Credit Risk Assessment: Estimating Probability of Default using Logistic Regression

Ritesh Parbhoo

2026-02-08

Email: riteshparbhoo@outlook.com GitHub:

Project Description This project, implemented in R, involves developing a credit risk model to estimate the probability of default for borrowers using logistic regression (GLM). The model examines how key variables - employment status, bank balance and annual salary - predict the probability of a counterparty defaulting on their credit obligations. The model is trained on a training set and tested on a validation set, and various methods are used to evaluate the discriminatory power of the model.

Motivation Credit risk - the risk that a counterparty will default on their financial obligations - is a principal concern of lenders and financial institutions. Key to measuring credit risk is estimating probability of default which will allow institutions to make data-informed lending decisions, manage risk exposure, and comply with regulatory requirements. By developing a logistic regression model using key predictors - employment status, bank balance and annual salary - this project can identify high risk borrowers and improve credit allocation, thus reducing potential financial losses.

```r
library(pROC)
library(PRROC)
library(ggplot2)
library(broom)
```

Credit Risk Dataset

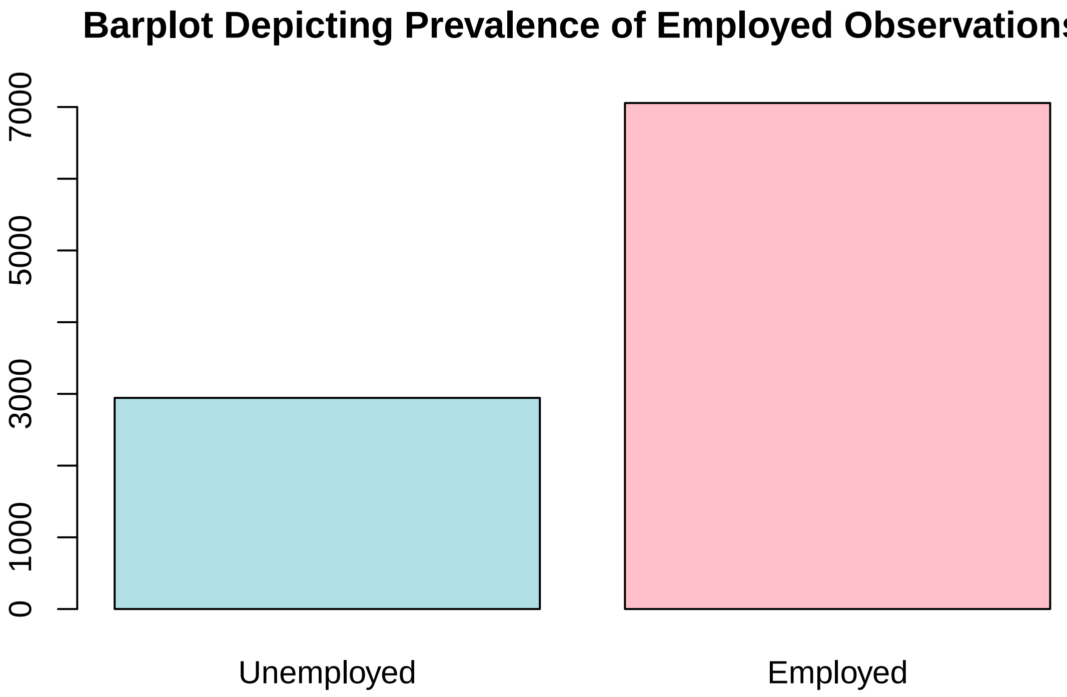
About the Dataset The data.frame containing the entire dataset is labeled df. The response variable, labeled "default_status" is a binary factor taking on values 0 or 1. The predictors consist of another binary factor, "employment_status", as well as two numerical values, "bank_balance" and "annual_salary", which have both been standardised and stored in the data.frame as "st_bank_balance" and "st_annual_salary" respectively.

```r
attach(df)
print(head(df))
```

