AFRICAN INSTITUTE FOR MATHEMATICAL SCIENCES

(AIMS RWANDA, KIGALI)

Default of Credit Card Clients

Dataset Analysis and Modeling

Predicting Credit Risk Using Logistic Regression

Course: Statistical Regression

Date: November 27, 2025

Academic Year: 2024/2025

Group 6 Members

Yan Kevin ZE

Arnaud FOUBEUDA BOZAHBE

Olusola Timothy OGUNDEPO

Consolee NISINGIZWE

Lecturer: Prof. Fabrizio Ruggeri

A comprehensive statistical analysis of credit card default prediction

Abstract

This report presents our group's work on predicting credit card defaults through logistic regression. Banks and financial companies struggle with figuring out which customers might not pay back their debts. We noticed that people who default usually share similar patterns in their payment history. This matters a lot because banks want to cut their losses but also treat all customers fairly. For our analysis, we worked with a dataset containing records from 30,000 credit card users in Taiwan.

What We Found:

- Recent payment behavior matters most - if someone paid late recently, their default risk jumps by 78%
- The model gets things right about 81.2% of the time overall, and it's really good (97.2%) at spotting customers who won't default
- We narrowed down from 23 possible features to 18 useful ones using forward stepwise selection with AIC
- The model's ability to separate defaulters from non-defaulters (AUC = 0.725) is decent but could be better

Basically, how people paid recently tells us way more than anything else. Whether they were late on their most recent payment is the biggest red flag. Things like education, marriage status, and credit limits help a bit, but payment history is where it's at.
One thing we noticed is that our model is much better at identifying people who won't default (97.2%) than catching those who will (only 25%). This happens because there are way more people who pay on time than those who don't in our dataset.

What This Means in Practice: Banks could actually use what we found here to watch for warning signs, especially when customers start missing payments. Our model works best as a first filter to spot obvious risk cases, but it shouldn't be the only thing banks rely on when making credit decisions.

Keywords: credit risk management, default prediction, payment behavior, logistic regression, statistical modeling, financial analytics, AIC optimization, ROC analysis

1 Introduction

Credit card defaults are a real headache for banks. With more and more people using credit cards everywhere, lenders really need solid methods to figure out who might not pay them back.

Background and Motivation

We're working with data from 30,000 credit card customers in Taiwan. For each person, we have 23 different records. Take for example, their age, education, credit limit, how they paid over six months, their bill amounts, and how much they actually paid. The key thing we're trying to predict is whether they defaulted the following month.
Banks basically want to know: who's going to default? The old way of doing things relies a lot on gut feelings and basic rules. But using data and statistical models gives us a more objective way to answer this question.

Problem Statement

When people don't pay their bills, banks lose money, spend more trying to collect what's owed, and take on extra risk. What we're trying to do is predict defaults early enough so banks can actually do something about these clients.

Research Objectives

What we wanted to accomplish:

1. Figure out which factors actually matter when predicting defaults
2. Build a logistic regression model that can estimate the chances that someone will default or not
3. Use AIC to help us pick only the variables that really add value to the model
4. Test our model properly to see how well it actually performs on new data

Data and Methods

Our dataset has 30,000 people with 23 different pieces of info like age, gender, education, and marital status, plus their credit limit, how they paid over six months, and their bill and payment amounts.
We picked logistic regression for a few good reasons: you can actually understand what the model's doing, it runs pretty fast even with lots of data, and it gives us probabilities instead of just yes/no answers. Here's how we approached it: First, we split our data into training and test sets (keeping the same proportion of defaulters in both). Then we used forward

stepwise selection with AIC to find which variables actually help without making the model too complicated. Finally, we looked at different metrics to really understand where our model does well and where it struggles.
Our process went like this: clean the data, explore it to see what patterns pop up, select the important features, train the model, and then test it on data it hasn't seen before (test sets).

2 Dataset and Data Preparation

Dataset Overview

Our dataset contains 30,000 credit card clients from Taiwan with 23 features. The variables include:

- **Demographics:** Gender, Education, Marital Status, Age
- **Credit:** Credit limit
- **Payment history:** 6 months of payment status records
- **Amounts:** 6 months of bill amounts and payment amounts

The target variable indicates whether a client defaulted (0/1). There are no missing values.

