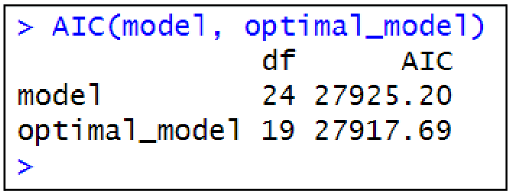
BIC:



```
> BIC(model, optimal_model)
              df      BIC
model         24 28124.62
optimal_model 19 28075.56
>
```

The lower BIC for Model 2 suggests that it is the preferable model when considering both the fit and complexity. The reduction in BIC from Model 1 to Model 2 (a difference of about 49.06 points) implies a significant improvement in terms of a balance between model simplicity and the ability to explain the dataset.

AIC:



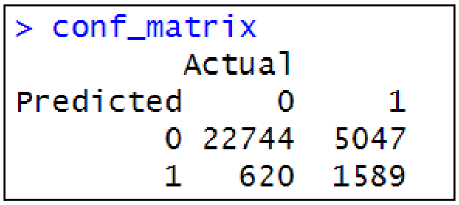
```
> AIC(model, optimal_model)
              df      AIC
model         24 27925.20
optimal_model 19 27917.69
>
```

The lower AIC for Model 2 indicates that it is the preferred model over Model 1. The decrease in AIC, though modest (7.51 points), still points towards Model 2 as offering a better balance between accuracy and complexity. This slight decrease suggests that the parameters removed during the optimization process were not crucial to the model's ability to describe the variability in the data, leading to a more parsimonious model without sacrificing much in terms of fit.

Because of these reasons, we decided to go with the Optimal Model.

Evaluating the optimal model

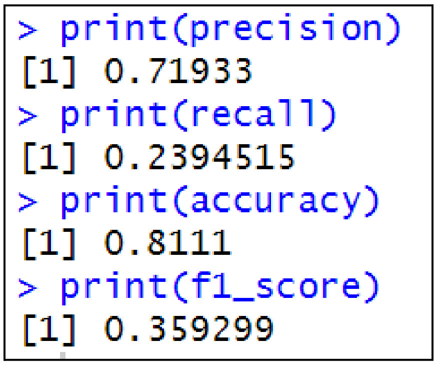
Confusion Matrix:



```
> conf_matrix
         Actual
Predicted     0     1
        0 22744  5047
        1   620  1589
```

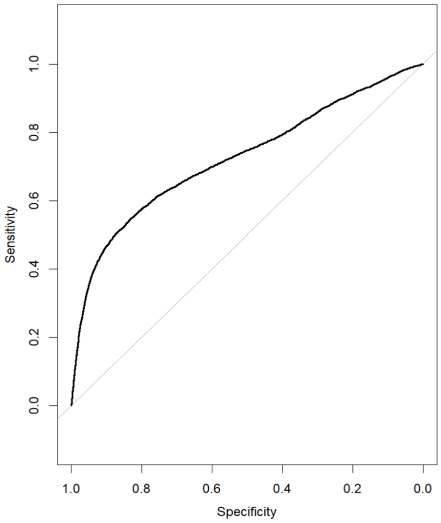
-True Negatives (TN): 22,744 (Predicted non-default, actual non-default)
-False Positives (FP): 620 (Predicted default, actual non-default)
-False Negatives (FN): 5,047 (Predicted non-default, actual default)
-True Positives (TP): 1,589 (Predicted default, actual default)

Measuring the model's performance:

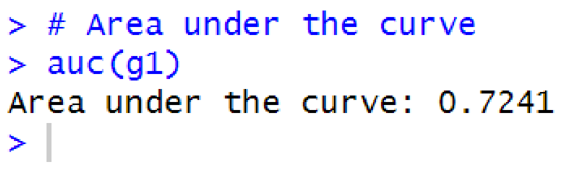


```
> print(precision)
[1] 0.71933
> print(recall)
[1] 0.2394515
> print(accuracy)
[1] 0.8111
> print(f1_score)
[1] 0.359299
```

ROC Curve:



AUC:



```
> # Area under the curve
> auc(g1)
Area under the curve: 0.7241
> 
```

CONCLUSIONS

Strengths of the model

-Good Discrimination Ability (AUC = 0.7241):
The AUC indicates that the model has a good ability to discriminate between those who will default and those who will not. An AUC of over 0.7 suggests that the model is relatively effective at ranking individuals by their likelihood of defaulting, which is crucial for prioritization in risk management.