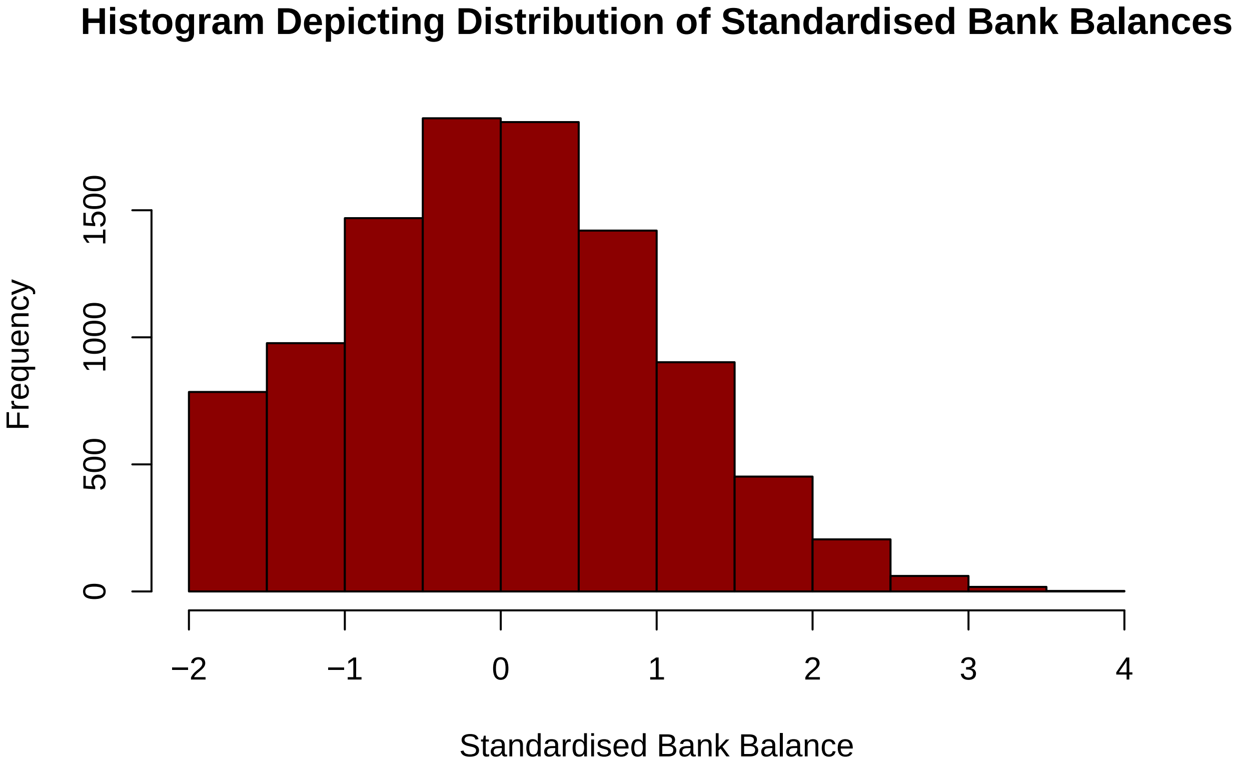
```
##   index employment_status st_bank_balance st_annual_salary default_status
## 1     1                 1     -0.21881663        0.8131470              0
## 2     2                 0     -0.03761487       -1.6054158              0
## 3     3                 1      0.49238734       -0.1312056              0
## 4     4                 1     -0.63286213        0.1640224              0
## 5     5                 1     -0.10277722        0.3708969              0
## 6     6                 0      0.17410072       -1.9514227              0
```