Data Cleaning

We dropped the ID column since it's just a label and doesn't tell us anything useful (it represent the row number of all customer records). The default variable (our target) is either 0 for no default or 1 for default. We kept all 30,000 records since our data is clean.
One important thing: about 21% of customers in our data defaulted while 79% didn't. This imbalance is something we had to keep in mind when evaluating our model later.

3 Exploratory Data Analysis

Before jumping into modeling, we spent time looking at the data visually. It's really helpful to see what patterns exist and which variables seem connected to defaults.

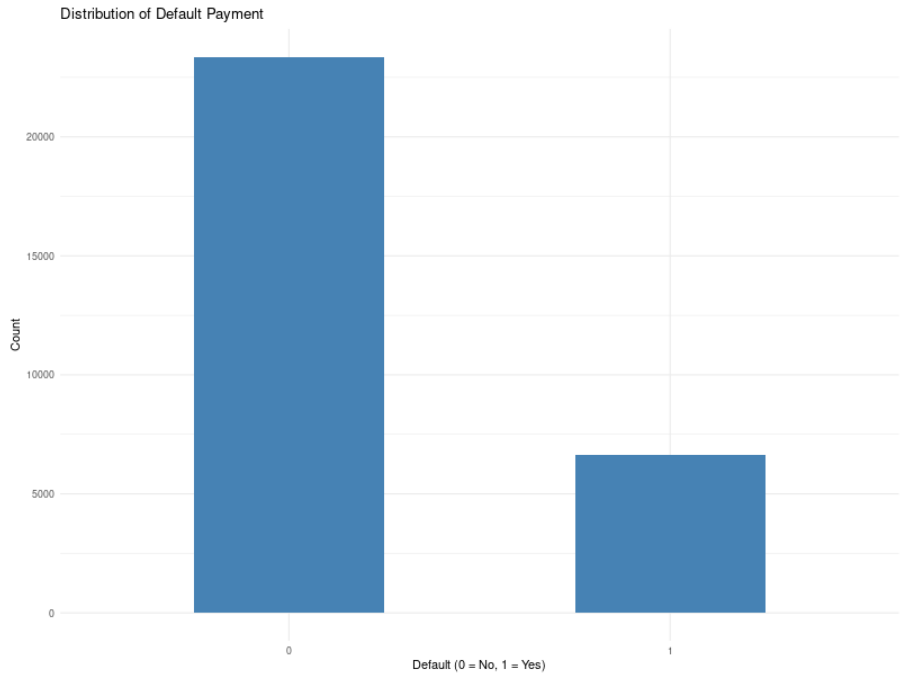

Figure 1: Target Variable Distribution

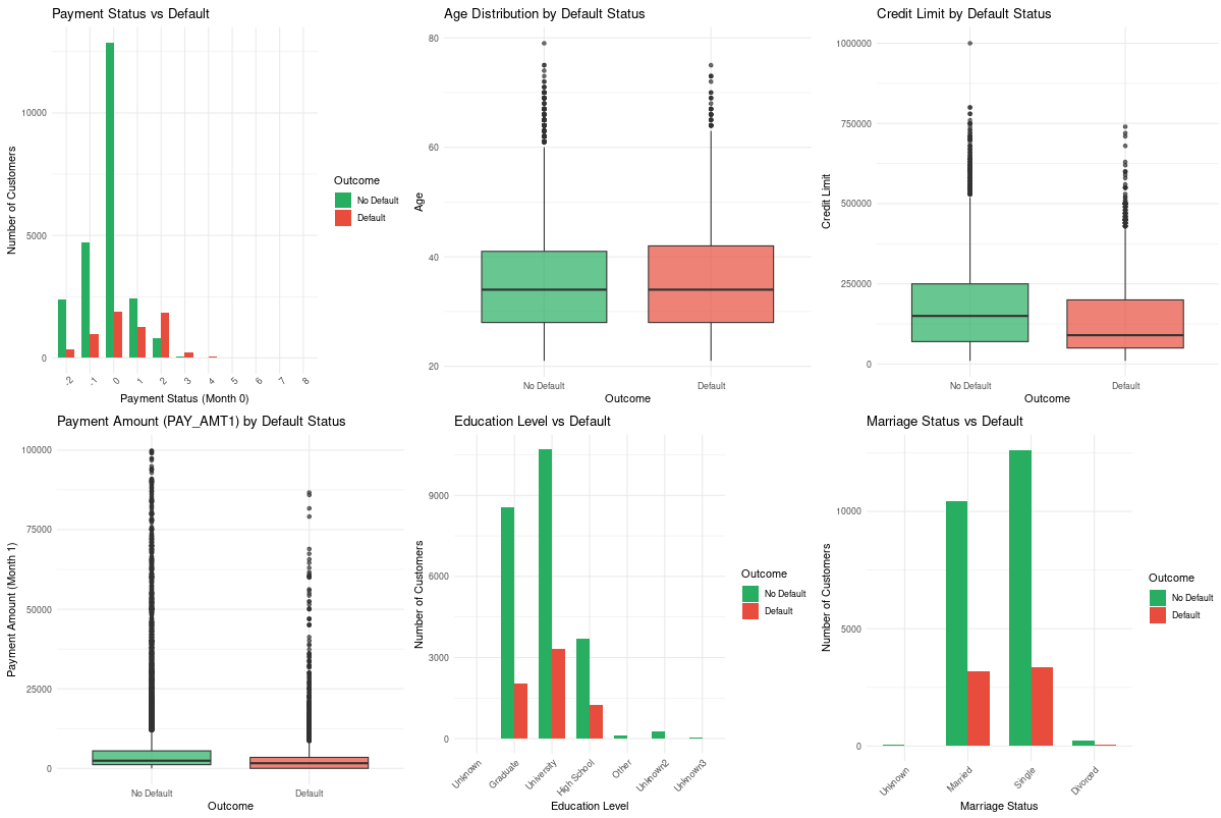

Figure 2: Exploratory Data Analysis: Relationship of Variables with Default Payment

We looked at several patterns that jumped out:

Payment Status (PAY_0): This was the biggest standout. People who were late on recent

payments had way higher default rates than those who paid on time. The numbers are coded as follows: -1 means paid on time, 1 means one month late, 2 means two months late, and so on. The time delay in payment really influence how likely clients were to default.
Credit Limit: People with higher credit limits tend to default less. This actually make sense, because probably banks gave higher limits to people they already trusted more.
Payment Amounts: Bigger payments generally meant lower default risk, which is somehow straightforward. If you're paying a lot, you're probably trying to stay on top of your debt.
Age and Demographics: Age did vary between the two groups, but not as dramatically as payment behavior. Education and marital status showed some patterns too, but nothing as strong as the payment stuff.
Bottom line: payment behavior is clearly the main predictor that matters the most, but the others also add some value to the model.