-High Accuracy (Accuracy = 81.11%)
The model achieves a high overall accuracy, indicating that it correctly predicts both defaults and non-defaults a significant majority of the time. This is particularly reassuring for general performance across a broad dataset.

-High Precision (Precision = 71.93%)
High precision suggests that when the model predicts a default, it is correct about 71.93% of the time. This is beneficial in scenarios where the cost of a false positive (e.g., incorrectly predicting that someone will default) is high, as it minimizes unnecessary interventions or actions based on false alarms.

Weaknesses of the model

-Low Recall (Recall = 23.95%)
The model's major drawback is its low recall, which indicates it fails to identify a significant number of actual defaults (about 76% are missed). This is a serious issue in credit risk modeling, where failing to detect potential defaults can lead to substantial financial losses. This suggests the model is overly conservative, perhaps biased towards predicting non-defaults.

-Low F1 Score (F1 Score = 35.93%)
The F1 score, which balances precision and recall, is quite low. This indicates that the model does not effectively balance catching as many actual defaults as possible (recall) with avoiding false alarms (precision). A low F1 score in a dataset possibly skewed towards non-defaulters suggests that improvements are needed in capturing more complex patterns that might indicate default.

-Possible Bias Towards the Majority Class
Given the high accuracy in the face of low recall and a moderate AUC, the model might be biased towards predicting the majority class (non-defaulters). This is often a challenge in imbalanced datasets where the model tends to favor the more frequent class, leading to high overall accuracy but poor performance in detecting the minority class (defaulters).

In conclusion, the model shows proficiency in predicting non-defaults accurately and is conservative in marking defaults, indicated by its high precision and accuracy. However, its utility in a practical scenario where catching defaults is crucial is limited due to its low recall and F1 score. Enhancing the model's ability to detect true positives without a substantial increase in false positives is essential for making it more effective and reliable in predicting credit defaults.

Suggestions for model improvement:

-Undersampling the Majority Class: This method reduces the number of instances from the majority class to balance the dataset.

-Weighted Classes: We could use a class_weight parameter to make the model pay more attention to the minority class.
-Interaction Terms: We could add interaction effects between variables if certain conditions combined increase the likelihood of default.

-Feature Selection: We could use techniques like Recursive Feature Elimination (RFE) to identify and keep the most informative features to enhance model performance.

-It is beyond the scope of this course, but we could use an algorithm like decision trees considering that they often perform well on imbalanced data because their hierarchical structure allows them to learn signals from both classes.