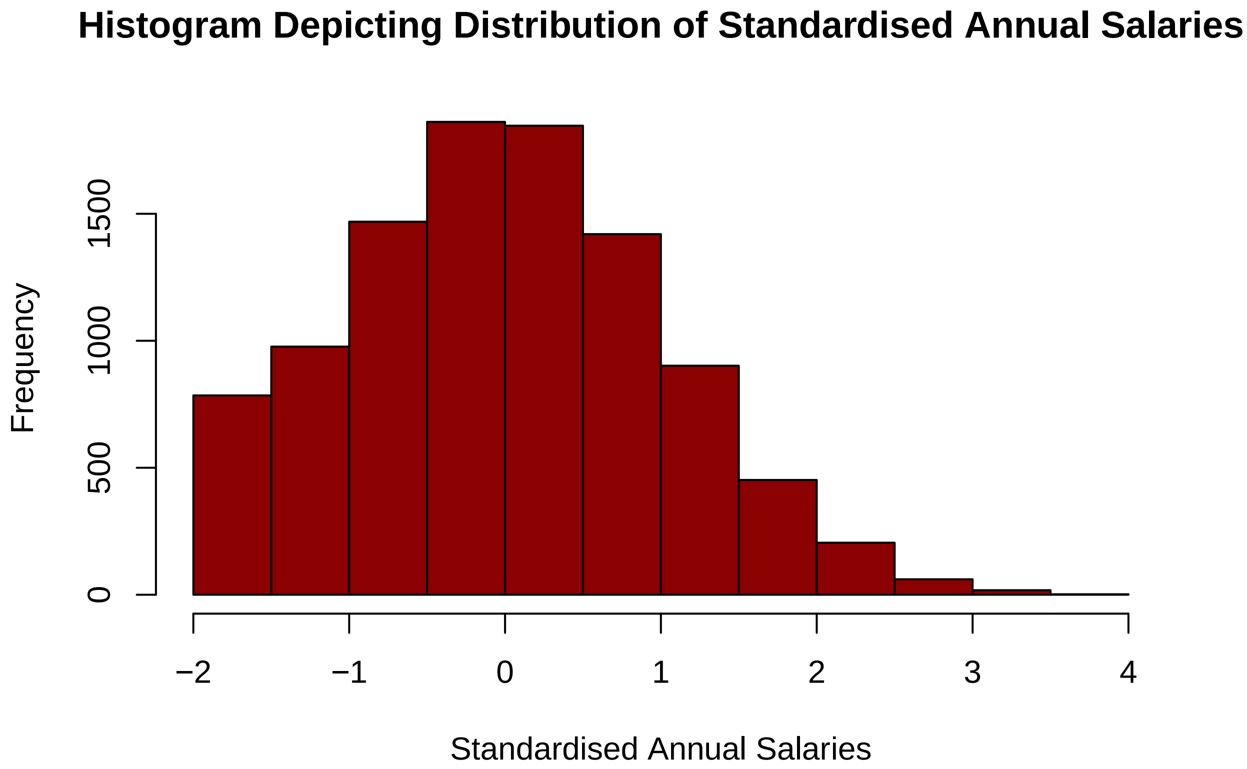
```r
print(summary(df))
```

```
##      index       employment_status st_bank_balance    st_annual_salary
##  Min.   :    1   0:2944            Min.   :-1.72700   Min.   :-2.45527
##  1st Qu.: 2501   1:7056            1st Qu.:-0.73110   1st Qu.:-0.91301
##  Median : 5000                     Median :-0.02427   Median : 0.07766
##  Mean   : 5000                     Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 7500                     3rd Qu.: 0.68414   3rd Qu.: 0.77161
##  Max.   :10000                     Max.   : 3.76037   Max.   : 3.00205
##  default_status
##  0:9667
##  1: 333
##
##
##
##
```