4 Feature Selection and Modeling

The Feature Selection Approach

We had 23 variables to choose from, but that doesn't mean we should use all of them. Including everything might cause overfitting - basically the model would memorize our specific data instead of learning real patterns. But using too few might mean we miss important information. Forward stepwise selection with AIC helps us find a middle ground: we start with the best single variable and keep adding others only if they actually improve the model.
From our earlier plots, here's what stood out:

- **Payment Status (PAY_0):** This was huge. Recent late payments were a big red flag. Remember: -1 = paid on time, 1 = one month late, 2 = two months late, etc. The longer the delay, the more likely they'd default.
- **Age:** The age distributions looked different between the two groups, though it wasn't as revealing as the payment history.
- **Credit Limit:** Higher limits usually meant lower default rates, probably because banks gave bigger limits to people they already trusted.
- **Payment Amounts:** Larger payments generally meant lower risk - makes sense since it shows you're working to pay things off.
- **Education and Marriage:** These showed some connection to default behavior, but not nearly as strong as payment patterns.

These observations helped us understand why certain variables ended up being important in our final model.

5 Feature Selection and Modeling

Full Logistic Regression Model

We first tried fitting a model with all 23 variables. A bunch of them were statistically significant, especially the late payment ones. The full model had a residual deviance of 27,877 and AIC of 27,925. When we looked at variables individually, PAY_0 (the most recent payment status) was clearly the strongest with AIC 28535.57. PAY_2 and PAY_3 came next. Things like bill amounts and age didn't add as much on their own.