APPENDIX

(python code)
EDA Code:

```
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
# Ignore warnings
warnings.filterwarnings("ignore")


# Read the dataset
df = pd.read_csv("UCI_Credit_Card.csv")

import pandas as pd

# Assuming df is your DataFrame with the 'AGE' column
# Get the minimum and maximum values of the 'AGE' column
min_age = df['LIMIT_BAL'].min()
```

```
max_age = df['LIMIT_BAL'].max()

print("Minimum LIMIT_BAL:", min_age)
print("Maximum LIMIT_BAL:", max_age)

# Assuming df is your DataFrame with the 'LIMIT_BAL' column
# Create bins for limit balance
bins = 10  # Number of bins

# Bin the 'LIMIT_BAL' column
df['LIMIT_BAL_bin'] = pd.cut(df['LIMIT_BAL'], bins=bins)

# Display the DataFrame with the new 'LIMIT_BAL_bin' column
print(df)

# Assuming df is your DataFrame with the 'AGE' column
# Create bins for age
bins = list(range(20, 80, 5))  # bins from 25 to 75, in intervals of 5

# Create labels for the bins
labels = [f"{i}-{i+4}" for i in range(20, 80, 5)]

# Bin the 'AGE' column
df['age_bin'] = pd.cut(df['AGE'], bins=bins, labels=labels, right=False)

# Drop the original 'AGE' column if needed
# df.drop(columns=['AGE'], inplace=True)

# Display the DataFrame with the new 'age_bin' column
df

import pandas as pd

# Assuming df is your DataFrame
# Delete rows with null values
df.dropna(inplace=True)

# Display the DataFrame after deleting rows with null values
print(df)

df = df[df['EDUCATION'] != 0]

# Assuming 'EDUCATION' is the name of the column
df.loc[df['EDUCATION'] == 6, 'EDUCATION'] = 5


df = df[df['MARRIAGE'] != 0]


import pandas as pd

# Assuming df is your DataFrame
# Check for null values in each column
null_columns = df.columns[df.isnull().any()]

# Print columns with null values and their corresponding counts
for column in null_columns:
    print(f"{column}: {df[column].isnull().sum()} null values")


# Assuming df is your DataFrame with the 'AGE' column
# Get the minimum and maximum values of the 'AGE' column
min_age = df['AGE'].min()
max_age = df['AGE'].max()

print("Minimum age:", min_age)
print("Maximum age:", max_age)


# Define features and target
X = df.drop(columns=['default.payment.next.month'])
y = df['default.payment.next.month']


# Fit logistic regression model
model = LogisticRegression()
model.fit(X, y)


# # Summary of the model
# print(model.coef_)
# print(model.intercept_)


# Predictions on test set
predicted_values = model.predict_proba(X)[:, 1]
print(predicted_values)


# You can convert probabilities to binary predictions based on a threshold
binary_predictions = (predicted_values > 0.5).astype(int)
print(binary_predictions)

# Model accuracy
accuracy = accuracy_score(y, binary_predictions)
print("Accuracy:", accuracy)


# # ANALYSIS

# Count the occurrences of each unique value in the 'default.payment.next.month' column
default_counts = df['default.payment.next.month'].value_counts()

print("Count of default.payment.next.month:")
print(default_counts)

# Create a bar plot
plt.figure(figsize=(8, 6))
sns.countplot(x='default.payment.next.month', hue='SEX', data=df)
plt.title('Count of default.payment.next.month by SEX')
plt.xlabel('default.payment.next.month')
plt.ylabel('Count')
plt.legend(title='SEX')
plt.show()


# Calculate proportions by SEX and
default.payment.next.month
proportions = df.groupby(['SEX',
'default.payment.next.month']).size().unstack()
proportions = proportions.div(proportions.sum(axis=1),
axis=0)

# Create a bar plot
plt.figure(figsize=(8, 6))
proportions.plot(kind='bar', stacked=True)
plt.title('Sex vs Default')
plt.xlabel('Sex')
plt.ylabel('Proportion')
plt.legend(title='default.payment.next.month', loc='lower left')
plt.xticks(rotation=0)
plt.show()


# Create a bar plot
plt.figure(figsize=(8, 6))
sns.countplot(x='default.payment.next.month',
hue='EDUCATION', data=df)
plt.title('Count of default.payment.next.month by
EDUCATION')
plt.xlabel('default.payment.next.month')
plt.ylabel('Count')
plt.legend(title='EDUCATION')
plt.show()


# Calculate proportions by SEX and
default.payment.next.month
proportions = df.groupby(['EDUCATION',
'default.payment.next.month']).size().unstack()
proportions = proportions.div(proportions.sum(axis=1),
axis=0)

# Create a bar plot
plt.figure(figsize=(8, 6))
proportions.plot(kind='bar', stacked=True)
plt.title('Eduacation vs Default')
plt.xlabel('Eduacation')
plt.ylabel('Proportion')
plt.legend(title='default.payment.next.month', loc='lower left')
plt.xticks(rotation=0)
plt.show()


# Calculate proportions by SEX and
default.payment.next.month
proportions = df.groupby(['MARRIAGE',
'default.payment.next.month']).size().unstack()
proportions = proportions.div(proportions.sum(axis=1),
axis=0)

# Create a bar plot
plt.figure(figsize=(8, 6))
proportions.plot(kind='bar', stacked=True)
plt.title('MARRIAGE vs Default')
plt.xlabel('MARRIAGE')
plt.ylabel('Proportion')
plt.legend(title='default.payment.next.month', loc='lower left')
plt.xticks(rotation=0)
plt.show()


# Calculate proportions by SEX and
default.payment.next.month
proportions = df.groupby(['age_bin',
'default.payment.next.month']).size().unstack()
proportions = proportions.div(proportions.sum(axis=1),
axis=0)

# Create a bar plot
plt.figure(figsize=(8, 6))
proportions.plot(kind='bar', stacked=True)
plt.title('Age vs Default')
plt.xlabel('Age')
plt.ylabel('Proportion')
plt.legend(title='default.payment.next.month', loc='lower left')
plt.xticks(rotation=0)
plt.show()
```