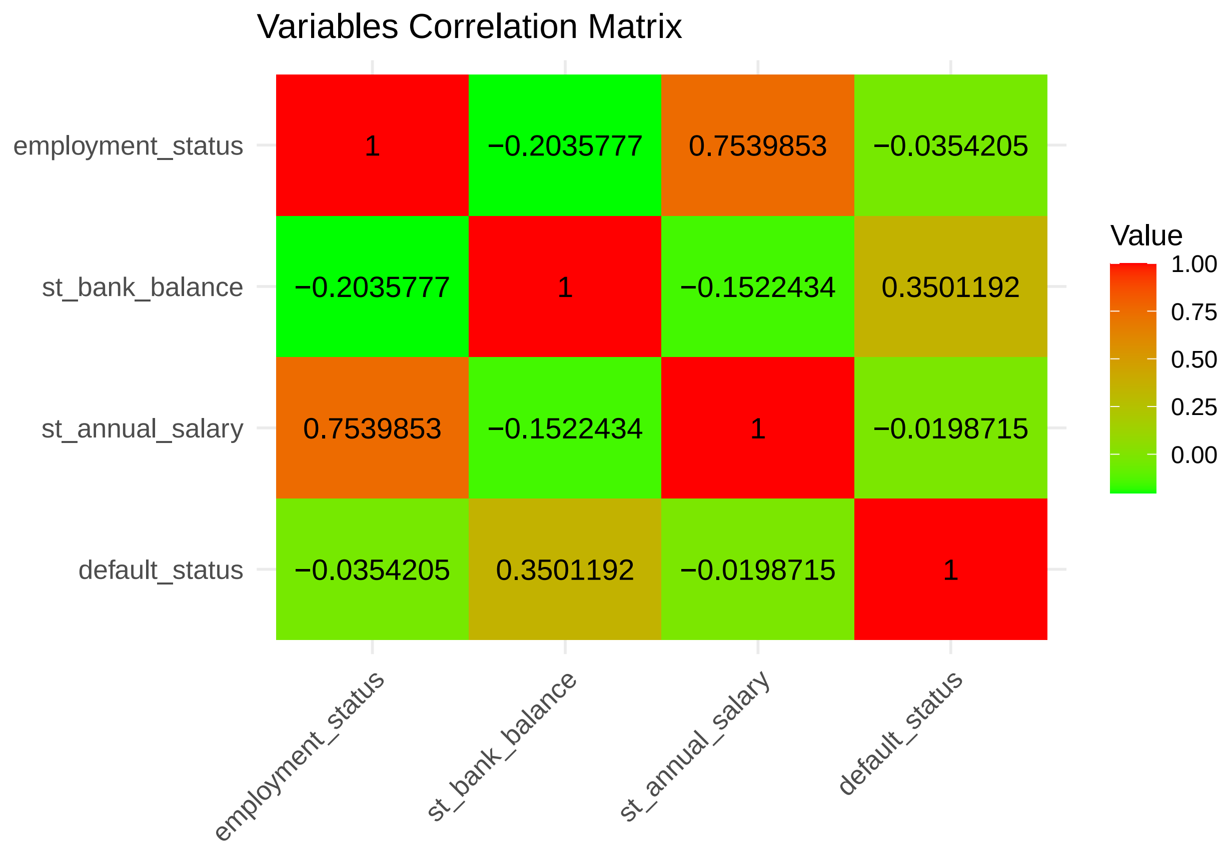
Visualising the Data





As would be expected, the distributions of the standardised variables are positively skewed since before they were standardised, their values were non-negative.

Correlation between Variables Analysis of the correlations between predictors and between the response and each predictor will help determine which predictors are worth including in the final fitted logistic regression model.



Of the three predictors, it is clear that the standardised bank balance shares the strongest correlation with default status, and this correlation is positive and one would expect intuitively. Employment status and annual salary share weak correlation with default status (approximately uncorrelated). Furthermore, employment status and standardised annual salary share a strong positive correlation and it may be worth removing one from the fitted logistic regression model. If either were to be removed, it makes sense that annual salary is removed as employment status is much more easily measurable.