Individual Feature AIC Evaluation

Feature	AIC	Delta_AIC
PAY_0	28535.57	0.000
PAY_2	29697.66	1162.094
PAY_3	30109.41	1573.843
PAY_4	30359.61	1824.041
PAY_5	30508.70	1973.135
PAY_6	30702.32	2166.753
LIMIT_BAL	30935.29	2399.718
PAY_AMT1	31358.87	2823.298
PAY_AMT2	31388.13	2852.558
PAY_AMT3	31527.77	2992.199

Table 1: Individual Feature AIC Evaluation

PAY_0 had the lowest AIC, with PAY_2 and PAY_3 coming after. Things like AGE and the BILL_AMT variables had much higher AIC values, which means they weren't as useful on their own.

Forward Stepwise Feature Selection using AIC

The idea is simple: start with PAY_0 (our best predictor) and then add other variables one at a time, but only if they actually make our model better.

Forward Selection Results

Through this process, we ended up with 18 out of 23 features. Starting from PAY_0 (AIC = 28,535.57), we kept adding variables that lowered the AIC. Our final model has AIC = 27,917.81, that's a drop of about 618 points, which is clearly good.
The 18 features we kept were: all the payment history variables (PAY_0 through PAY_5), payment amounts, bill amounts, credit limit, and demographic details like age, education, marital status, and gender.

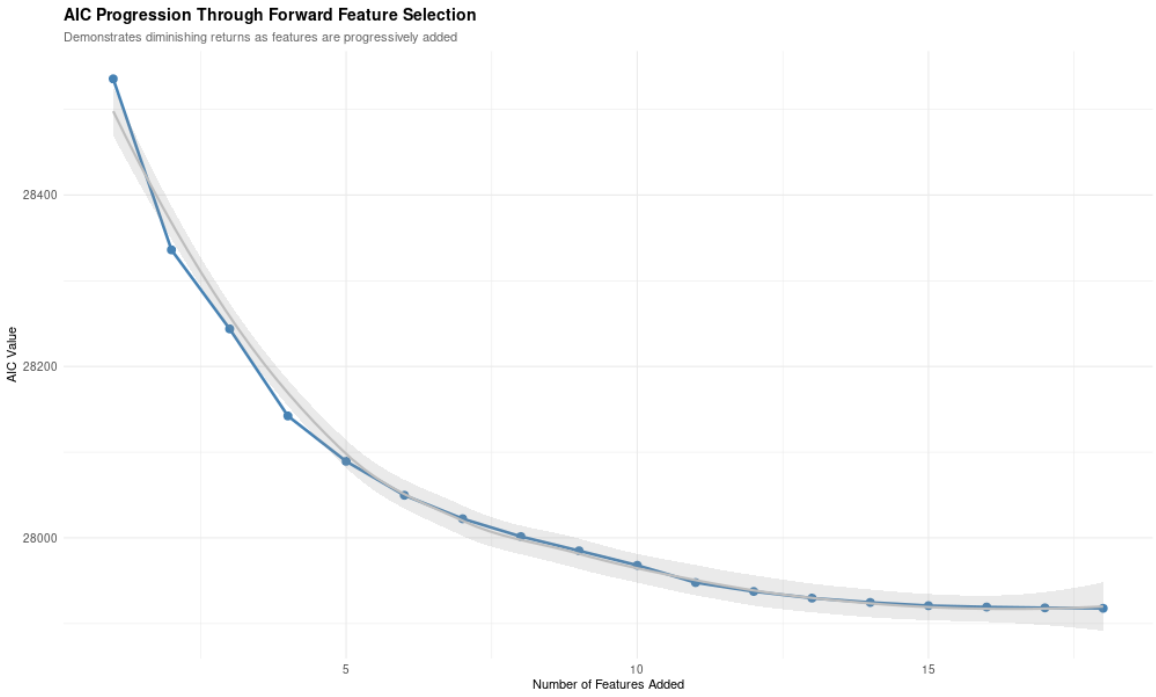

Figure 3: AIC Progression Through Forward Feature Selection

This plot shows how the AIC drops as we add more features. Each time we add a good variable, the AIC goes down (which is what we want). After about 18 features, the curve flattens out, which means adding more variables wouldn't really help the model much.