LIMIT_BAL

```
# Calculate proportions by SEX and
default.payment.next.month
proportions = df.groupby(['LIMIT_BAL_bin',
'default.payment.next.month']).size().unstack()
proportions = proportions.div(proportions.sum(axis=1),
axis=0)

# Create a bar plot
plt.figure(figsize=(8, 6))
proportions.plot(kind='bar', stacked=True)
plt.title('Limit vs Default')
plt.xlabel('Limit')
plt.ylabel('Proportion')
plt.xticks(rotation=90)
plt.show()
```

R Code:

```
#Read the dataset
df <- read.csv(file.choose())

#Check the dataset
df

#Fit logistic regression model
model <- glm(default.payment.next.month ~ .- ID, data = df,
family = binomial)

#Summary of the model
summary(model)

#Full Model
```

```
#p = probability of defaulting the payment next month
#log(p/1-p) = -6.863 -7.623LIMIT_BAL -1.087SEX
-1.016EDUCATION -1.544MARRIAGE +7.420AGE
+5.774PAY_0 +8.282PAY_2 + 7.214PAY_3 + 2.389PAY_4 +
3.401PAY_5 + 8.038PAY_6 -5.492BILL_AMT1 +
2.356BILL_AMT2 + 1.365BILL_AMT3 -1.821BILL_AMT4 +
6.155BILL_AMT5 + 3.938BILL_AMT6 -1.363PAY_AMT1
-9.616PAY_AMT2 -2.742PAY_AMT3 -4.023PAY_AMT4
-3.311PAY_AMT_5 -2.064PAY_AMT6


#Use step to find the optimal model
optimal_model<-step(model)

summary(optimal_model)

#Comparing the two models
anova(model, optimal_model)
BIC(model, optimal_model)
AIC(model, optimal_model)


# Predictions on test set
predicted_values <- predict(optimal_model, newdata = df,
type = "response")
predicted_values

predicted_class <- ifelse(predicted_values > 0.5, 1, 0)
actual_values <- df$default.payment.next.month
conf_matrix <- table(Predicted = predicted_class, Actual =
actual_values)
conf_matrix

# Extract elements from the confusion matrix
tp <- conf_matrix[2, 2]  # True positives
fp <- conf_matrix[2, 1]  # False positives
fn <- conf_matrix[1, 2]  # False negatives
tp
fp

# Calculate precision and recall
precision <- tp / (tp + fp)
recall <- tp / (tp + fn)

# Calculate F1 score
f1_score <- 2 * (precision * recall) / (precision + recall)

# Calculate accuracy
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)

# Print results
print(precision)
print(recall)
print(accuracy)
print(f1_score)

# Plotting the ROC curve
probab=predict(optimal_model,newdata = df, type =
"response")
df$prob=probab
library(pROC)
g1=roc(default.payment.next.month~prob, data=df)
plot(g1)

# Area under the curve
auc(g1) # An AUC of 0.7241 suggests the model has fair
predictive power, though there might still be room for
improvement
```

REFERENCES

Agresti, Alan. "Categorical Data Analysis." John Wiley & Sons, 2002.

Hosmer Jr, David W., Stanley Lemeshow, and Rodney X. Sturdivant. "Applied Logistic Regression." John Wiley & Sons, 2013.

Harrell Jr, Frank E. "Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis." Springer, 2015.