Fitting the Logistic Regression Models

The dataset of 10000 observations will be partitioned into two subsets, the training set and the validation set. The training set comprises 80% of the original dataset, selected at random. The remaining 20% of the observations make up the validation set (V.set) and will be used to evaluate the final model once it has been chosen.

```
logit_model.ebs <- glm(default_status ~ employment_status + st_bank_balance + st_annual_salary, family =
logit_model.eb <- glm(default_status ~ employment_status + st_bank_balance, family = binomial, data=T.se
logit_model.bs <- glm(default_status ~ st_bank_balance + st_annual_salary, family = binomial, data=T.set
logit_model.b <-glm(default_status ~ st_bank_balance, family = binomial, data=T.set)
```

It is clear that st_bank_balance is the strongest predictor of default_status. There are four combinations of predictors that include st_bank_balance and a logistic regression model using each such combination of predictors was fitted and they will be compared.

E := employment_status, B := st_bank_balance, S := st_annual_salary

Model Comparison

```
##         Residual_Deviance Null_Deviance      AIC Predictors
## Model 1          1206.627      2286.454 1214.627    E, B, S
## Model 2          1207.315      2286.454 1213.315       E, B
## Model 3          1210.250      2286.454 1216.250       B, S
## Model 4          1225.128      2286.454 1229.128          B
```

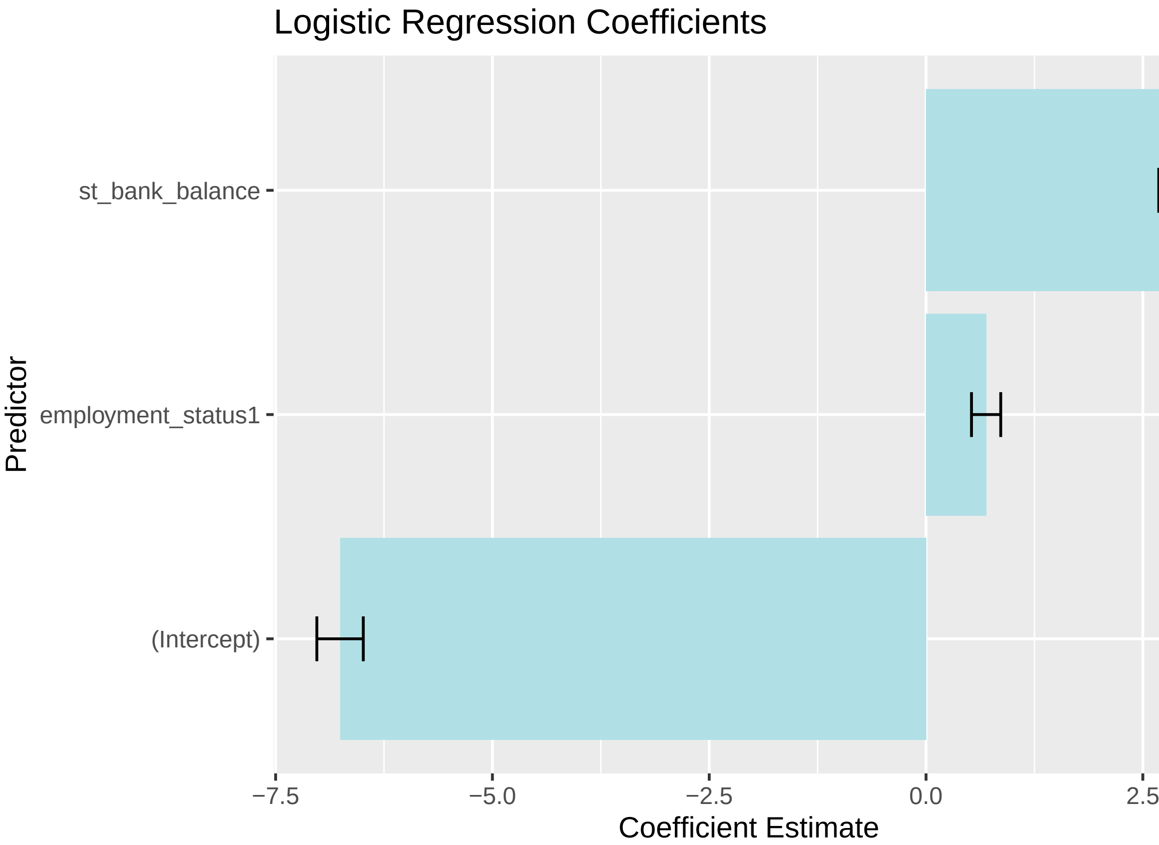
Null deviance is the deviance of the intercept-only model and serves as a baseline. It is the same across all four models and therefore lacks discriminative power in this case. Residual deviance measures the unexplained variation once the predictors are included; lower values indicate better fit. AIC is the most useful metric here, as it balances residual deviance with model complexity by applying a penalty for additional predictors. Model 4, which considers only st__bank__balance, is the simplest but has the poorest fit since its AIC value the greatest by a wide margin. Including at least one additional predictor to Model 4 adds predictive power. Model 2 outperforms Model 3 in both AIC and residual deviance and hence Model 2 dominates Model 3. Incorporating st__annual__salary into Model 2 to obtain Model 1 yields a marginal improvement to residual deviance, but worsens AIC. Therefore, the most parsimonious model and best-fitting model is Model 2 (employment__status, st__bank__balance).