6 Model Coefficients and Interpretation

Now let's talk about what the model actually found. With our 18 features, we can look at the coefficients to see what increases or decreases default risk. Positive numbers mean higher risk, negative numbers mean lower risk.

Key Risk and Protective Factors

Things that INCREASE default risk:

- **PAY_0** (coefficient = 0.577): Being late on recent payments is the biggest factor. Each month of delay bumps up your default odds by 78%.
- **PAY_2, PAY_3, PAY_5**: Past payment delays also matter - payment issues tend to stick around.

Things that DECREASE default risk:

- **MARRIAGE** (coefficient = -0.156): Being married seems to help - it lowers default odds by about 14%.
- **EDUCATION** (coefficient = -0.105): More education helps too, cutting default odds by around 10%.
- **Payment amounts**: Making bigger payments is a good sign and lowers risk quite a bit.
- **LIMIT_BAL**: Having a higher credit limit usually means the bank already thought you were lower risk.

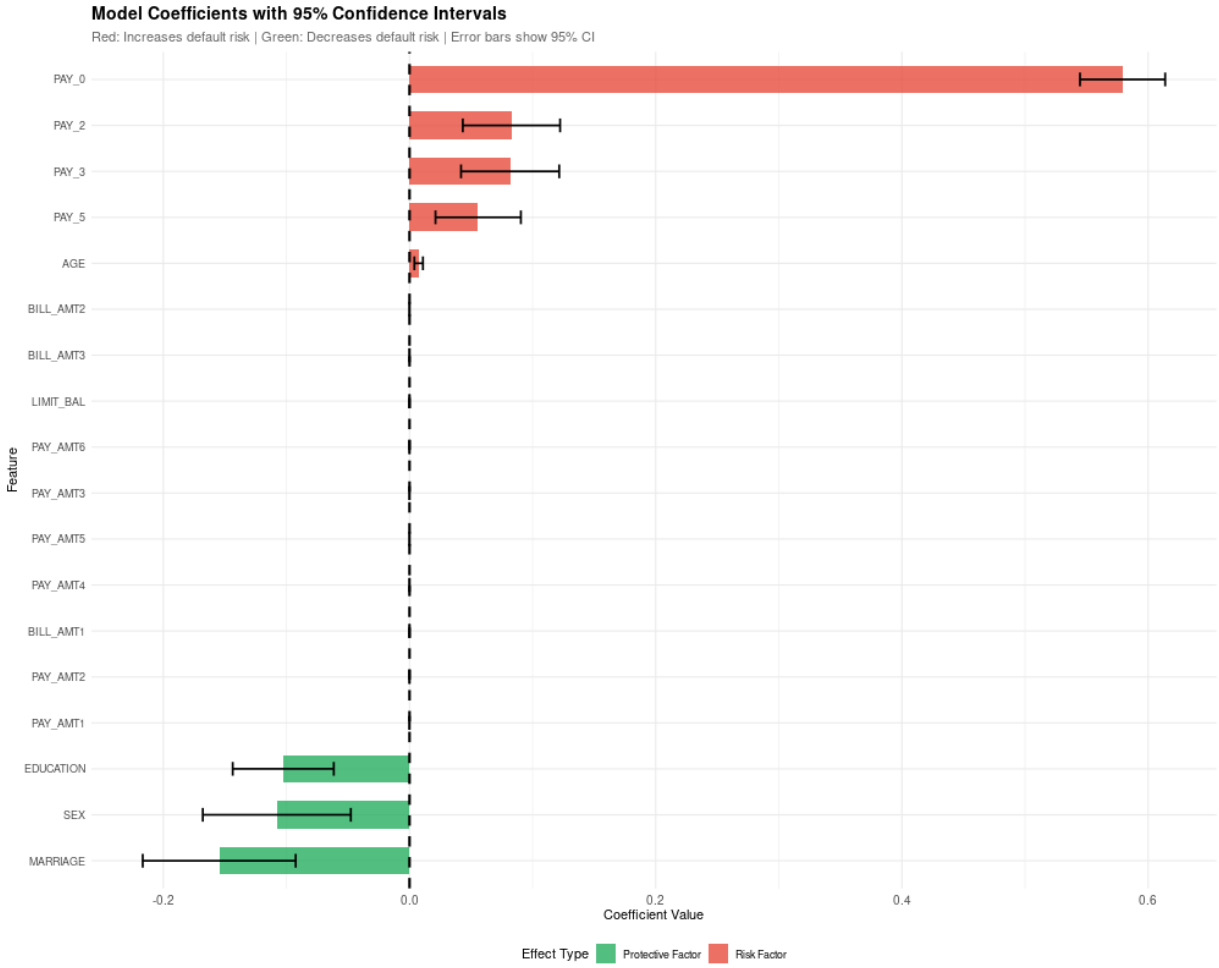

Figure 4: Model Coefficients with 95% Confidence Intervals

Most of these variables are statistically significant at $p < 0.001$, which means we can be pretty confident they actually matter. A couple variables (BILL_AMT2, PAY_AMT3, PAY_AMT6) aren't significant on their own at the usual 0.05 level, but we kept them because the AIC selection process showed they still add something to the overall model.

7 Model Evaluation

We need to test our model on fresh data it hasn't seen before. This tells us whether it'll actually work on new customers or if it just memorized the training data.

Data Preparation for Evaluation

To test this properly, we split our 30,000 records into training (70%) and test (30%) sets. We used stratified sampling, which just means we made sure both sets had the same 21% default rate. This keeps things fair - the test set looks like what we'd see in the real world. We ended up with 21,001 records for training and 8,999 for testing.

Model Predictions

We trained the model on the training set, then used it to predict on the test set. The predicted probabilities ranged from basically zero (1.12×10^{-5}) all the way up to 0.994. We used 0.5 as our cutoff - so if the model predicted a probability above 0.5, we called it a default.

Performance Metrics

There are several ways to measure how well our model does:

- **Accuracy**: How often does it get the answer right overall?
- **Sensitivity**: Out of everyone who actually defaulted, how many did we catch?
- **Specificity**: Out of everyone who didn't default, how many did we correctly identify?
- **Precision**: When we predict someone will default, how often is that right?
- **Balanced Accuracy**: The average of sensitivity and specificity (this matters since we have way more non-defaulters)
- **AUC**: Area under the ROC curve - basically tells us how good the model is at separating the two groups

Confusion Matrix

The confusion matrix shows what the model predicted vs. what actually happened. The diagonal shows when we got it right (true negatives and true positives). Off the diagonal is where we messed up (false positives and false negatives).