Best-Fitting Model: Model 2 (employment__status, st__bank__balance)

```r
pred_prob <- predict(logit_model, newdata = V.set, type="response")

tidy_model <- broom::tidy(logit_model)

# Plot coefficients
ggplot(tidy_model, aes(x = term, y = estimate)) +
  geom_col(fill = "powderblue") +
  geom_errorbar(aes(ymin = estimate - std.error, ymax = estimate + std.error), width = 0.2) +
  coord_flip() +
  labs(title = "Logistic Regression Coefficients", y = "Coefficient Estimate", x = "Predictor")
```



Model Coefficients

In the logistic regression model, the intercept represents the log-odds of the outcome. Here the intercept is large and negative, indicating the baseline probability of default is small. Baseline in this context refers to (employment_status = 0, st_bank_balance = 0). A baseline observation is one that is unemployed and has exactly the mean bank balance of all the observations (since bank balance was standardised).

We can obtain the baseline probability of default by inputting the intercept into the log-odds (intercept) into the logistic transformation f(x) = 1/(1+e^-x).

```
baseline_prob.1 <- plogis(logit_model$coefficients[1])
```

```
## [1] "Baseline probability using logistic transformation on model intercept: 0.00116159589972863"
```

Reproducing the baseline probability by applying the model to an artificial baseline observation (employment_status = 0, st_bank_balance = 0):

```
baseline_obs <- data.frame(employment_status = as.factor(0),
                           st_bank_balance = 0)
baseline_prob.2 <- predict(logit_model, newdata = baseline_obs, type = "response")
```

```
## [1] "Baseline probability obtained by applying model to baseline observation: 0.00116159589972863"
```

```
tidy_model <- tidy(logit_model)
```

```
bank_seq <- seq(min(T.set$st_bank_balance), max(st_bank_balance), length.out = 100)

#ggplot(V.set, aes(x = salary_seq, y = pred_prob)) +
#  geom_point(color = "darkred", size = 1) +
#  labs(title = "Predicted Probability of Default vs Standardised Annual Salary",
#       x = "Bank Balance", y = "Predicted Probability")
```