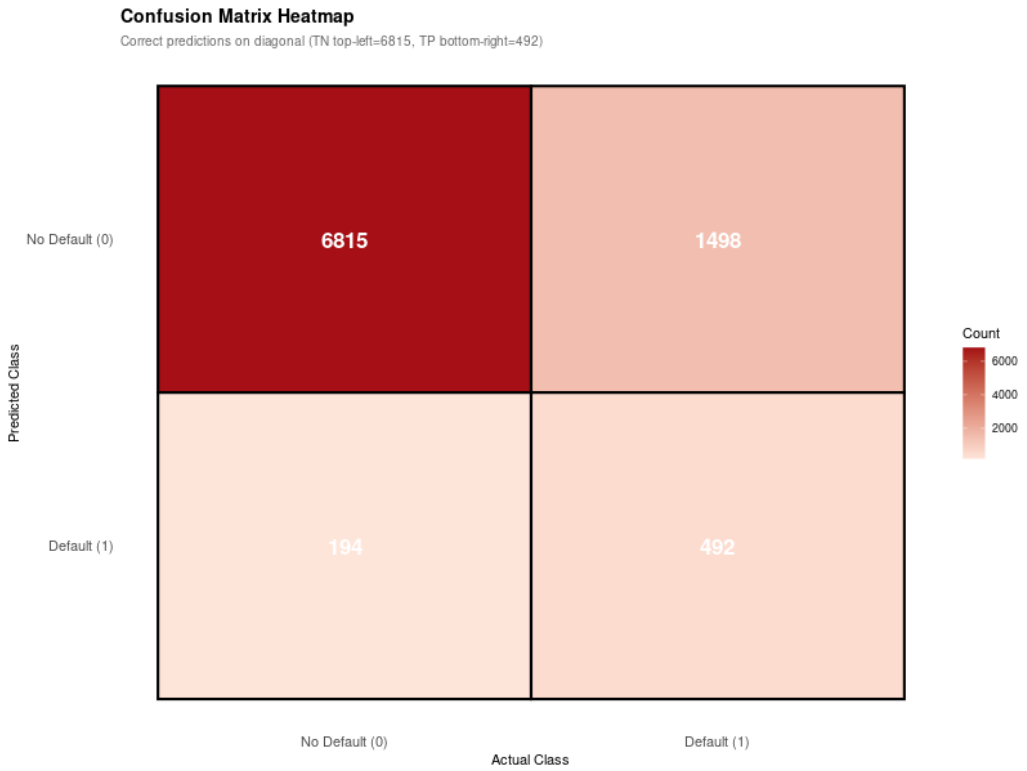

Figure 5: Confusion Matrix Heatmap

8 Results and Model Performance

Performance Metrics Summary

Metric	Value	Interpretation
Accuracy	81.20%	Overall correct predictions across both classes
Sensitivity	24.72%	Proportion of actual defaults correctly identified
Specificity	97.23%	Proportion of actual non-defaults correctly identified
Precision	71.72%	Of predicted defaults, how many are actually correct
Balanced Accuracy	60.98%	Average performance accounting for class imbalance
AUC	0.7250	Discrimination between default and non-default

Table 2: Performance Summary

Detailed Performance Analysis

Accuracy (81.20% with 95% CI: [80.38%, 82.00%])

Our model gets it right 81.20% of the time overall. So out of 100 people, we'd correctly classify about 81 of them. The confidence interval is pretty tight, which is good - means the result should hold up.

Sensitivity (24.72%): Catching Defaulters

This is where we struggle. We only catch about 25% of people who actually default. If 100

people were going to default, we'd only spot about 25 of them. This is partly because of the class imbalance - the model is being pretty conservative.

Specificity (97.23%): Identifying Safe Customers

On the flip side, we're really good at identifying people who won't default. We correctly identify 97.23% of them. So we almost never wrongly label a good customer as risky.

Precision (71.72%): How Often We're Right About Defaults

When we do predict someone will default, we're right about 72% of the time. So our warnings are fairly reliable when we make them.

Balanced Accuracy (60.98%)

Since our data is imbalanced (22% defaults, 78% don't), this metric gives us a fairer picture. At about 61%, we're doing okay but not great across both groups.

AUC (0.7250): How Well We Separate the Groups

An AUC of 0.7250 means our model does a decent job of telling defaulters apart from non-defaulters. It's definitely better than just guessing randomly (which would be 0.5), but there's definitely room to improve.

What the Trade-offs Mean

Our model is definitely on the conservative side: it's great at spotting safe customers but misses a lot of actual defaulters. Here's why:

1. The data has way more non-defaulters, so the model leans that way
2. We're using 0.5 as our threshold, which might not be the best choice
3. Logistic regression naturally favors the bigger class unless you do something about it

In the real world, this model could be useful for flagging obviously risky accounts. But banks definitely shouldn't use it as their only decision-making tool.

Practical Uses and Limitations

- **What It Does Well**: Really good at spotting safe customers (97% specificity).
- **Where It Falls Short**: Misses a lot of people who actually default (only 25% sensitivity).
- **Best Use Case**: Catching obvious red flags like recent late payments.
- **What It Might Miss**: People who suddenly default even though they had a clean record.
- **How To Improve**: We could lower the threshold to catch more defaults, but that would also mean more false alarms.

9 ROC Curve and Model Discrimination

The ROC curve shows how our model performs at different probability thresholds. It's really useful because you can see how sensitivity and specificity change when you adjust the cutoff. Different banks might want different trade-offs depending on how much risk they're willing to take.

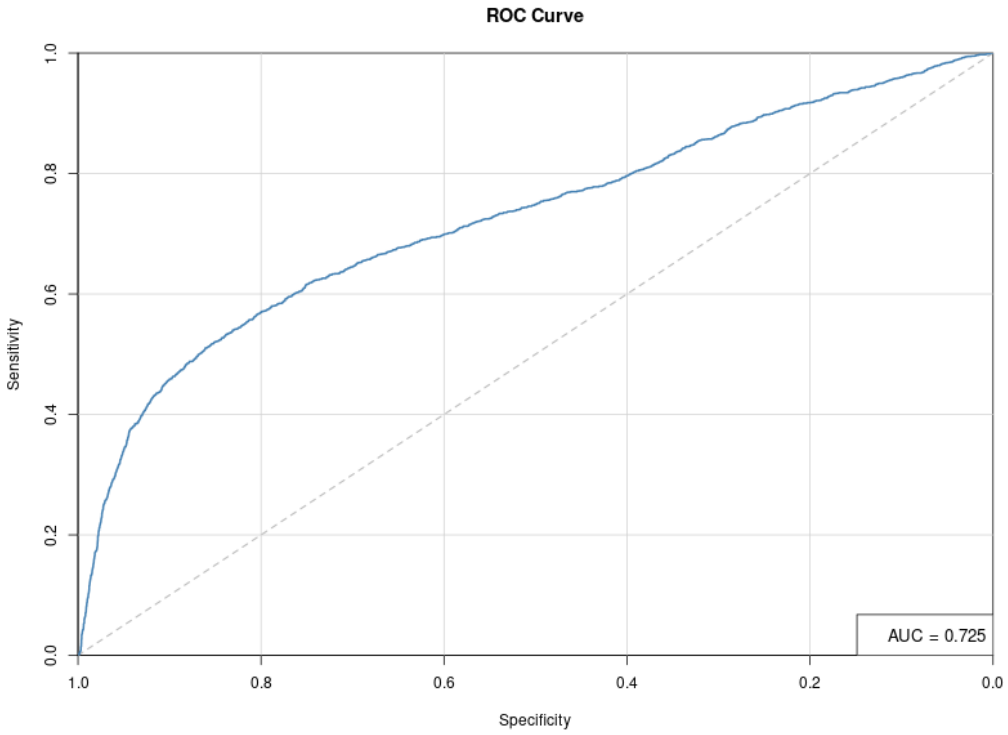

Figure 6: ROC Curve with AUC = 0.7250

Right now we're using 0.5 as our threshold - predict default if probability is above 0.5. But we could change that. If we lower it, we'd catch more actual defaulters but also get more false alarms. If we raise it, we'd have fewer false alarms but miss more real defaulters.

Our AUC of 0.725 tells us the model is doing better than random guessing. Here's one way to think about it: if you randomly picked one person who defaulted and one who didn't, our model would correctly give the defaulter a higher risk score about 72.5% of the time. That's decent but not amazing. The curve confirms what we already knew - we're way better at spotting non-defaulters than catching defaulters.

The nice thing is banks could adjust the threshold based on their priorities. Want fewer false alarms? Use a higher threshold. Want to catch more defaults early? Go lower.

Conclusion

Honestly, predicting credit card defaults is tough. You're always balancing between catching people who'll default and not falsely accusing good customers. Our model ended up being really conservative - it's great at identifying safe people (97% specificity) but misses most of the actual defaulters (only 25% sensitivity). This happened because our data has way more people who pay than who default, so the model kind of defaults to predicting "won't default." But that doesn't mean it's useless. When it does flag someone as risky, it's right about 72% of the time. And when it says someone's safe, it's almost always correct. So, it is safe for banks or financial institutions to trust the model when it marks someone as low-risk. Just don't expect it to catch every single defaulter. The biggest takeaway is that recent payment behavior matters way more than feature like age or education.

Where would this actually be useful? Probably as a first screening step, not the final answer. A bank could use it to quickly spot accounts that need a closer look. They could also lower the threshold to catch more defaults if they're willing to deal with more false positives. It really depends on how risk-averse they are.

What we learned is that even with messy, imbalanced real-world data, a simple logistic regression with good feature selection can work pretty well. It's not perfect by any means, but it's definitely better than nothing and could actually help inform decisions.