Visualisation of Model

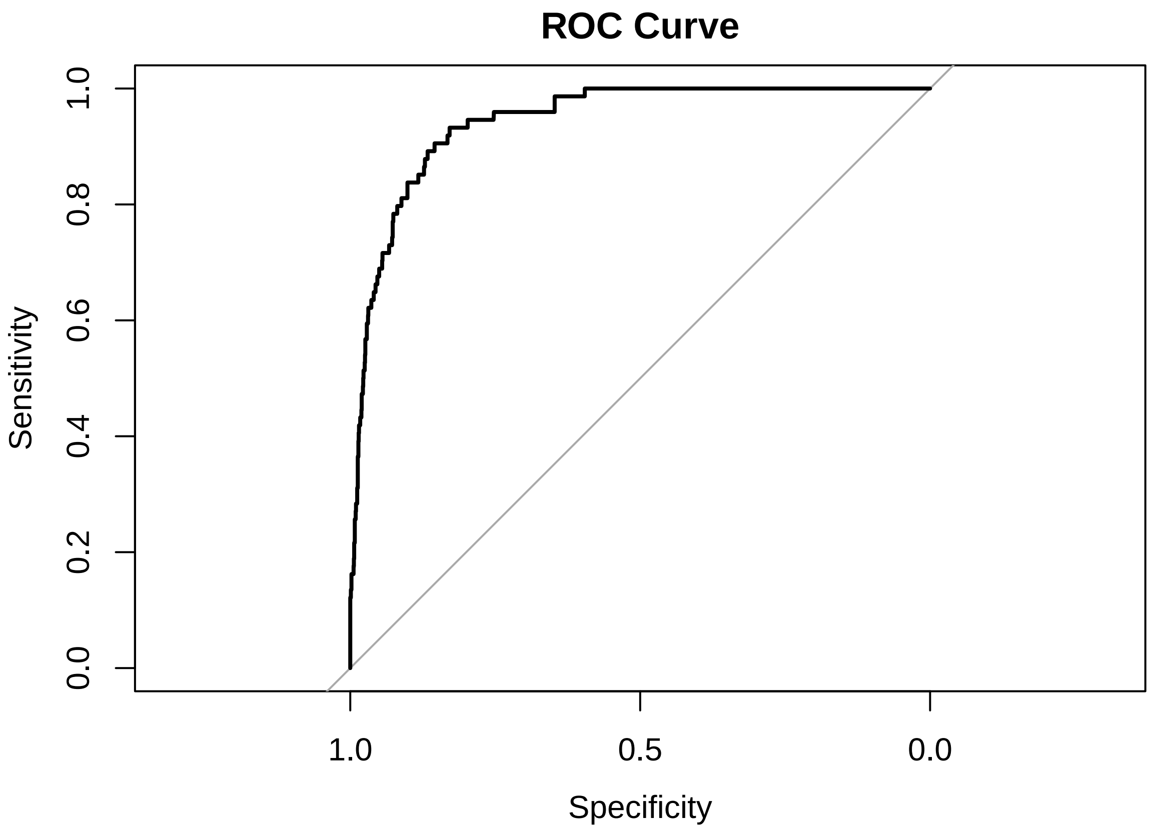
ROC Curve and AUC (Area under Curve)

```
roc_curve <- roc(V.set$default_status, pred_prob)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(roc_curve, main = "ROC Curve")
```



```
auc <- auc(roc_curve)
gini <- 2*auc - 1
```

“The logistic regression model demonstrates strong discriminatory power on an out-of-sample validation set (AUC = 0.943). This performance is driven by economically intuitive predictors and a clean data-generating process. Further diagnostics focus on probability calibration and stability, as ROC metrics alone do not assess absolute PD accuracy.”

Confusion Matrix

```
# this is done on the validation set only
# the usual cutoff

cutoff <- length(V.set$default_status[V.set$default_status == 1])/length(V.set$default_status)

val_predictions <- ifelse(pred_prob >= cutoff, 1, 0)

confusion_matrix <- table(
  Actual = V.set$default_status,
  Predicted = val_predictions
)

# [TN FP]
# [FN TP]
```

Precision-Recall Curve

Calibration Plots